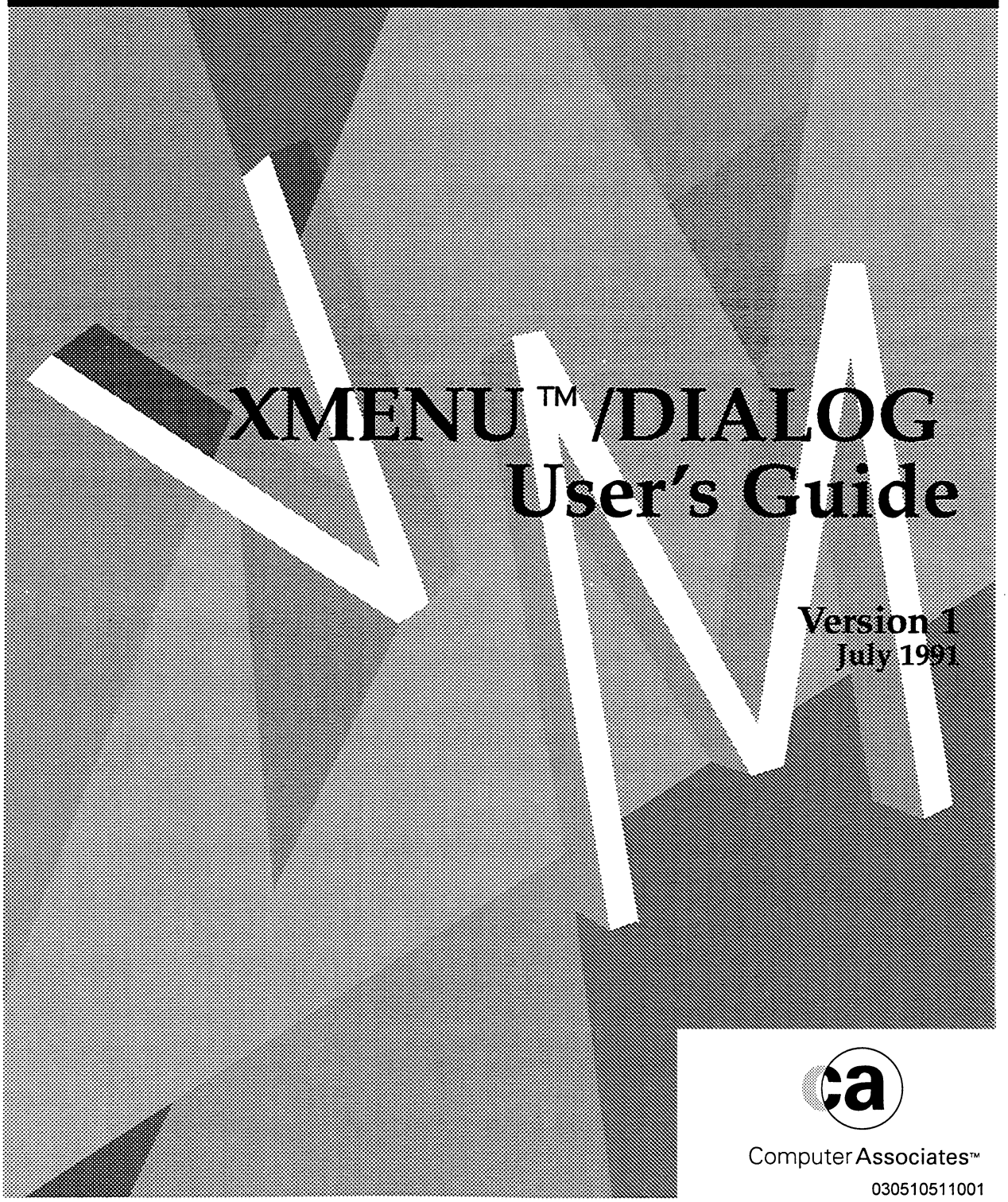
XMENU



XMENU™/DIALOG User's Guide

Version 1
July 1991

Computer Associates™

030510511001

Preface

This document describes XMENU/DIALOG and its primary components, XMENU Common User Access (XMENUCUA) and XMENU Hierarchical Menu Facility (XTREE). These components work individually and cooperatively to provide a high level of application development using XMENU as the screen manager.

The first section of the manual describes XMENUCUA and its facilities. The second section describes XTREE.

Contents

XMENU Common User Access (CUA) Facility

This section describes the XMENU CUA Guide Mode facility, referred to by its run-time name, *XMENUCUA*.

The purpose of this document is to allow you to use XMENUCUA to create XMENU menus that are compatible with the Systems Applications Architecture (SAA) Common User Access (CUA) standards as defined in the IBM *Systems Applications Architecture Common User Access Basic Interface Design Guide,* number SC26-4583, dated December, 1989.

This document does not attempt to teach you all of the SAA/CUA standards. In fact, we are going to assume that you have at least a passing familiarity with SAA/CUA and SAA/CUA terms. Please refer to the IBM SAA/CUA Basic Design Guide if you need to learn this information.

The sample menu used in this guide was taken from the Basic Design Guide so that you can see how a real SAA CUA panel is created.

Creating a CUA Compatible Panel

This chapter shows you a sample XMENUCUA session. This will show you how to create an SAA/CUA panel. It will also show you how to use XMENUCUA.

To make the information clearer, we have placed each screen on the top of a new page, followed by a description of its contents.

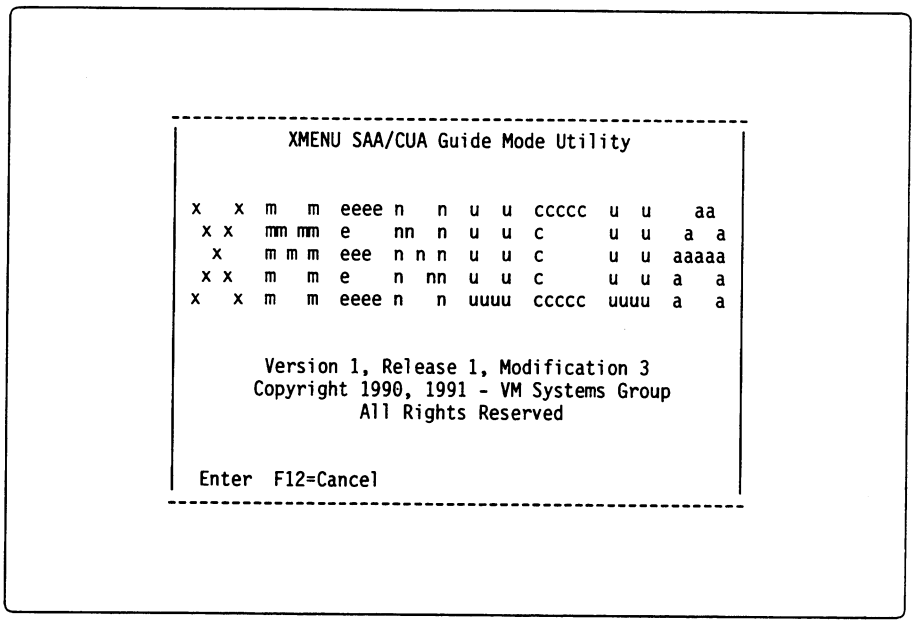
Starting XMENUCUA

To run XMENUCUA, type:

```
XMENUCUA [filename] [( NOLogo]
```

If a filename is supplied, this file is automatically loaded. If *NOLOGO* is specified, the initial copyright menu is not displayed.

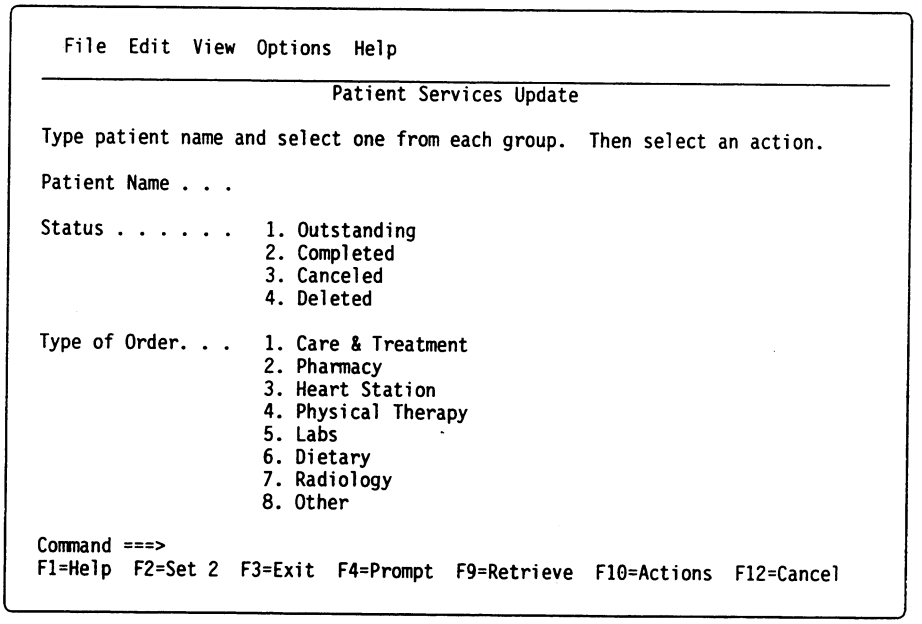
Here is what you will see when you simply type *XMENUCUA*:



Press *Enter* to continue running XMENUCUA. Press *PF12* to exit XMENUCUA.

The Sample CUA Panel

Before we show you how XMENUCUA works, here's a copy of the menu we are going to create:



This menu is used extensively in the IBM SAA/CUA Basic Design Guide as an example. You might want to refer back to this page from time to time to see what we are going to create.

To edit this menu, type:

```
XMENUCUA PAGE159
```

This will open the XMENUSAA file (files containing CUA-specific information have a filetype of XMENUSAA) defining this already-created CUA panel.

If you want to create the panel completely from scratch, use a different filename, for example,

```
XMENUCUA MYPANEL
```

Note that input fields may be displayed using underscore characters on your terminal, even though the pictures in this book do not show them.

The XMENUCUA Main Menu

Once XMENUCUA begins, after the optional copyright menu, the following screen is displayed:

```
  File  Edit  View  Options  Help
XMECUA0                          XMENU CUA Prompter

Select one of the following.  Then press Enter.

    1. Create an Action Bar
    2. Create the Panel I.D.
    3. Create the Panel Title
    4. Create an Informational Area
    5. Create a Selection Field or List
    *. Create an Action List
    7. Create an Entry Field or Protected Text
    8. Create a Command Area and/or Error Message Line
    9. Create a Function Key Area

    10. Change the order of Panel Elements
    11. Save the menu in XMENU menu format
    *2. Save the menu in Dialog Tag Language (DTL) format


Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

All CUA panel creation is performed by selecting items from this menu. Each portion of a CUA panel is defined by choosing the number that corresponds to the area you are creating or editing. For example, choosing *1* would display another menu allowing you to define your action bar.

Note that this menu, and all XMENUCUA menus are in SAA/CUA format, and are controlled using SAA/CUA interface rules.

You can type a number to the left of the first item in the selection list and press *Enter* to go to that XMENUCUA function. You can also position the cursor beside or on a selection and press *Enter* to go to that function.

You can move the cursor to an item on the Action Bar at the top of the menu to perform functions defined by that item.

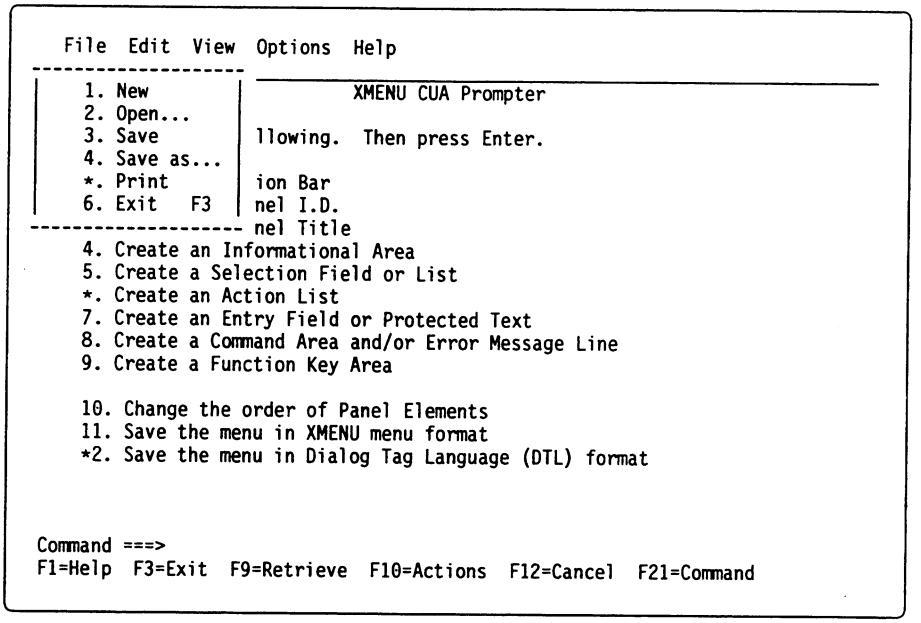
Finally, you can use the command line at the bottom of the menu to enter CP, CMS, or a small number of XMENUCUA commands.

XMENUCUA maintains a ten command retrieve stack, so that you can recall and reissue previously entered commands. The "?" subcommand will retrieve previous commands. The "=" subcommand will reexecute the last command issued.

Notice that a number of XMENUCUA commands can be invoked by pressing Program Function Keys (PFK). These are shown at the bottom of the menu. In this document we refer to these keys as PFKeys or function keys. We also on occasion refer to them by name, for example, we may ask you to press the *Action* key.

The XMENUCUA Action Bar

To show you how the Action Bar works, place the cursor on the word *File* or press the *Actions* PFK. Then press *Enter.* Here is what you should see:



In keeping with SAA/CUA protocols, a pull-down window appears showing you the choices available under *File.*

There are several choices:

New allows you to begin a new panel or panel section.

Open allows you to open an existing panel or panel section.

Save allows you to save this panel or panel section under the current filename.

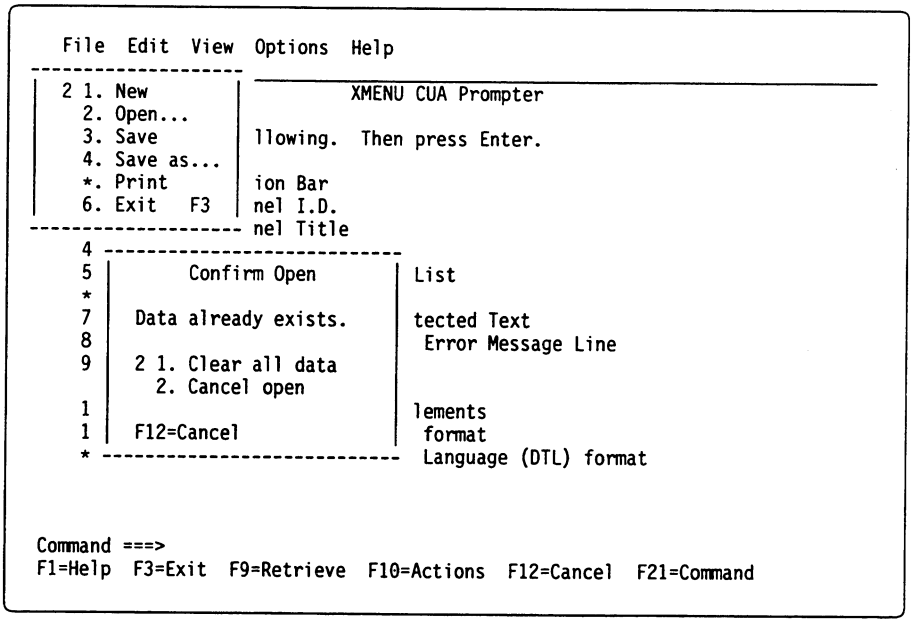
Save as... allows you to save this panel or panel section under the a new filename.

Print currently has no XMENUCUA function.

Exit allows you to exit XMENUCUA.

Opening an XMENUCUA File

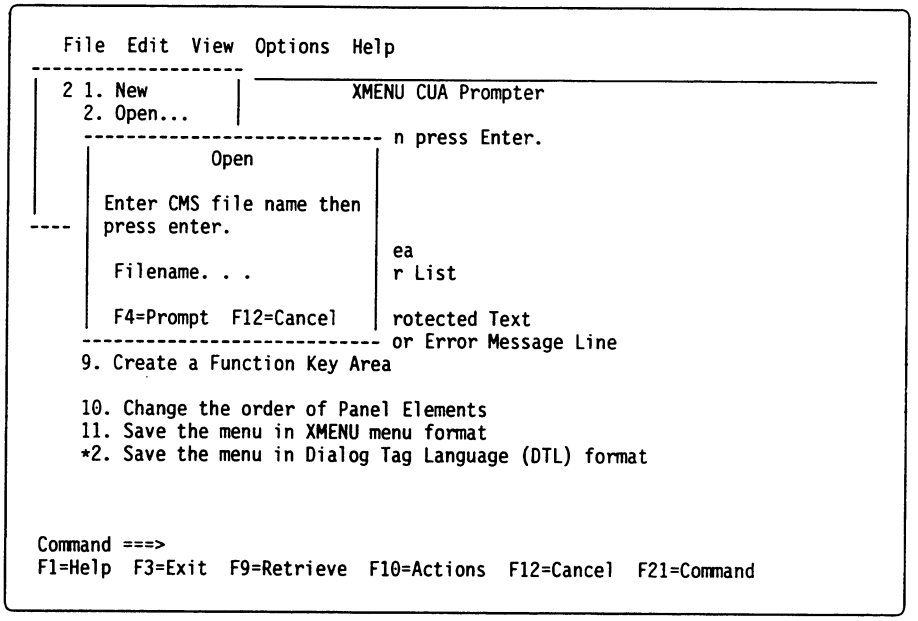
XMENUCUA places CUA-specific information into files with a filetype of XMENUSAA. To show you how an existing file is opened, type *2*, then press *Enter*. If you were already editing a panel, the following additional pop-up window is displayed:



Note that the default, that is if you were to simply press *Enter*, is to cancel the open (so that you don't lose your existing work).

If you change the selection to *1*, the open process would continue.

If you are creating a new file, or if you selected item 1 from the Open Confirmation menu, you will see the following:

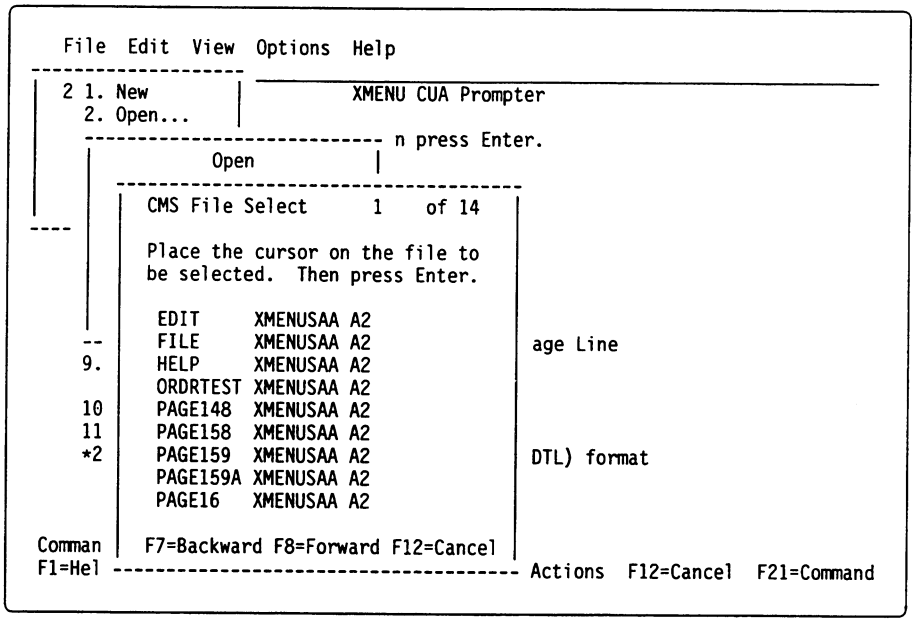


You can do one of three things at this point:

1. You can cancel the Open processing by pressing PF12.
2. You can type a valid CMS filename and press *Enter*.
3. You can display a list of existing CMS XMENUSAA files by pressing the Prompt PFK.

If you use file prompting, you can use the SAA/CUA wildcard characters, * and *?*.

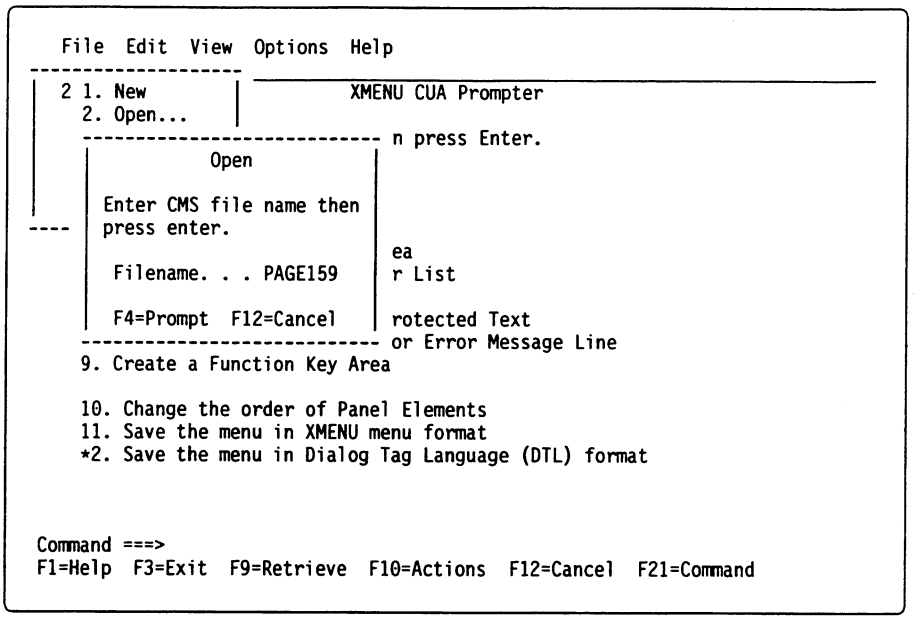
If you press the Prompt PFK, you will see yet another pop-up window such as the following:



At this point, you can position the cursor on any displayed file and press *Enter*. This will select the file, and move it into the previous pop-up window.

As an example, position the cursor on *PAGE159* and press *Enter*.

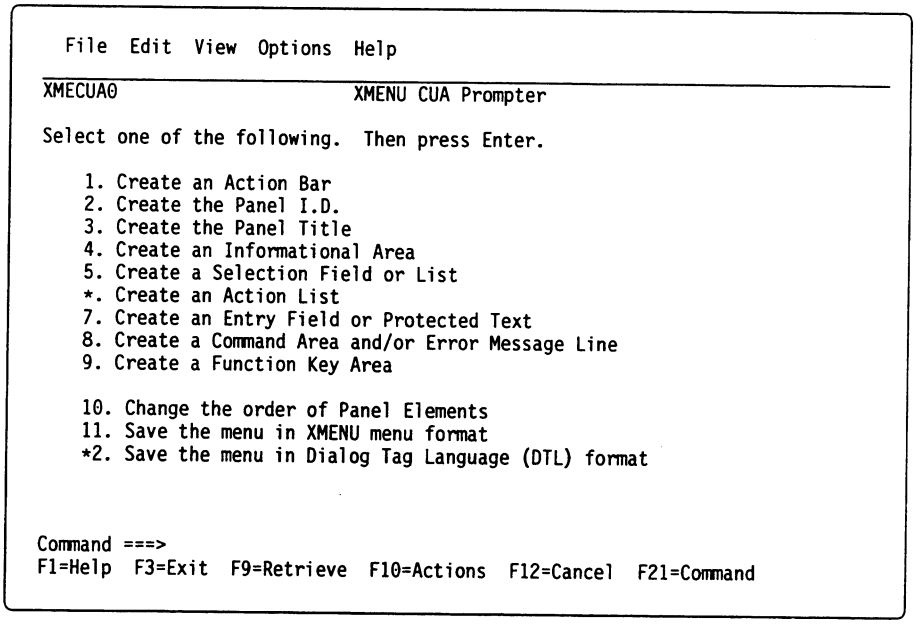
If you either selected PAGE159 from the CMS file menu or typed it in yourself, you should get the following display:



Press *Enter* and the XMENUSAA file PAGE159 will be opened for editing.

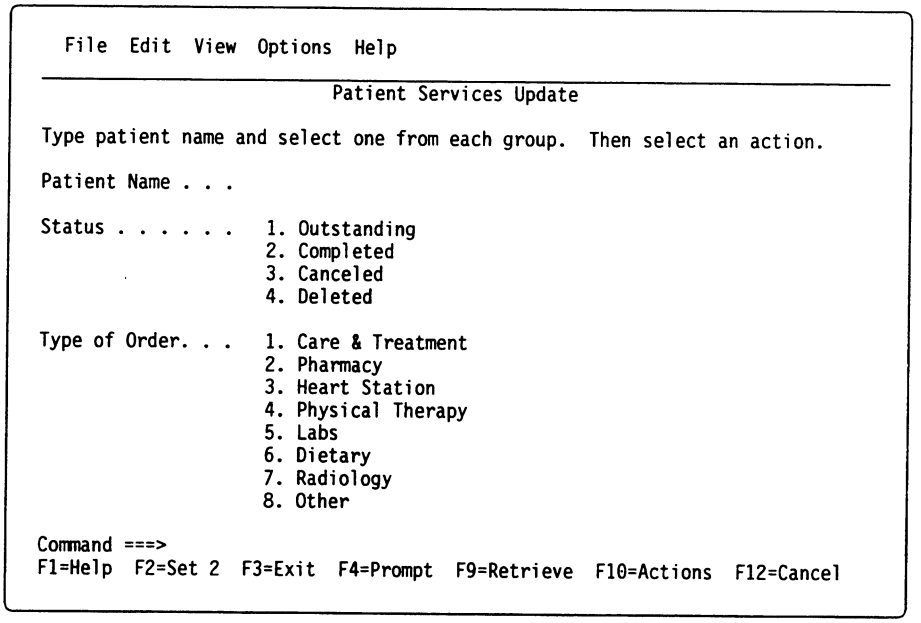
This file contains the example we are working with in this chapter. You can clear it out by selecting *New* from the File pull-down window now if you want to type the example in yourself.

Once you have opened the file, all pop-up windows are removed and you return to this menu:



It looks exactly the same—you won't notice loaded data until you go to CUA panel sections.

The Sample CUA Panel Revisited



If you look at our example above which is named PAGE159, (because that's where it is in our copy of the SAA/CUA Basic Design Guide manual), you will see that moving from top to bottom it is made up of various sections:

- An Action Bar
- A Title
- A line of informational text
- An entry area
- Two selection lists
- A command line
- A function key line

We will go through and create each of these sections in order, from top to bottom by selecting from the XMENUCUA main menu's selection list.

First we will create the Action Bar. To do this, select item *1* from the list and press *Enter*.

Creating an Action Bar

When you select *Create Action Bar*, the following will be displayed:

```
  File  Edit  View  Options  Help
XMECUA1                  XMENU CUA Prompter - Create Action Bar

Enter Action Bar items below.  Then press Enter.

Action Bar Item  1. . . . File                1-20 characters
Action Bar Item  2. . . . Edit
Action Bar Item  3. . . . View
Action Bar Item  4. . . . Options
Action Bar Item  5. . . . Help
Action Bar Item  6. . . .
Action Bar Item  7. . . .
Action Bar Item  8. . . .
Action Bar Item  9. . . .
Action Bar Item 10. . . .
Action Bar Item 11. . . .
Action Bar Item 12. . . .
Action Bar Item 13. . . .
Action Bar Item 14. . . .


Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

You type each separate action bar item on a single entry line in this menu. In our example, there are five action bar items—File, Edit, View, Options, and Help. Type each on an individual line, then press *Enter*.

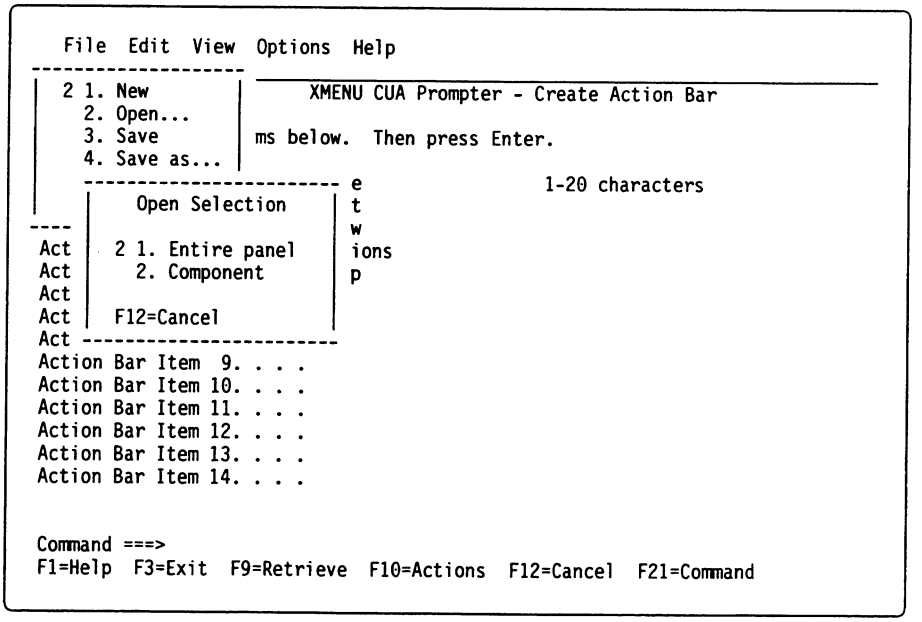
Note: Pressing ***Enter*** ensures that all updates to the screen are accepted.When you are done, the display should look like the one on this page.

When you have finished entering the action bar items, press the *Cancel* key to return to the XMENUCUA main menu.

Do not worry about the length of action bar items—XMENUCUA will automatically place them on the top of the panel so that they never wrap a line boundary.

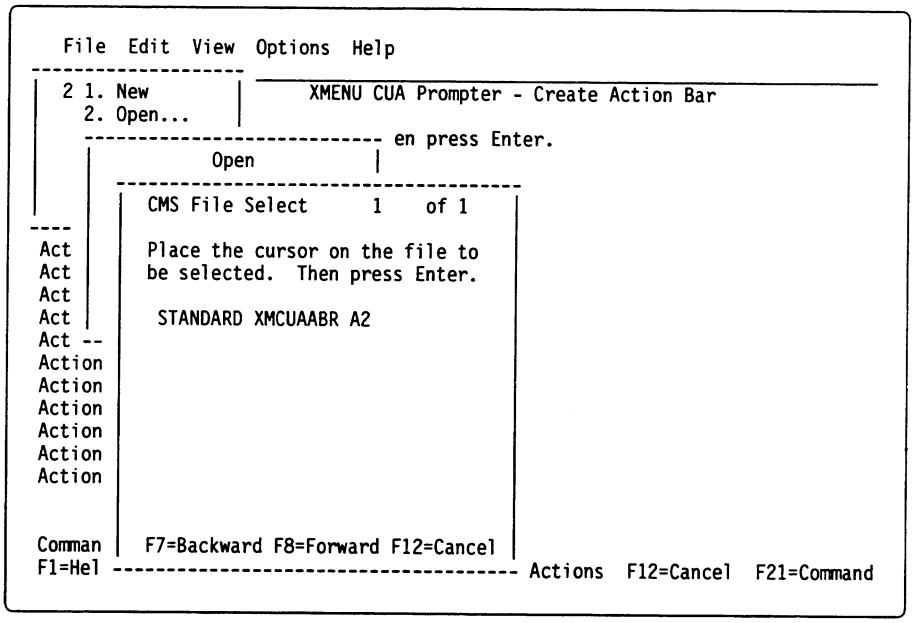
These five action bar items are standards for CUA, and can be found in most applications (including XMENUCUA). For this reason, we've included them in a separate file that can be loaded into your CUA panel, thus saving you the trouble of retyping them.

If you want to load the standard action bar items into your CUA panel, select the XMENUCUA action bar item *File* and then *Open*. You should get the following display:



Since you can either open the entire panel or simply this component, a new pop-up is displayed asking you which you want. Choose *2* (the default).

Again, you could ask for a prompted list of action bar files. In this case, only one is around—the "standard" set:



You can select STANDARD by moving the cursor to it (since it's the only one, the cursor is there already) and pressing *Enter*.

Creating a Panel I.D.

Now your CUA panel has an action bar defined. The CUA panel we are defining doesn't have a panel I.D. displayed, but here is the menu you would use to create one:

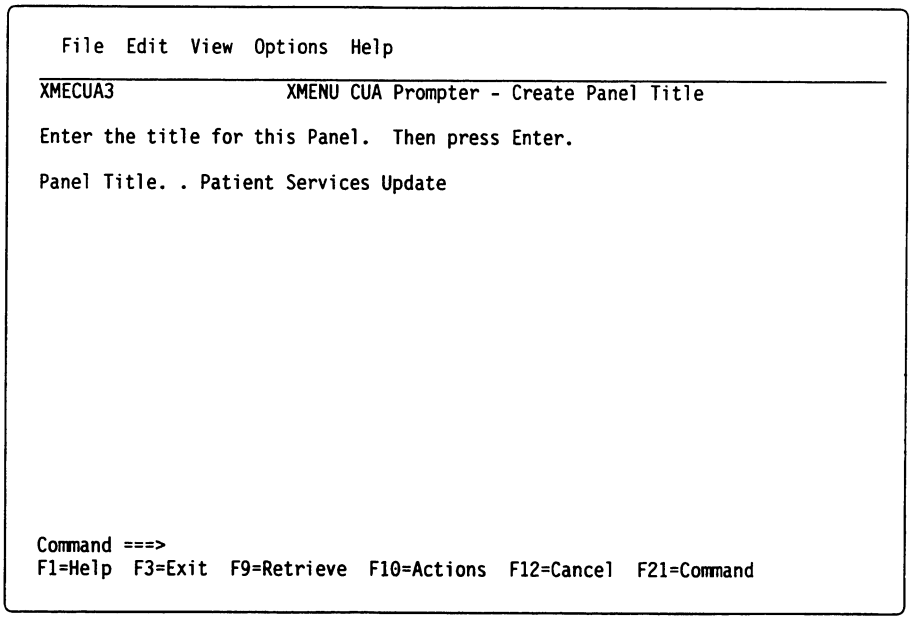
```
  File  Edit  View  Options  Help
 ------------------------------------------------------------------------
 XMECUA2                   XMENU CUA Prompter - Create Panel I.D.

 Enter the name of this Panel below.  Then press Enter.

 Panel I.D. . . . . . . .           1-10 characters














 Command ===>
 F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

You get to this menu by choosing *2* from the XMENUCUA main menu. Since we have no panel I.D. the entry field is blank. You can type a 1 to 10 character panel I.D. if you wish, although your panel would then differ from the example.

Press the *Cancel* PFK to return to the XMENUCUA main menu.

Creating the Panel Title

Now we will create the panel title. Select *3* from the XMENUCUA main menu and the following menu will be shown:



Type the panel title in the entry field and press *Enter*.

You do not have to worry about centering the title— XMENUCUA will do this for you automatically.

Press the *Cancel* PFK to return to the XMENUCUA main menu.

Creating an Information Area

We are now ready to create the body of the panel. We'll do this proceeding from top to bottom.

First, we'll create the line of informational text following the title.

Select *4* from the XMENUCUA main menu and the following menu will be shown:

```
  File  Edit  View  Options  Help
-------------------------------------------------------------------------------
XMECUA4          XMENU/CUA Prompter - Create Information Area

Enter the type of area you are generating, its placement, and its text.

Area Name. . : T1
Space after. . Y  (Y=space after this area, N=no space after this area)

Text . . . . .
Type patient name and select one from each group.  Then select an action.









Command ===>
F1=Help  F3=Exit  F7=Backward  F8=Forward  F10=Actions  F12=Cancel  F21=Command
```

The cursor is contained on the first line available for data entry. Type the line as shown in this example, then press *Enter.*

One or more lines can be typed. They will appear exactly as typed—no word wrapping occurs. Blank lines can be entered by leaving space between two typed lines.

The *Area name* is automatically assigned by XMENUCUA. It is used for XMENU menu field naming and allows you to reorder panel components later.

The *Space after* entry field allows you to specify whether a line of space is placed into the panel following this text. Since we want one, leave it as *Y.*

Press the *Cancel* PFK to return to the XMENUCUA main menu.

You can create several areas at the same time by using the *Forward* and *Backward* function keys. Forward will move you to the next area, or on to a new empty area. Backward will move you backward to a previous area, stopping at the first one created.

Creating an Entry Area

Now let's create the *Patient Name* entry area.

Select *7* from the XMENUCUA main menu and the following menu will be shown:

```
  File  Edit  View  Options  Help
XMECUA7       XMENU/CUA Prompter - Create an Entry Field or Protected Text
Enter spacing information, group heading, and entry field data.
Entry name . . : E1
Heading. . . . .
Field Prompt . . Patient Name
Input size . . . 32 (1-99)
Descriptive Text
Space after. . . Y  (N=no space after this line, Y=space after this line)
Protected Text . N  (N=entry field, Y=protected text, D=masked input)
Auto Tab Field . N  (N=not an Auto Tab field, Y=Auto Tab field)
Command ===>
F1=Help  F3=Exit  F7=Backward  F8=Forward  F10=Actions  F12=Cancel  F21=Command
```

The minimum data you need to type on this menu are the text preceding the input area, and the length of the input area. The preceding text is entered in the field *Field Prompt*. The size of the input area is specified in the field *Input size*.

Type the data as shown, then press *Enter*.

XMENUCUA automatically aligns input areas, and provides leader dots and protected field colon markers. You do not have to do this yourself.

Press the *Cancel* PFK to return to the XMENUCUA main menu.

While we're here, let's review the other menu areas:

- The *Entry name* is automatically assigned by XMENUCUA. It is used for XMENU menu field naming and allows you to reorder panel components later.
- The *Heading* field allows you to preface this entry area by a heading. If you type a heading it appears on the line directly above the entry area. You can leave a few spaces in the Field Prompt area to indent the preceding text below this heading if you wish.
- The *Descriptive Text* field is used to place descriptive text following the input area. You can also use leading blanks in this field if you want to align descriptive text after input areas of varying length.
- The *Space after* entry field allows you to specify whether a line of space is placed into the panel following this text. Since we want one, leave it as *Y*.

- The *Protected Text* entry field allows you to specify whether the input area is either for user input, displayed data only (data that cannot be modified by the end user), or masked user input (input the user can't see—such as for password entry).
- Finally, the *Auto Tab Field* entry field allows you to specify whether the cursor automatically advances to the next input field when this field is completely overtyped. You might want to use this for small, fixed length data elements to save keystrokes.

You can create several areas at the same time by using the *Forward* and *Backward* function keys. Forward will move you to the next area, or on to a new empty area. Backward will move you backward to a previous area, stopping at the first one created.

Creating a Selection List

Now let's create the two selection lists.

Select *5* from the XMENUCUA main menu and the following menu will be shown:

```
  File  Edit  View  Options  Help
-------------------------------------------------------------------------------
XMECUA5          XMENU/CUA Prompter - Create a Selection Field or List

Enter the type of list you are generating, group heading, and selections.

List Name. . : L1
List Type. . . S  (S=Single Selection, M=Multiple Selection)
Space after. . Y  (Y=space after this area, N=no space after this area)
Heading. . . .
Field Prompt . Status
                                                          Item 1  of 50
List items . . 1. Outstanding                           1-40 characters
               2. Completed
               3. Canceled
               4. Deleted




Command ===>
F1=Help  F3=Exit  F4=Up  F5=Down  F7=Bkwd  F8=Fwd  F10=Actions  F12=Cancel
```

The minimum data needed on this menu are the text preceding the selection list and the list items. The preceding text is entered in the field *Field Prompt*. The selection items are entered, one to a line, in the *List Items* area.

Type the data as shown, then press *Enter*.

Make sure that you type the leading numbers (*1.*, *2.*, etc.) also.

XMENUCUA automatically aligns selection lists, and provides leader dots on preceding text. You do not have to do this yourself.

Here's what else is on the menu:

- The *List name* is automatically assigned by XMENUCUA. It is used for XMENU menu field naming and allows you to reorder panel components later.
- The *List type* is used to specify whether this selection list is a single entry list (one input area), or a multiple entry list (one input area per line). In our example, both lists are single entry lists (the default).
- The *Heading* field allows you to preface this selection list by a heading. If you type a heading it appears on the line directly above the selection list. You can leave a few spaces in the Field Prompt area to indent the preceding text below this heading if you wish.
- The *Space after* entry field allows you to specify whether a line of space is placed into the panel following this text. Since we want one, leave it as *Y*.

You can create several areas at the same time by using the *Forward* and *Backward* function keys. Forward will move you to the next area, or on to a new empty area. Backward will move you backward to a previous area, stopping at the first one created.

Now, let's create the second selection list.

Press the *Forward* key and the following menu will be displayed:

```
  File  Edit  View  Options  Help
XMECUA5          XMENU/CUA Prompter - Create a Selection Field or List

Enter the type of list you are generating, group heading, and selections.

List Name. . : L2
List Type. . . S  (S=Single Selection, M=Multiple Selection)
Space after. . N  (Y=space after this area, N=no space after this area)
Heading. . . .
Field Prompt . Type of Order
                                                   Item 1  of 50
List items . . 1. Care & Treatment               1-40 characters
               2. Pharmacy
               3. Heart Station
               4. Physical Therapy
               5. Labs
               6. Dietary
               7. Radiology
               8. Other


Command ===>
F1=Help  F3=Exit  F4=Up  F5=Down  F7=Bkwd  F8=Fwd  F10=Actions  F12=Cancel
```

The minimum data needed on this menu are the text preceding the selection list and the list items. The preceding text is entered in the field *Field Prompt.* The selection items are entered, one to a line, in the *List Items* area.

Type the data as shown, then press *Enter.*

Press the *Cancel* PFK to return to the XMENUCUA main menu.

Both of the lists on this panel are short. If you need to enter a longer list, use the *Up* and *Down* keys to scroll the selection list data.

Creating the Command Line and Error Message Line

We've now created the main body of the menu. All that remain are the command line and the function keys.

Select *8* from the XMENUCUA main menu and the following menu will be shown:

```
  File  Edit  View  Options  Help
_____________________________________________________________________________
XMECUA8       XMENU CUA Prompter - Create Command and Error Message Areas

Enter the number of lines you want for the Command and/or Error Message Areas.
Then press Enter.

Command Area Size. . . . 1 (0-2)

Error Message Line . . . Y (Y=reserve a line for error message,N=no line)










Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

Here we can define the number of command entry lines and the presence or absence of a line reserved for error messages.

Type *1* for a single command line. XMENUCUA automatically generates the command line prefix (*Command* ===>).

If you specify *0* (the default) no command line is created.

Type the data as shown, then press *Enter*.

Press the *Cancel* PFK to return to the XMENUCUA main menu.

Creating the Function Key Area

When you select *Create a Function Key area*, the following will be displayed:

```
  File  Edit  View  Options  Help
XMECUA9              XMENU CUA Prompter - Create Function Key Area

Enter text for Function Keys below.  Leave undefined Function Keys blank.


     F1 = Help                      F13=
     F2 = Set 2                     F14=
     F3 = Exit                      F15=
     F4 = Prompt                    F16=
     F5 =                           F17=
     F6 =                           F18=
     F7 =                           F19=
     F8 =                           F20=
     F9 = Retrieve                  F21=
     F10= Actions                   F22=
     F11=                           F23=
     F12= Cancel                    F24=


Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

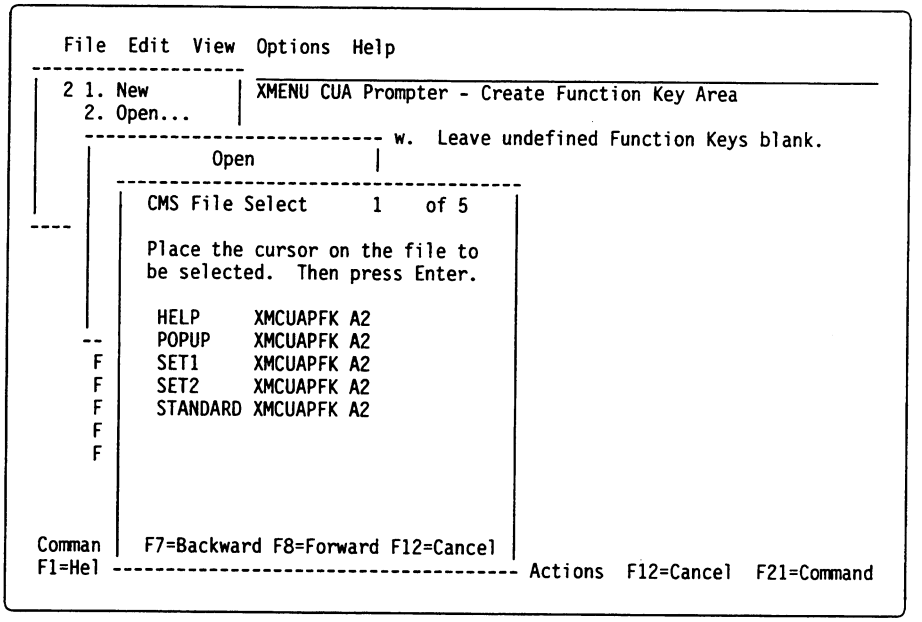
You type each function key definition in a single entry area. Be sure that you put the definition in the proper place. Type each function key definition, then press *Enter*. When you are done, the display should look like the one on this page.

When you have finished entering the function keys, press the *Cancel* key to return to the XMENUCUA main menu.

Do not worry about the length of function key items—XMENUCUA will automatically place them on the bottom of the panel so that they never wrap a line boundary.

We have provided a couple of standard function key lists. If you would like to load one into your CUA panel, select the XMENUCUA action bar item *File* and then *Open*.

If you use the *Prompt* key, you should get the following display:

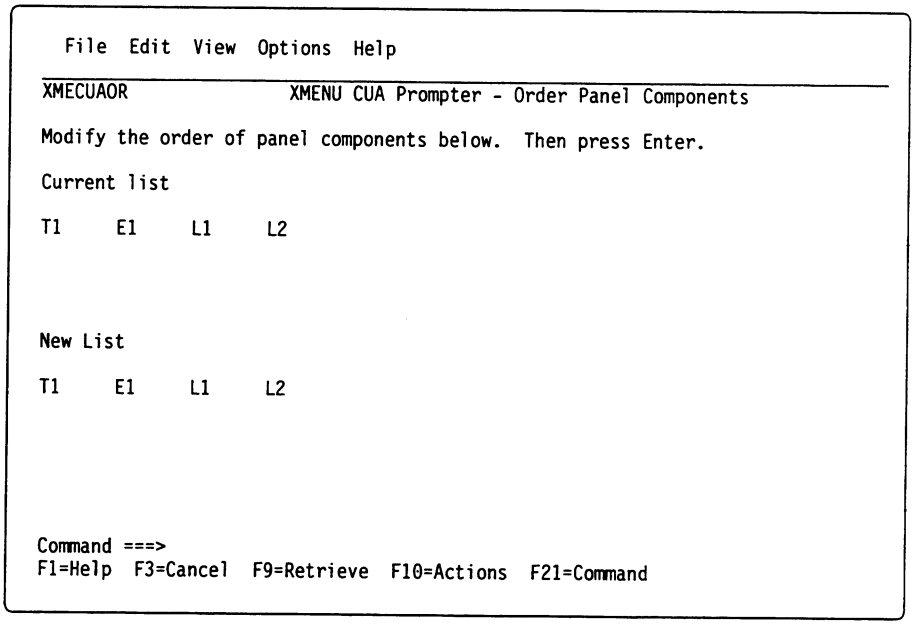


The set used in this menu is called *SET1*. You can load it by moving the cursor to it and by pressing *Enter*.

Rearranging the Areas of a CUA Panel

Now the entire CUA panel is created. We will pause to show you the area ordering menu. You won't need it now since you created the panel items in position order.

Select *10* to see this menu:



Two lists are shown—the top list is the current ordering, the bottom list can be changed to reorder the elements. The element names are those assigned by the Information, Entry area, and Selection list menus. These names begin with T, E, and S respectively.

For now, leave things as they are—just press the *Cancel* key to return to the main menu. Later, you might want to return to this menu to move things around to see what happens.

Displaying and Creating the XMENU menu

Now we're ready to see what the menu looks like.

Enter *11* from the main menu, and the following will appear:

```
  File  Edit  View  Options  Help
XMECUAXM                      XMENU CUA Prompter - Create XMENU Menu

Enter the name of the menu, and its size.  Then press Enter.
Use MAX for a size if you want the menu to take up the whole current screen
size.  Use MIN if you want the menu to be as small as possible, for example,
to be used as a pop-up menu.

Menu name. . . . . . . . PAGE159  1-8 characters

Lines in the menu. . . . 24  A number, MAX or MIN

Columns in the menu. . . 80  A number

Save the XMENU menu. . . Y Y for yes, N for no

Remain in XMEDIT . . . . N Y for yes, N for no



Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

You don't need to make any changes on this menu. When you press *Enter*, XMENUCUA will call XMEDIT to create an XMENU menu, display it so you can see what it looks like, then save it to disk.

You can change the following fields on the menu:

Menu name
: is the CMS filename of the menu to be created. It is the same as the name of the XMENUSAA file.

Lines in the menu
: is the number of lines in the menu to be created. You can specify a specific size, *MIN* to create a menu of minimum size (for example, to be used as a pop-up window), or *MAX* to create a menu as large as necessary, regardless of size.

After the menu is created, or if there is an error, this field is changed to the size actually used. You may want to run a menu twice—once using *MIN* to see how it looks in its smallest configuration, then once again to add lines of space to improve its layout. In general, you will want to make main menus 24 lines by 80 columns to be compatible with most 3270 terminals. These are the default line and column values.

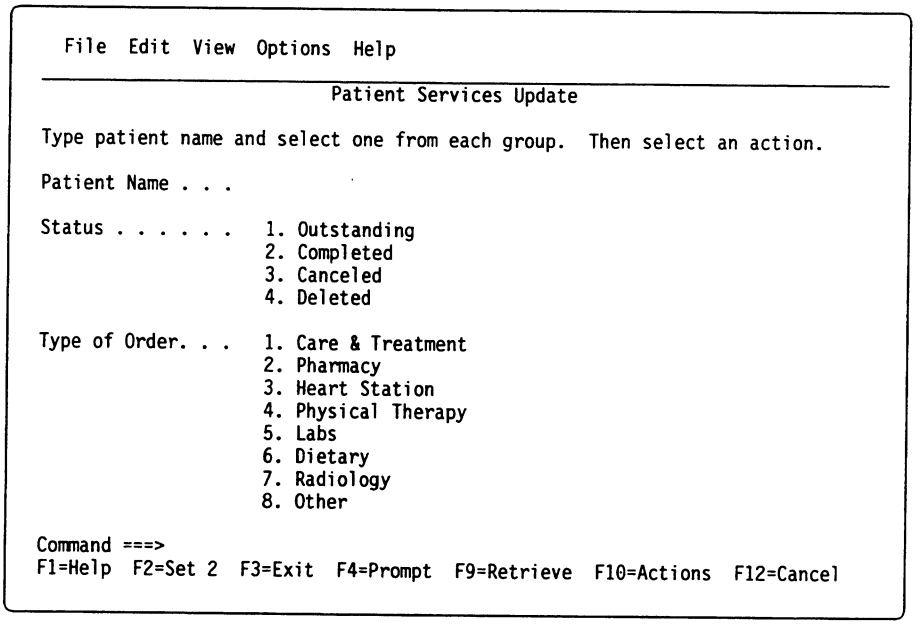
Columns in the menu
: is the number of columns in the menu to be created. You can specify a specific size. If either MIN or MAX were specified above, this field is ignored. After the menu is created, or if there is an error, this field is changed to the size actually used.

Save the XMENU menu specifies whether an XMENU menu is saved by XMEDIT.

Remain in XMEDIT specifies whether you want to stay in XMEDIT to make custom changes after the menu is created.

Note: Any changes made in XMEDIT are not saved by XMENUCUA so that if XMENUCUA recreates the menu again, these changes will be lost.

If you've followed us this far, you should now have a menu that looks like this:

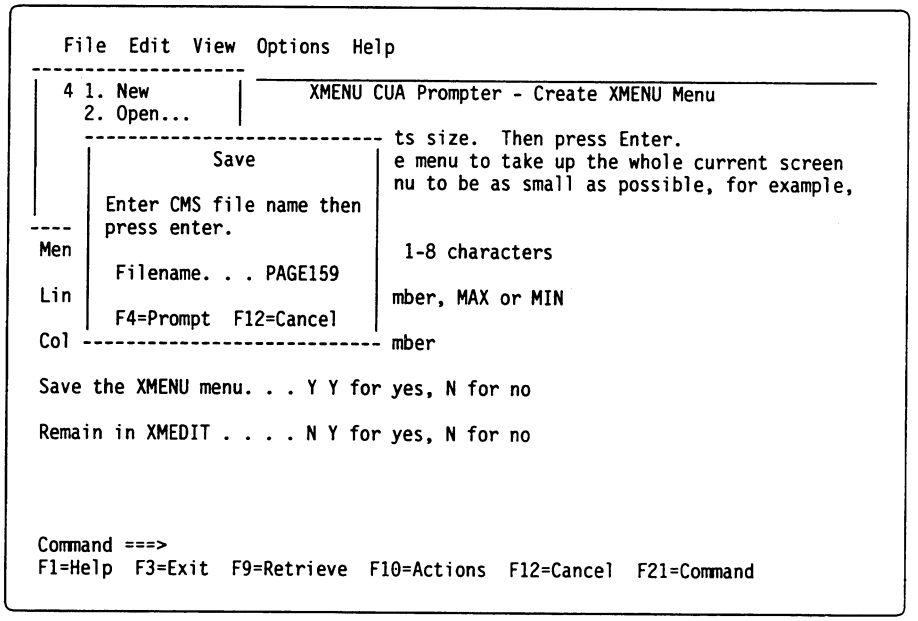


Press any key to return to the XMENU creation menu.

Saving the CUA Panel

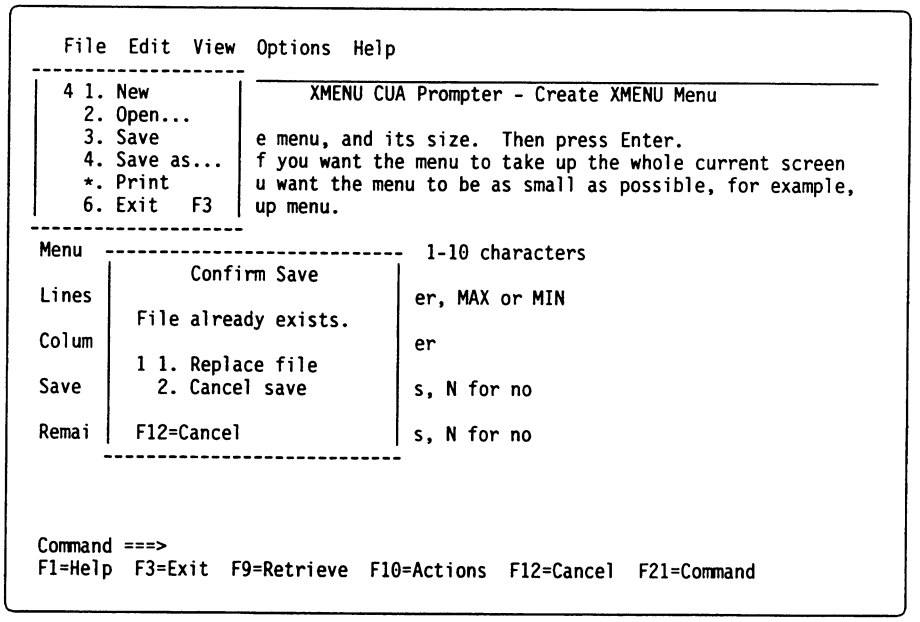
Even though the menu we've created was saved in the previous steps, we have not yet saved out CUA definition. The CUA definition is saved as ***<filename> XMENUSAA A***. This is a source-level file that is used by XTREE, and by XMENUCUA if changes are to be made to the CUA panel.

Select *File* from the action bar, and select *Save as*. You should see the following pop-up windows:



Press *Enter* to save the XMENUSAA file.

If the XMENUSAA file already exists, you will see this:



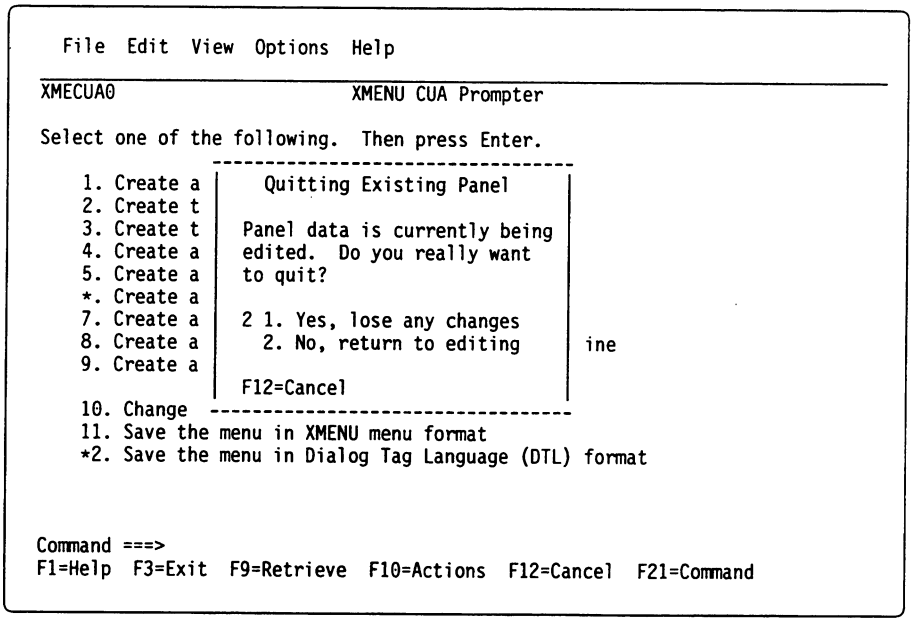
Press *Enter* to save the XMENUSAA file. If you don't want to overwrite this file, type *2* to cancel this save, then run another save, and select a different CMS filename.

You can also use the XMENUCUA file select list to select a saved filename.

Leaving XMENUCUA

If you want to leave XMENUCUA now, press the *Exit* key.

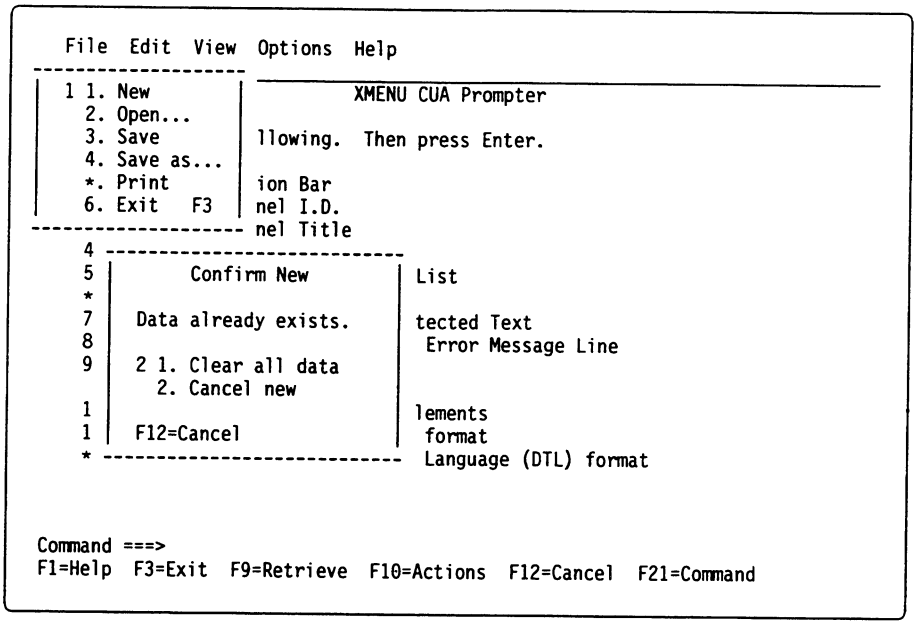
You should get the following:



Select *1* to quit, then press *Enter*. Otherwise, just press *Enter* to continue using XMENUCUA. Only changes made from the last ***SAVE*** will be lost.

If you want to start a new XMENUCUA CUA panel, select *File* from the action bar and select *New* from the pull-down window.

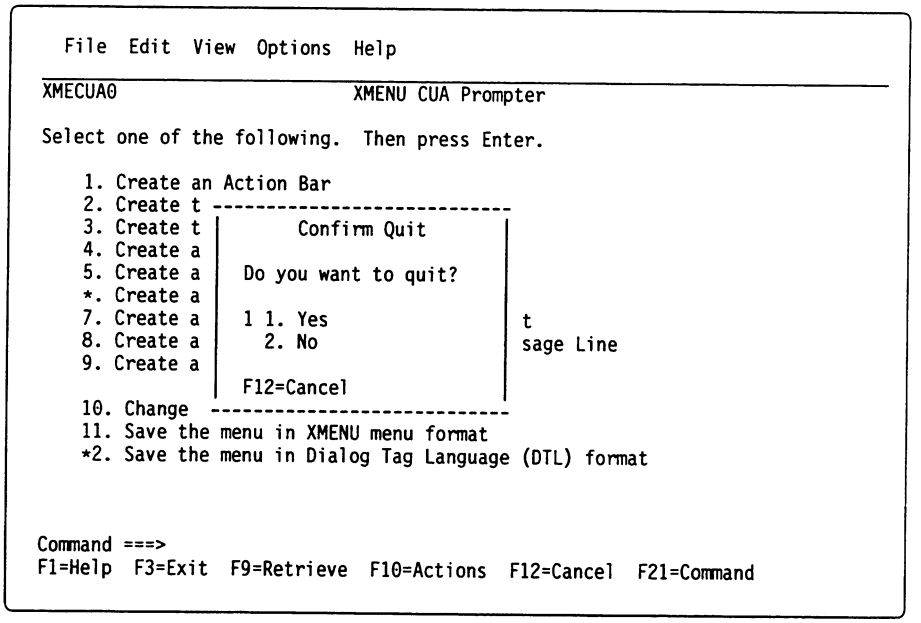
The following should appear:



This allows you to confirm that you want to start a fresh new editing session.

Select *1* to start a new panel, then press *Enter*. Otherwise, just press *Enter* to continue using the existing data.

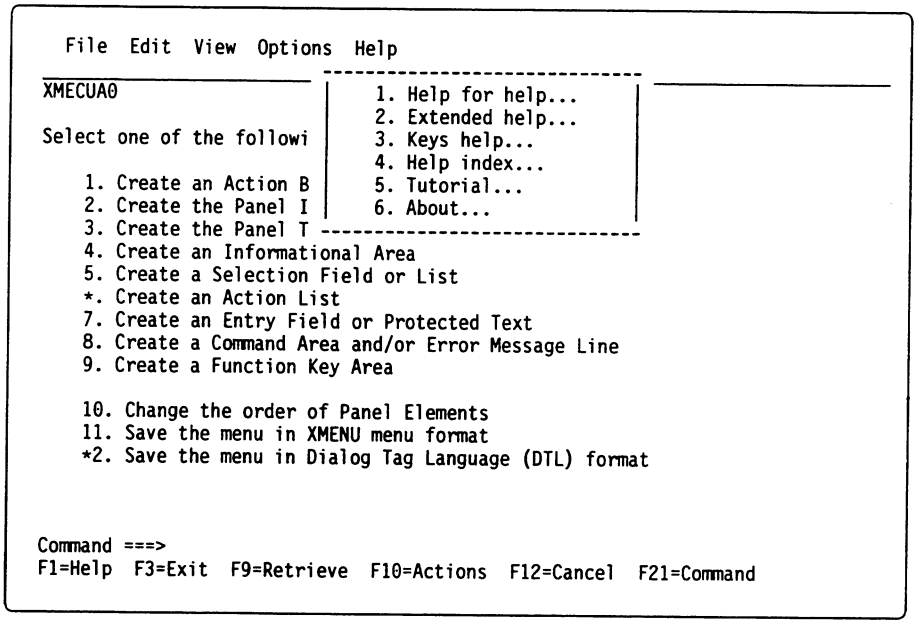
If you attempt to leave XMENUCUA before you have edited anything, you will see this display:



Just press *Enter* to exit XMENUCUA. Otherwise, select *2* to continue using XMENUCUA.

Displaying XMENUCUA Help

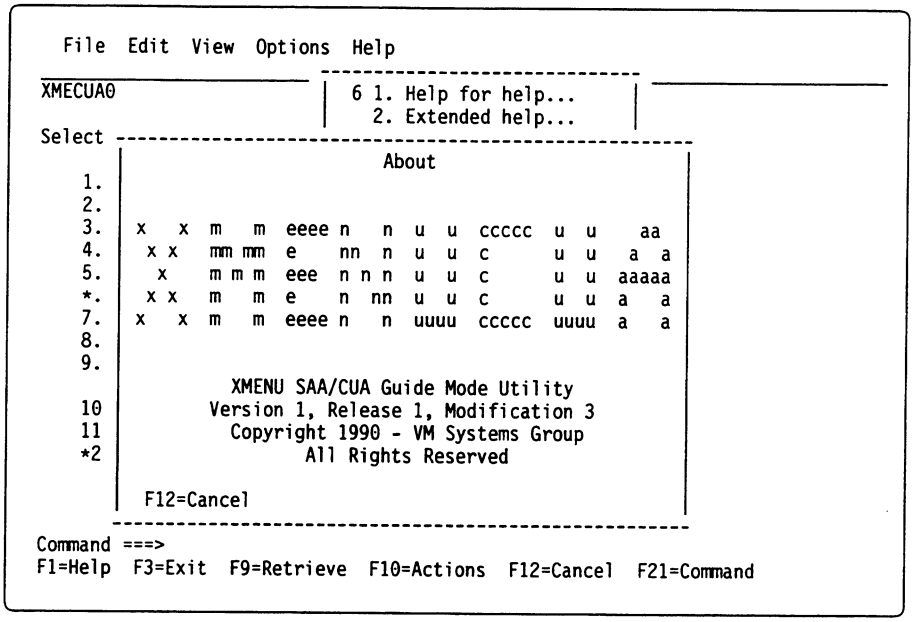
A certain amount of online help data is provided with XMENUCUA. If you select *Help* from the action bar you will get this pop-up menu:



You can choose the kind of help you want by selecting an item from this pull-down window.

You can use the *Cancel* key to return to XMENUCUA.

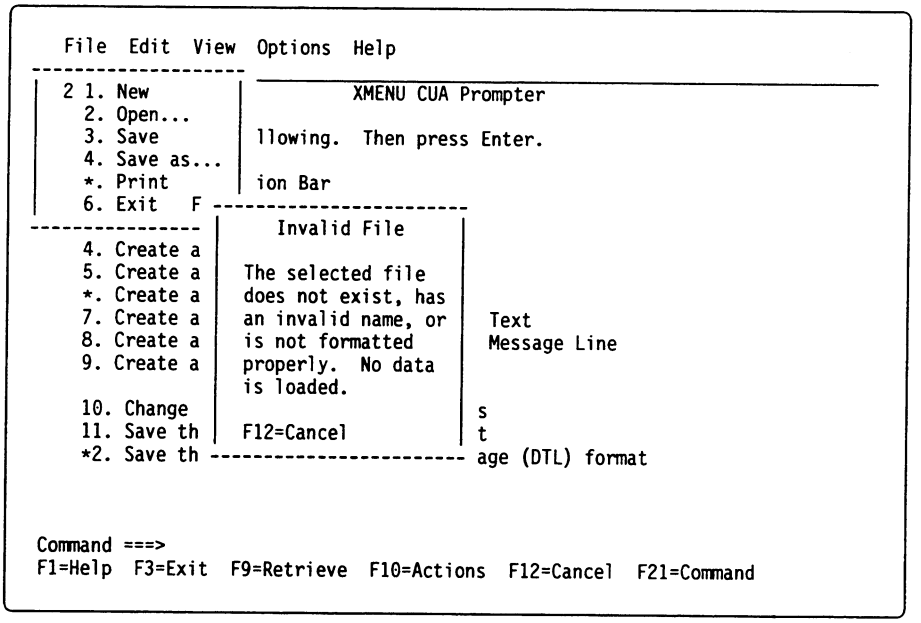
If you selected *6* from the Help pull-down window you will get this display:



You can choose the kind of help you want by selecting an item from this pop-up window.

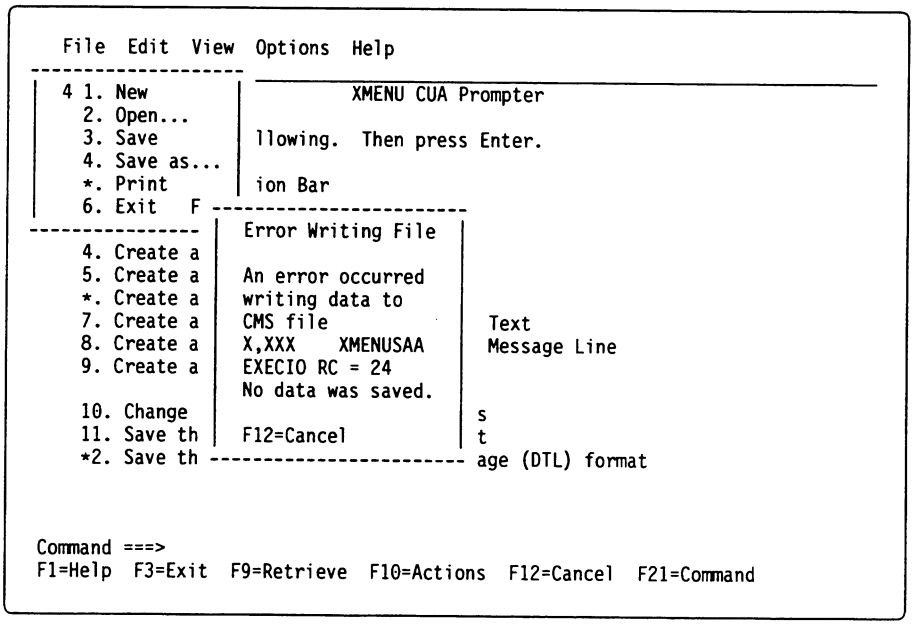
You can use the *Cancel* key to return to the Help pull-down window, and then XMENUCUA.

If you attempt to open an invalid or nonexistent CMS file, you will get an error message pop-up window:



The open process will be stopped. You can retry with a correct filename.

If XMENUCUA detects an error saving a CMS file, you will get an error message pop-up window:



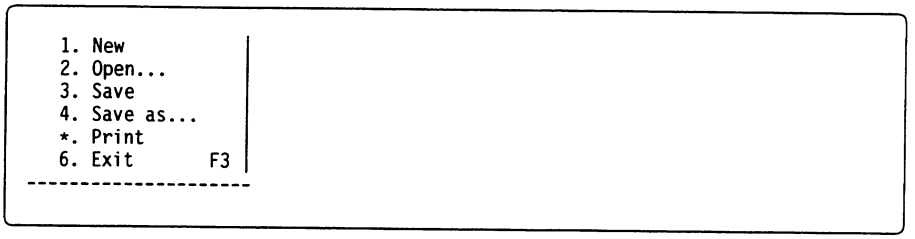
The save process will be stopped. You can use the CMS command line to access another disk, free up some space, or correct the incorrect CMS filename.

Creating a Pop-up/Pull-down Window

This chapter shows you how to create an SAA/CUA pop-up/pull-down window. It will also show you how to use XMENUCUA.

We assume that you've read through the first chapter already. Thus, we will skip loading, and saving the menu, and simply concentrate on the steps needed to create the window.

First of all, here's what we are going to create:



This menu is used as part of the standard CUA action bar when the *File* object is selected. Note that it has no action bar, no title, no command line, and no program function key definitions. It only has a single selection list.

First call XMENUCUA by entering:

```
XMENUCUA FILE
```

This will bring you into XMENUCUA and load the FILE example if it exists. You should first see the menu displayed in "Starting XMENUCUA" on page 1. Then, after you press *Enter*, you should see the menu displayed in "The XMENUCUA Main Menu" on page 4.

Entering the Selection List

Now let's create the selection list.

Select *5* from the XMENUCUA main menu and the following menu will be shown:

```
  File  Edit  View  Options  Help
 ──────────────────────────────────────────────────────────────────────
 XMECUA5          XMENU/CUA Prompter - Create a Selection Field or List

 Enter the type of list you are generating, group heading, and selections.

 List Name. . : L1
 List Type. . . S  (S=Single Selection, M=Multiple Selection)
 Space after. . N  (Y=space after this area, N=no space after this area)
 Heading. . . .
 Field Prompt .

                                                          Item 1  of 50
 List items . . 1. New                                  1-40 characters
                2. Open...
                3. Save
                4. Save as...
                5. Print
                6. Exit       F3



 Command ===>
 F1=Help  F3=Exit  F4=Up  F5=Down  F7=Bkwd  F8=Fwd  F10=Actions  F12=Cancel
```

The minimum data needed on this menu are the list items and the *Space after* entry field. The selection items are entered, one to a line, in the *List Items* area.

Type the data as shown, then press *Enter*.

Make sure that you type the leading numbers (*1.*, *2.*, etc.) also.

You need to change *Space after* to *N* so that an additional line isn't reserved following the selection list.

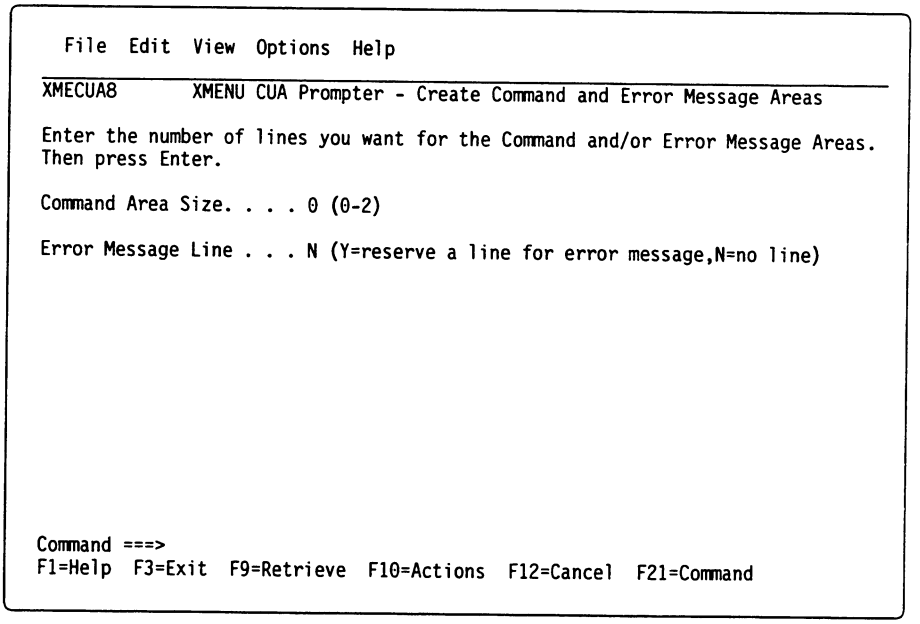
Press the *Cancel* PFK to return to the XMENUCUA main menu.

See "Creating a Selection List" on page 20 for more information about creating selection lists.

Disabling the Command Line and Error Message Line

We've now created the main body of the window. We now need to be sure that no command line or error message line is created.

Select *8* from the XMENUCUA main menu and the following menu will be shown:



Here we can define the number of command entry lines and the presence or absence of a line reserved for error messages.

Type *0* for no command line. Type *N* so that no error message line is reserved.

Type the data as shown, then press *Enter*.

Press the *Cancel* PFK to return to the XMENUCUA main menu.

Displaying and Creating the Pop-up/Pull-down Window

Now we're ready to see what the window looks like.

Enter *11* from the main menu, and the following will appear:

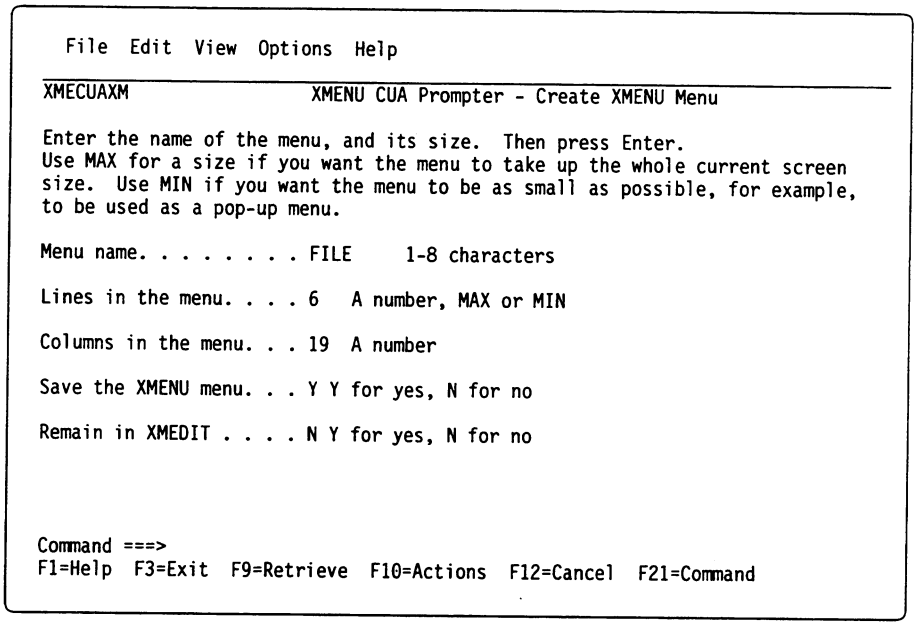
```
  File  Edit  View  Options  Help
-------------------------------------------------------------------------------
XMECUAXM                  XMENU CUA Prompter - Create XMENU Menu

Enter the name of the menu, and its size.  Then press Enter.
Use MAX for a size if you want the menu to take up the whole current screen
size.  Use MIN if you want the menu to be as small as possible, for example,
to be used as a pop-up menu.

Menu name. . . . . FILE     1-8 characters

Lines in the menu. . . . MIN A number, MAX or MIN

Columns in the menu. . .     A number

Save the XMENU menu. . . Y Y for yes, N for no

Remain in XMEDIT . . . . N Y for yes, N for no



Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

Change the number of menu lines to *MIN*. This will cause a menu of minimum size to be created.

When you press *Enter*, XMENUCUA will call XMEDIT to create an XMENU menu, display it so you can see what it looks like, then save it to disk.

See "Displaying and Creating the XMENU menu" on page 27 for more information about creating the XMENU menu.

After the menu is created, the following menu will appear:



We show you this to remind you that XMENUCUA changes your values from *MIN* to those actually used to create the menu. You may want to jot these numbers down so that you know how big your window is when you go on to creating your CUA compatible application.

"Saving the CUA Panel" on page 30 contains information about saving your XMENUCUA pop-up/pull-down window data to disk.

XMENUCUA Files

XMENUCUA can create a number of CMS files. Here's a list of them and what they are used for:

Filetype	Function
XMENUSAA	Contains a complete XMENUCUA CUA Panel definition. This file is used by XMENUCUA to make subsequent alterations to a CUA Panel definition. This file contains data that is not in the XMENU menu file— the XMENU menu only contains 3270 information, it does not contain the CUA definitions.
MENU	Contains an XMENU menu. This is the file used by an XMENU application to present and retrieve terminal data from your end users.
XMCUAABR	Contains an XMENUCUA Action Bar definition.
XMCUAINF	Contains XMENUCUA Information area definitions.
XMCUASEL	Contains XMENUCUA Selection list definition.
XMCUAENT	Contains XMENUCUA Entry area definitions.
XMCUAPFK	Contains XMENUCUA Function Key definitions.

XMENUCUA comes with a few CUA panel, pop-up/pull-down windows, action bar prototypes, and function key definitions. These can be used when creating your own CUA panels to save typing time.

XMENUCUA Hints

This chapter gives a handful of random hints, limitations, and suggestions for using XMENUCUA. We're very interested in your feedback—we'll want to include any interesting uses you find for XMENUCUA or clever usage techniques.

The information contained in this chapter is not in any particular order.

Action Lists

XMEDIT does not support the creation of action lists at this time. We did this because Programmable Workstations do not support this form of CUA panel. We felt that by not supporting action lists, we would save you a later additional conversion effort if you attempt to move your applications to workstations.

If you really want to create action lists, use XMENUCUA to create all but the middle of your menu, then use XMEDIT to create the columnar format needed by the action list.

You can also simulate single character entry action lists by creating a multiple selection list and using the header line as column headings.

Pop-up/pull-down windows

1. Be sure to specify *MIN* the first time you create a window to make it as small as possible. After this, you may want to massage the values to make the window more pleasing to the eye.
2. Don't forget to drop the last line following any selection area or entry area, as well as the error message line.

Multiple line entry areas

Since the CUA Basic Design Guide does not define multiple line entry areas, XMENUCUA does not provide for their creation.

If you want a multiple line entry area, use XMENUCUA to create a single line area. Follow it by a dummy information area to reserve enough lines of space, then change the input area size using the XMENU XMEDIT editor.

Scrollable area

XMENU has no provision for scrolling the middle of a menu while keeping the top and bottom fixed.

There are several ways around this:

1. If you just want to scroll a list, use the field names to move data into the selection choices at run time.
2. If you want to scroll the entire center of the menu, create a series of menus with the appropriate parts displayed.
3. Create different sized menus for different sized terminals so that a person using a 43 line terminal will see more at once than someone using a 24 line terminal.

Scrolling information

You can use XMEDIT to create a field to display scrolling information. You could also use an information area to display this data.

Panel I.D.

You can "turn off" panel I.D.'s by using XMENU at run time to change the PANELID field to dark intensity, thus making it invisible.

Function Key Area

You can "turn off" the function key area by using XMENU at run time to change the PFKLINE field to dark intensity, thus making it invisible.

You could also create two menus, one containing the keys, another leaving them off. This method gives you an extra line to work with if function key displays are disabled.

Prompting

XMENUCUA comes with and uses an EXEC called XMECUAFL to display a list of CMS files for a user to select. You can use this in your EXEC applications if you wish. Its usage and parameters are defined in the EXEC source.

XMENU Hierarchical Menu Facility (XTREE)

Introduction to Menu Hierarchies

This chapter describes the concepts of menu hierarchies and introduces you to some of our terminology. Whenever a phrase is first introduced, it is printed *in italics—just like these.*

What are Menu Hierarchies?

XMENU has always allowed you to create individual menus, but it was up to you to program the code necessary to display them in any sort of ordered manner.

With XTREE, you can define the relationships among menus. You can specify what actions on a menu cause another menu to be displayed, or some program action to be taken.

When you use the XMENU menu hierarchy support, you no longer need to program the control flow of menu displays.

Since a menu hierarchy can be pictured as an organizational chart or *tree*, menu hierarchies are sometimes referred to as *menu trees*. In fact, most of the XMENU menu hierarchy utility names contain the word tree.

All hierarchical data is kept in CMS files separate from the XMENU menus themselves. This allows you to use menus in more than one hierarchy or application, as well as use a given menu more than once in a hierarchy. Menus are only tied to a hierarchy within the MENUTREE control file.

How do I control a menu hierarchy?

XTREE lets you control the order and display of menus based on real-time user input.

When a given menu is displayed, and you decide to display a new menu, you can do one of three things:

1. You can specify that an action causes the display of a new menu that is *called.* Calling a menu means that a new menu is displayed, but at some time in the future the original menu will be redisplayed. This allows you to specify *hierarchies* of menus—that is, each *child menu* has a *parent menu.* Think of called menus as subroutines—they (almost) always return to their caller.
2. You can specify that an action causes a *return* to the previously displayed menu. Think of this as a return from a called subroutine.
3. You can specify that an action causes the display of a new menu replacing the existing menu. In this case, no return is made to the original menu. We use the term *goto* for this support because the first menu "goes to" the second menu. The term goto comes from programming parlance.

These three actions give you the ability to present your end users with an ordered list of menus, while at the same time allowing a quick transversal of the menu tree.

Windows, be they *pull-downs* (from the action bar), or *pop-ups* (displayed "on top of" existing menu(s)) are handled the same way as any other menu—they are completely supported by XTREE.

At any point in the tree, you can request that some program action takes place, for example, a CMS routine or an EXEC is called. The data from any displayed menu's named fields can be used as part of a command.

Since there is no restriction on which menu is displayed at any *node* in the tree (in fact, the same menu could be used at different parts of the hierarchy), XTREE makes you name each node, or hierarchy *section*. All call and goto commands refer to *section names*.

What kind of user input is supported?

XTREE supports several different types of user input:

1. *Action Bar*—selecting an action bar item can cause the menu hierarchy to execute a command—typically to display a pull-down menu for that action.
2. *Selection Lists*—entering a numeric value, or the positioning of the cursor onto a selection list item can cause the menu hierarchy to execute a command—often causing the display of a new menu of more specific choices.
3. *Command line* entry can be used to issue CP and CMS commands, as well as cause new displays to be shown. Command input can also be used to quickly move from one menu to another—transversing the tree, or to allow experienced users to enter commands rather than go through many input screens.
4. *Function key input* can also be defined to issue commands or display new menus.

Each action type, action, and resulting command can be individually specified.

How do I handle user input?

Just like when you use the rest of XMENU, assign each menu input and output field a *field name*. When the menu hierarchy is run, any user input is automatically assigned to a REXX variable with the same name as its input field. This data can be transferred to programs in many ways:

1. Data can be passed as part of a called command's command line.
2. Data can be used within a converted menu hierarchy REXX EXEC.
3. Data can be passed to called programs and EXECs via XMENU subroutines, GLOBALVs or KVARXFER.

You can use MENUEXEC subcommands to force data validation while the menu hierarchy is running. For example, you could specify that a field be numeric, or only allow a range of values.

You can also use special XTREE commands to place data onto menus before they are displayed, and retrieve changed data after the menu is displayed.

What are XTREE Special Field Names?

When you run XTREE, or a converted hierarchy EXEC, XMENU automatically fills appropriately named menu fields with useful information, such as date, time, your userid, and so on. All you do to display this data is to create fields on your menu with the proper names. A list of these fields and their assigned contents can be found in "Menu Hierarchy Special Field Names" on page 109.

How does XTREE work with SAA/CUA compatible panels?

XTREE, the menu hierarchy prompter, was specially designed to support XMENU menus created with XMENUCUA, the XMENU CUA prompter.

Whenever you define a new menu tree section, you must specify the menu to be displayed. If an XMENUSAA file (produced by XMENUCUA) exists for this menu, it is automatically read, and information describing the kinds of action this menu supports are filled into XTREE input fields.

XMENUCUA defined action bars, function keys, and selection lists are directly supported.

A Simple Example

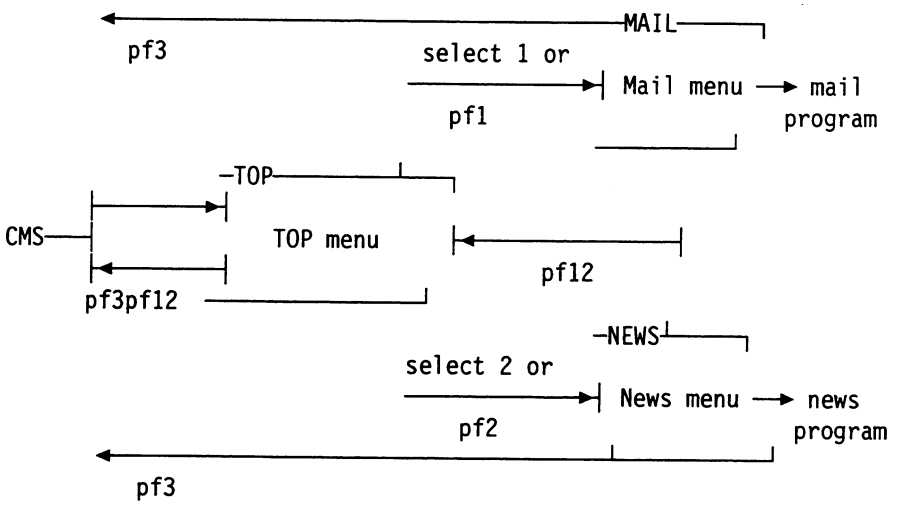
Suppose we had a *top*, or initial menu that had a two line selection list, such as this:

```
            Top Menu
Select one of the following, then press Enter.
  1. Display mail.
  2. Display news.
PF1=Display mail  PF2=Display news  PF3=Exit
```

Let's call this menu *topmenu*. We won't show the two menus that this displays, but let's call them *mailmenu* and *newsmenu*.

- We want the mail menu to be displayed if the user either selects item 1, or presses function key 1.
- We want the news menu to be displayed if the user either selects item 2, or presses function key 2.
- If the user presses function key 12 while in either the mail or news menu, we want to return to the top menu.
- If the user presses function key 3 while in the mail or news menus, the application should exit
- If the user presses function key 3 or 12 while in the top menu, the application should exit.

Pictorially, our program flow looks like this:



Our MENUTREE control file should look like this:

```
TOP:    topmenu ( map
        selection 1 slc1 call mail
        selection 2 slc2 call news
        pf1 call mail
        pf2 call news
        pf3 exit
        pf12 exit
MAIL:   mailmenu ( map
        pf3 exit
        pf12 return
NEWS:   newsmenu ( map
        pf3 exit
        pf12 return
```

The upper-cased words (TOP, MAIL, and NEWS) in both the picture and the control file are the *section names* for this menu hierarchy (They don't have to be upper case, we just did this to make them stand out). TOP is the first section displayed. The colon following the name identifies it as a section name.

The words following the section name (topmenu, newsmenu, and mailmenu) are the names of the XMENU menus displayed at that section. The data following the menu names are MENUEXEC parameters— we've passed "map" to request that PF13-24 be treated as PF01-PF12.

The *selection* commands tell XMENU that if a certain selection value is entered (1 or 2), or if the cursor is placed within a selection list entry (signified by the XMENU menu field names of those fields—S1C1 and S1C2), then a new menu tree section is displayed (mail or news).

The *pf* entries direct that if a certain program function key is pressed, action is taken. Undefined function keys are ignored.

For brevity, we've left out any coding relating to having this application do anything other than display the menus.

The MENUTREE file we've just shown could be created using XEDIT, or can be built interactively by using XTREE.

How do I create a Menu Hierarchy?

There are only a few steps to creating a menu hierarchy:

1. Design your application (at least in your head).
2. Create your menus. If you are creating a brand new application, we recommend that you consider using XMENUCUA (the XMENU SAA/CUA prompter) to ensure that your application is compatible with IBM's System Application Architecture guidelines.

   You can use either XMEDIT (the XMENU menu editor) to create your menus, or XMENUCUA. XMEDIT is more flexible, XMENUCUA forces you to adhere to SAA/CUA rules.
3. Create your menu hierarchy. This can be done in two ways.

   You can use the XTREE utility. This utility leads you through a collection of SAA compatible menus, and allows you to build and test your hierarchy. We show you around XTREE in "Interactively Creating a Menu Hierarchy" on page 55.

   Alternatively, you can create your hierarchy file directly. This is usually done using XEDIT. A hierarchy file has a relatively simple set of commands. These are described in "Menu Hierarchy File Format and Commands" on page 95.

   If you edit your own hierarchy file, you can still use XTREE to test it. See "Testing your Menu Hierarchy" on page 82 for more details.
4. Use MENUTREE to run your application. MENUTREE loads your control file, then executes all structures except for ICALL statements, since we have not yet built an executable EXEC. You can find out more about running MENUTREE in "The MENUTREE Program" on page 93.
5. Once your application is stable, you can use XTRECOMP to convert it from a control file into a self contained REXX EXEC. This EXEC has all of the functions of MENUTREE built right into it. "XTRECOMP—The MENUTREE Compiler" on page 105 describes XTRECOMP. If you have a REXX compiler, you can compile this REXX EXEC for even better performance.

What else can I do with menu hierarchies?

This chapter only gives a small overview of the capabilities of the XMENU menu hierarchy system. Here are some other things you can do:

- Specify global function key and commands that are valid for all sections.
- Specify the processing order of action bar, function key, and user input.
- Specify default selections.
- Customize error messages displayed when a user makes an invalid function key, selection choice, or command entry.
- Specify commands to be issued the first time a menu is displayed.
- Specify commands to be issued before each menu is displayed.

- Specify commands to be issued after each menu is displayed.
- Specify and code internal REXX subroutines to improve application performance.
- Automatically display system data such as the date, time, userid, menu section name, and so on.
- Use the entire range of MENUEXEC functions, including data validation, saving field contents across application calls, and so on.

Interactively Creating a Menu Hierarchy

This chapter shows you a sample XTREE session. This will show you how to interactively create an XMENU menu hierarchy. It will also show you how to use XTREE.

In this chapter, we are going to show example XTREE menus based on editing the supplied sample *XTRSAMPL*. As menu hierarchy files can get rather large, we are not going to go through the entire process of creating this file. Instead we will show and explain to you the displays you will interact with.

You can also code your own MENUTREE files. See "Menu Hierarchy File Format and Commands" on page 95 for information on the content of these files.

To make the information clearer, we have placed each screen on the top of a new page, followed by a description of its contents.

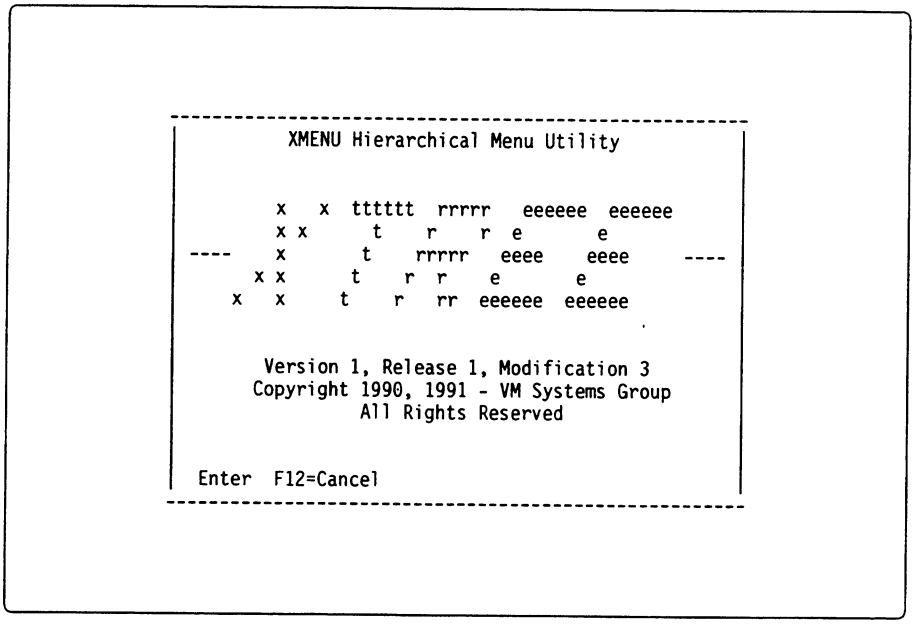
Starting XTREE

To run XTREE, type:

```
XTREE [filename] [( NOLogo]
```

If a filename is supplied, this file is automatically loaded. If *NOLOGO* is specified, the initial copyright menu is not displayed.

Here is what you will see if you simply type *XTREE*:



Press *Enter* to continue running XTREE. Press *PF12* to exit XTREE.

The XTREE Main Menu

Once XTREE begins, after the optional copyright menu, the following screen is displayed:

```
  File  Edit  View  Options  Help
XMETRE0                          MENUTREE Prompter Main Menu

Select one of the following.  Then press Enter.

   1. Select local menu name and menu options.
   2. Define local menu parameters.
   3. Define local action bar commands.
   4. Define local function key commands.
   5. Define local selection list commands.
   6. Define local command synonyms.
   7. Define local MENUEXEC subcommands.
   8. Define local error messages.
   9. Define global function key commands.
   10. Define global command synonyms.
   11. Test your MENUTREE application.

Current local section. . . . TOP      1-8 character name


Command ===>
F1=Help  F3=Exit  F7=Back  F8=Fwd  F9=Retrieve  F10=Actions  F12=Cancel
```

All hierarchy creation is performed by selecting items from this menu. Each portion of a hierarchy is defined by choosing the number that corresponds to the area you are creating or editing. For example, choosing *1* would display another menu allowing you to define a hierarchy section's menu and menu parameters.

Note that this menu, and all XTREE menus are in SAA/CUA format, and are controlled using SAA/CUA interface rules.

You can type a number to the left of the first item in the selection list and press *Enter* to go to that XTREE function. You can also position the cursor beside or on a selection and press *Enter* to go to that function.

You can move the cursor to an item on the Action Bar at the top of the menu to perform functions defined by that item.

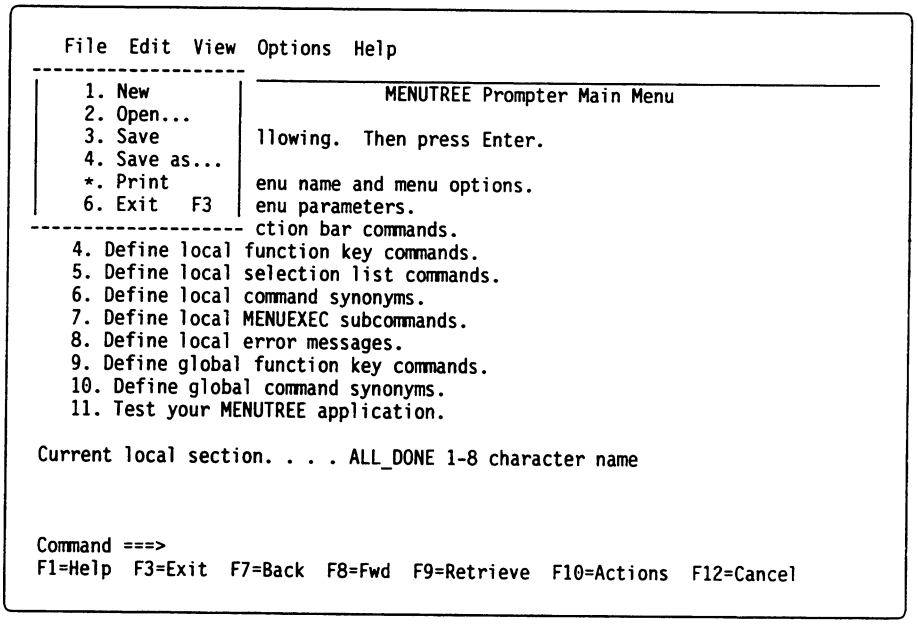
Finally, you can use the command line at the bottom of the menu to enter CP, CMS, or a small number of XTREE commands.

XTREE maintains a ten command retrieve stack, so that you can recall and reissue previously entered commands. The "?" subcommand will retrieve previous commands. The "=" subcommand will reexecute the last command issued.

Notice that a number of XTREE commands can be invoked by pressing Program Function Keys (PFK). These are shown at the bottom of the menu. In this document we refer to these keys as PFKeys or function keys. We also on occasion refer to them by name, for example, we may ask you to press the *Action* key.

The XTREE Action Bar

To show you how the Action Bar works, place the cursor on the word *File* or press the *Actions* PFK. Then press *Enter*. Here is what you should see:



In keeping with SAA/CUA protocols, a pull-down window appears showing you the choices available under *File*.

There are several choices:

New allows you to begin creating a new menu hierarchy.

Open allows you to open an existing menu hierarchy.

Save allows you to save this menu hierarchy section under the current filename.

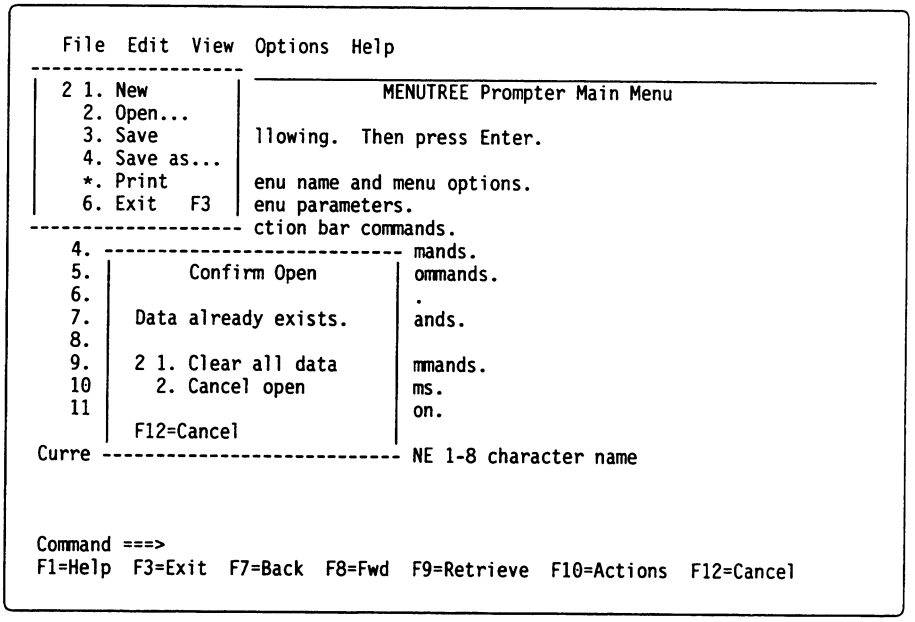
Save as... allows you to save this menu hierarchy section under the a new filename.

Print currently has no XTREE function.

Exit allows you to exit XTREE.

Opening an XTREE File

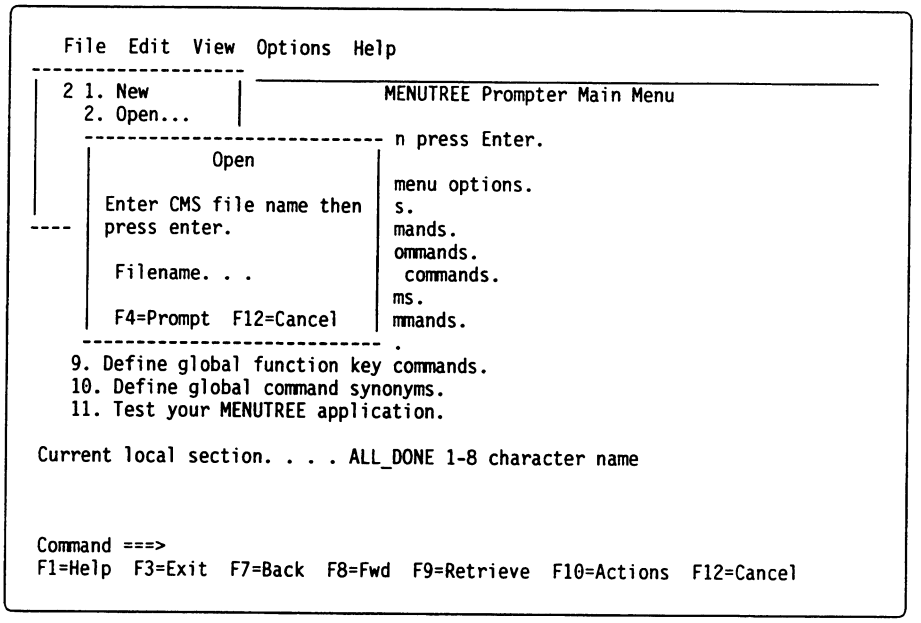
To show you how an existing file is opened, type *2*, then press *Enter*. If you were already editing a hierarchy, the following additional pop-up window is displayed:



Note that the default, that is if you were to simply press *Enter*, is to cancel the open (so that you don't lose your existing work).

If you change the selection to *1*, the open process would continue.

If you are creating a new file, or if you selected item 1 from the Open Confirmation menu, you will see the following:

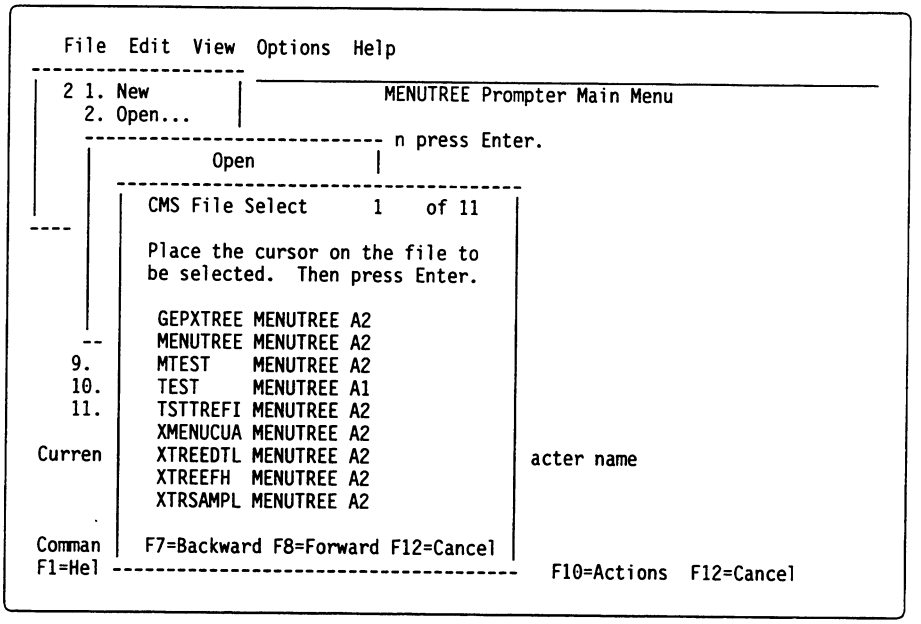


You can do one of three things at this point:

1. You can cancel the Open processing by pressing PF12.
2. You can type a valid CMS filename and press *Enter*.
3. You can display a list of existing CMS MENUTREE files by pressing the Prompt PFK.

If you use file prompting, you can use the SAA/CUA wildcard characters, * and *?*.

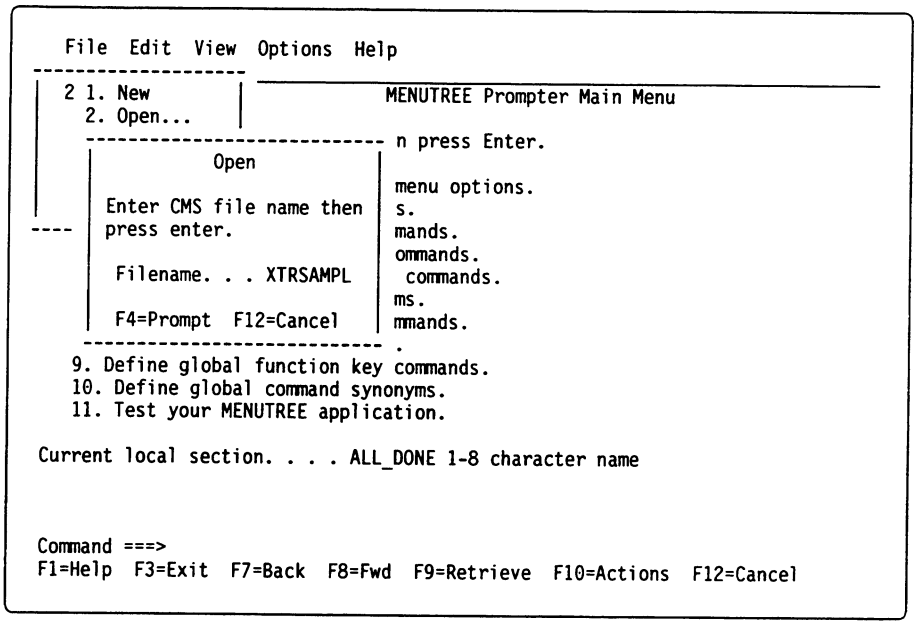
If you press the Prompt PFK, you will see yet another pop-up window such as the following:



At this point, you can position the cursor on any displayed file and press *Enter*. This will select the file, and move it into the previous pop-up window.

As an example, position the cursor on *XTRSAMPL* and press *Enter*.

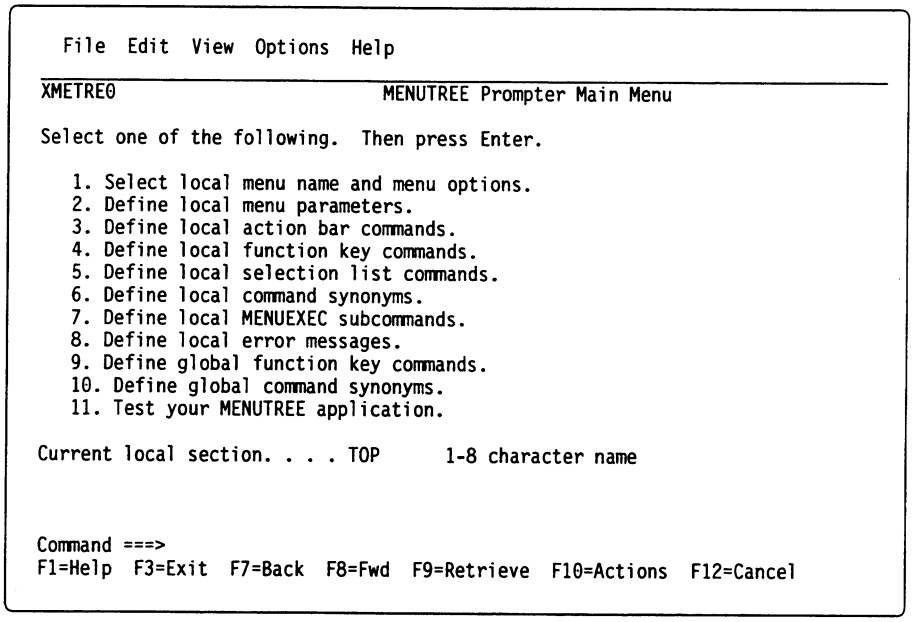
If you either selected XTRSAMPL from the CMS file menu or typed it in yourself, you should get the following display:



Press *Enter* and the menu hierarchy file XTRSAMPL will be opened for editing.

This file contains the example we are working with in this chapter. You can clear it out by selecting *New* from the File pull-down window now if you want to type the example in yourself.

Once you have opened the file, all pop-up windows are removed and you return to this menu:

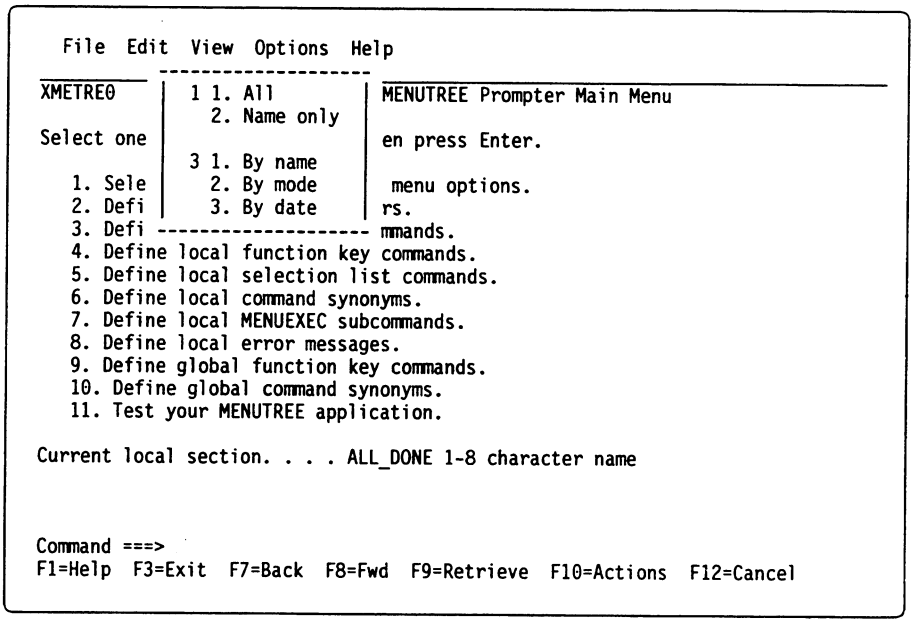


It looks exactly the same—you won't notice loaded data until you go to menu hierarchy sections.

Displaying All File Information

The last couple of pages showed you how to open a file. You can alter the display you get when you ask for a prompted list of CMS files. This is done using the *View* action bar command.

If you place the cursor on View and press enter, you will get the following display:



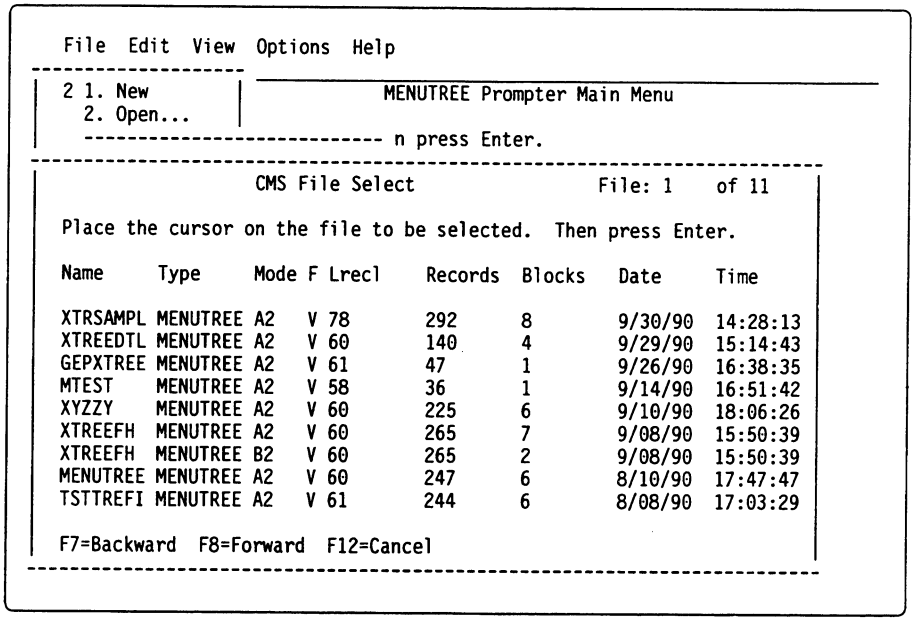
Normally, only the CMS file name is displayed, and the names are sorted in ascending name order.

You can change either the sorting order (ordered by name, ordered by disk mode, or ordered by descending date/time).

You can also display full file information by selecting *All.*

In the sample above we've selected all file information, sorted by descending date.

If you've changed the viewing options as we've just done, you will get a display like the following the next time you ask for a prompted list of files:



Adding, Deleting, and Changing Sections

When we first entered XTREE, the first section (named TOP) was displayed by default. XTREE always names the topmost section *TOP*, so you will see this even if you are starting a new hierarchy. If you want to change this default name, enter a new name, then scroll back and delete section TOP. Your new name will become the top section.

You can press PF8 or PF20 to scroll forward through the list of sections. You can press PF7 or PF19 to scroll backward through the list of sections.

Our example XTRSAMPL has many sections—you can use these scroll keys to look through the entire list.

You can also type in the name of a section in the area called *Current local section.* If the section exists, you go to that section. If the section doesn't exist, a new section is created.

If you want to delete a section, clear out the section name, and the following menu will appear:

```
  File  Edit  View  Options  Help
XMETRE0                          MENUTREE Prompter Main Menu

Select one of the following.  Then press Enter.
    ------------------------------------
 1. |  Confirm Clearing MENUTREE Section | .
 2. |                                    |
 3. |  Please confirm your request that  |
 4. |  data for this section be cleared. |
 5. |                                    |
 6. |  1 1. Do not clear this section    |
 7. |    2. Clear this section           |
 8. |                                    |
 9. |  F12=Cancel                        |
10 ------------------------------------
11. Test your MENUTREE application.

Current local section. . . . TOP      1-8 character name


Command ===>
F1=Help  F3=Exit  F7=Back  F8=Fwd  F9=Retrieve  F10=Actions  F12=Cancel
```

Enter *2* to clear this section from the hierarchy, *1* to leave it as is (the default).

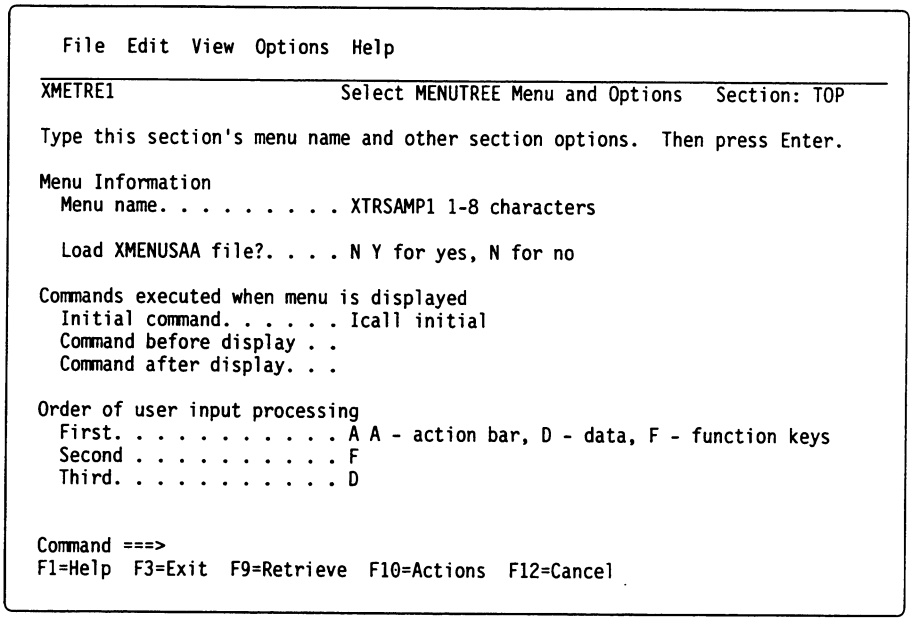
You need to enter data on a number of XTREE menus for each section in your menu hierarchy. You can enter data section-by-section, or XTREE menu-by-menu.

We've found it easier to enter an entire section's worth of data at one time, since you need to return to the XTREE main menu to change sections.

Specifying Menu Information

Each hierarchy section must display a menu, therefore you must name the menu to be displayed.

From the main menu, selection item *1.* and the following menu will be shown:



Note that all XTREE menus have the section name displayed as part of the title. This way, you can see what section you are working on.

Type the CMS filename of the menu to be displayed in the *Menu name* field.

Loading data from XMENUSAA files

You can request that data from an existing XMENUSAA file (the output generated by XMENUCUA) be loaded at this time by entering *Y* into the field labelled *Load XMENUSAA file*. You should only do this once for each section— either the first time you create the menu hierarchy, or if you modify the XMENUSAA file. Once the file is loaded, XTREE changes your input to *N*.

Specifying initial, before, and after commands

You can specify that a command be issued once before the menu is first displayed by typing it into the *Initial command* field. You can use this to initialize this section, or, if specified on the top section, to initialize the entire application.

You can specify that a command be issued every time before the menu is displayed by typing it into the *Command before display* field. You can use this command to fill in user input fields and output display data.

You can specify that a command be issued every time after the menu is displayed by typing it into the *Command after display* field. You can use this command to process user input fields before calling a processing application. This can be thought of as an action with the same priority as the data-level of processing discussed next. Assuming

that the order of processing specifies that action-bar items and PF keys have priority over data, the following is true:

- 'After' processing will ***not*** be invoked:
  1. When the cursor is positioned on a valid action-bar item and the user presses Enter
  2. When a valid PF key is pressed
  3. When a command is issued on the command line
- 'After' processing ***will*** be invoked:
  1. Before any item selected in a selection list is processed after the user presses Enter
  2. When the cursor is positioned in the action-bar area, but not on a valid action, and the user presses Enter
  3. When an undefined PF key is pressed
  4. When a null command is entered

The exception to this rule is when the order of processing specifies that data is to take precedence over action-bar items and PF keys. In this case:

- 'After' processing will ***not*** be invoked:
  1. When a command is issued on the command line
- 'After' processing ***will*** be invoked:
  1. Before any other action after the user presses Enter or any PF or PA key, i.e. any 3270 AID key

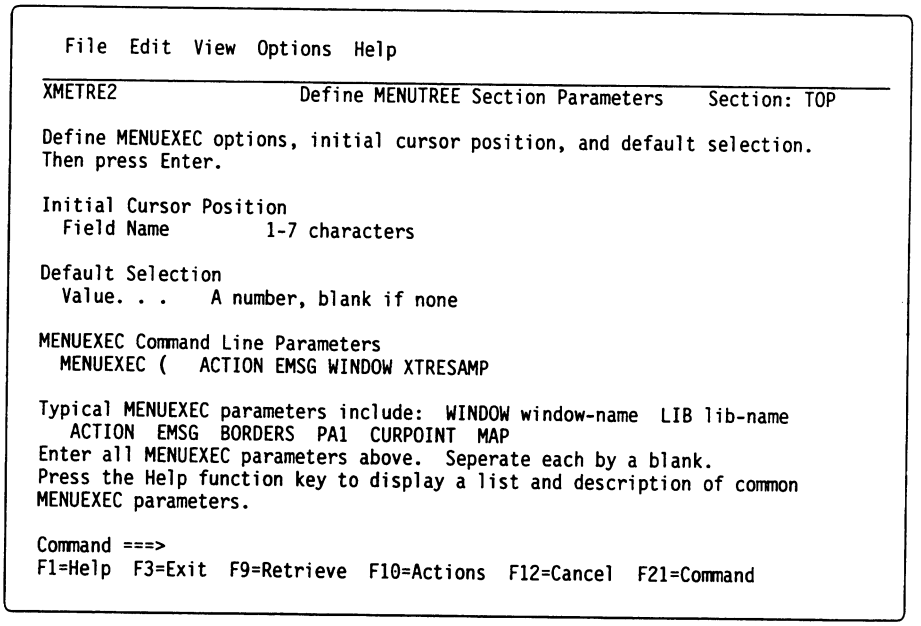
See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

Specifying the order of processing

XTREE allows you to alter the default order of user input processing. Normally, the menu hierarchy facility processes the action bar first, function keys second, and selection lists and user data third. By changing the values in the *Order of user input processing*, you can tailor the order to your application needs.

When you are finished, press PF12 or PF24 to return to the main menu.

Specifying Menu Parameters



You get to this menu when you type *2* in the main menu selection list.

The above menu allows you to tailor the display format of this hierarchy section.

Initial Cursor Position

The *Initial Cursor Position* field can be used to change the position of the cursor when the menu is displayed. You can type the XMENU field name of the field where you want the cursor to appear.

If you do not enter a cursor position, the cursor will be placed at the position defined when the XMENU menu was created. For XMENUCUA menus, this is typically the first input field.

If you want to position the cursor on a selection list input field, and are using XMENUCUA default names, type *Sx* where x is the selection field number. If you enter *SxCy* the cursor will be placed on choice y of selection list x (not in the input area).

Default Selection

If your menu has a selection list, you can specify the number of a default selection item, that is, the item that would be selected if the user simply presses enter. If this field is left blank, the user will have to make a specific selection. If this field is filled, the number specified will be displayed in the selection list input area.

If your menu has several selection lists, this value is used for the first list.

MENUEXEC command line parameters

This input area is used to enter MENUEXEC command line parameters. MENUEXEC is the XMENU utility used to display menus in the hierarchy. Information about MENUEXEC can be found in the *XMENU/REXX Interface User's Guide and Reference.*

Here are some typical MENUEXEC parameters you might want to specify:

WINDOW name names the window this menu will be displayed in. Any application using windows must give a window name for all displayed menus. Generally, you should use the same name for all full-size menus (those taking up the entire screen), and different names for each pull-down or pop-up window.

LIB libname used if your menu is in an XMENULIB.

ACTION needed if your menu has an action bar.

EMSG causes MENUEXEC to display informative messages in the case of errors.

BORDERS causes pull-down or pop-up windows to be displayed in borders. If using borders, also specify *TXTBORDE* in the XTREE MENUEXEC subcommands menu.

MAP returns PF13-24 and PF25-36 as PF01-12.

PA1 allows you to trap the pressing of the PA1 key.

CURPOINT allows you to use the cursor to point at character strings on the menu.

There are many more MENUEXEC subcommands—check to see if you can benefit from their use. XTREE doesn't check the correctness of these parameters; you should use the XTREE test facility to be sure that you have entered these correctly.

When you are finished, press PF12 or PF24 to return to the main menu.

Defining Action Bar Commands

This menu is used to define commands issued when action bar items are selected.

You get to this menu when you type *3* in the main menu selection list.

```
  File  Edit  View  Options  Help
XMETRE3                  MENUTREE Actions Commands        Section: TOP

Enter an action and a command for each Action.  Then press Enter.

Action               Command
File                 Call File
Edit                 Call Edit
View                 Call View
Options              Call Options
Help                 Call Help









Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

Type the action bar character string (as displayed in the action bar—case is not significant) in the column labelled *Action*, and the command to be invoked in the column marked *Command.*

To be SAA/CUA compatible, action bar elements should always display a pull-down menu directly below the action bar item. Therefore, if you want your application to be SAA compatible, you should only use *call* commands referencing pull-down menu sections.

This example shows only call commands.

See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

If an XMENUCUA menu was automatically loaded by XTREE, the action bar character strings are automatically filled into the action column.

When you are finished, press PF12 or PF24 to return to the main menu.

Defining Function Key Commands

This menu is used to define what commands are issued when program function keys are pressed:

```
  File  Edit  View  Options  Help
XMETRE4                MENUTREE Function Key Commands        Section: TOP

Enter a function key and command for each key.  Then press Enter.

Key             Command
PF01            Help
PF02
PF03
PF04
PF05
PF06
PF07
PF08
PF09            Retrieve
PF10            Actions
PF11
PF12            Call All_Done


Command ===>
F1=Help  F3=Exit  F7=Back  F8=Fwd  F9=Retrieve  F10=Actions  F12=Cancel
```

You get to this menu when you type *4* in the main menu selection list.

The valid function keys are displayed to the left. Type the command to be issued when this key is pressed in the column marked *Command.*

You can press PF08 or PF20 to move forward to another set of keys, or press PF07 or PF19 to move back to a previous set of keys. The first set of keys that are displayed are PF01 through PF12. The second set of keys that are displayed are PF13 through PF24. The last set of keys that are displayed are PA1 through PA3, and Enter.

See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

If an XMENUCUA menu was automatically loaded by XTREE, the function key character strings are automatically filled into the command area. If the displayed string doesn't match the command you want to issue, you may want to make changes on this menu.

If you want a function key to be accepted from all hierarchy sections, define it in the global function key menu instead of defining it in each section.

Any command defined for the enter key is only executed if the cursor is outside of the command line field. If the cursor is inside the command line field (CMDLINE) the enter key is taken as a request to execute the command line command.

When you are finished, press PF12 or PF24 to return to the main menu.

Specifying Selection List Commands

This menu is used to specify the commands that are issued when an item is chosen from a selection list.

You get to this menu when you type *5* in the main menu selection list.

```
  File  Edit  View  Options  Help
XMETRE5              MENUTREE Selection List Commands      Section: TOP

Enter a selection number, field name and command for each selection list
element.  Then press Enter.

Select  Field    Command                                        Value: 1
1       S1C1     call xtrsamp2
2       S1C2     call xtrsamp3
3       S1C3     call xtrsamp4
4       S1C4     call xtrsamp5
5       S1C5     call xtrsamp6
6       S1C6     call xtrsamp7
7       S1C7     call xtrsamp8
8       S1C8     call xtrsamp9
9       S1C9     call xtrsmp10



Command ===>
F1=Help  F3=Exit  F7=Back  F8=Fwd  F9=Retrieve  F10=Actions  F12=Cancel
```

You must specify three things for each selection list item:

1. The number value used to make a selection.
2. The field name of the field the cursor can be moved into to make a selection. By convention, if you use XMENUCUA, selection fields are named SxCy, where x is the selection list number, and y is the selection number.
3. The command to be issued if this selection is chosen.

You can press PF08 or PF20 to add additional selection commands, or press PF07 or PF19 to move back to a previous set of selection commands. If you have more than one selection list on your menu, XTREE determines the list this command is for by the field naming convention, that is, field names *Sx* refer to selection list *x*.

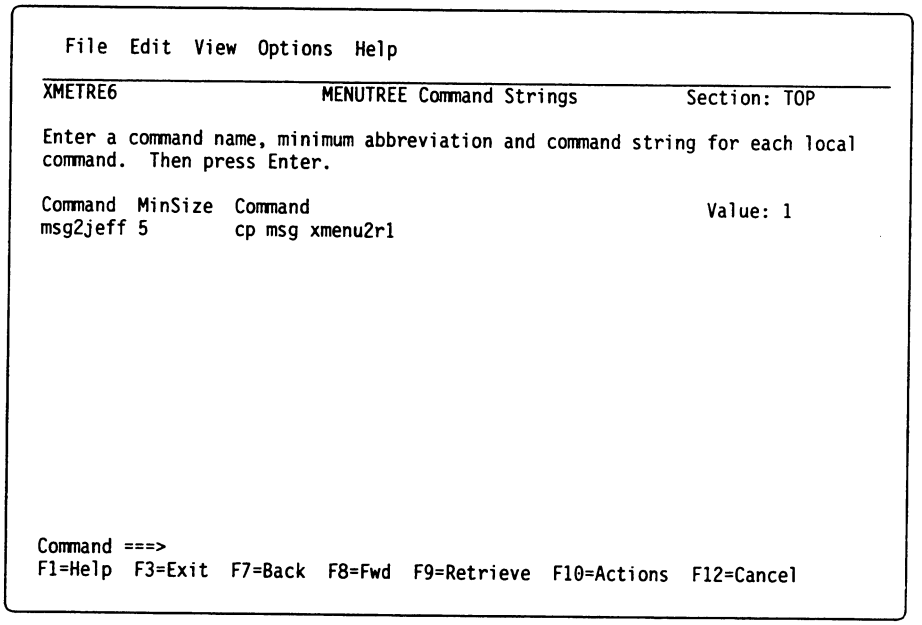
See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

If an XMENUCUA menu was automatically loaded by XTREE, the selection list numbers, field names, and character strings (prefixed by an asterisk to make them a comment) are automatically filled into the three selection list columns. You will most likely want to edit these comments into commands. XTREE simply starts you out with the selection list display data so that you can easily see what each selection is meant to do.

When you are finished, press PF12 or PF24 to return to the main menu.

Defining Local Commands

This menu is used to define command line commands:



You get to this menu when you type *6* in the main menu selection list.

You must specify three things for each command:

1. The name of the command—one to eight characters.
2. The minimum number of characters the user must type for the command to be recognized. If you don't enter this, XTREE sets it to the number of characters in the command name.
3. The command to be issued if this command name is typed on the command line.

You can press PF08 or PF20 to enter additional commands, or press PF07 or PF19 to move back to a previous set of commands.

When your application runs, if the user enters anything following the command name, it is appended to the command string defined in this menu. For example, using the sample data above, if the user enters

```
MSG2J How are you doing?
```

the command

```
cp msg xmenu2r1 How are you doing?
```

will be issued to CP (sending a message to userid xmenu2r1). Note that in this example the minimum abbreviation was typed (five characters).

If you want a command to be accepted from all hierarchy sections, define it in the global command menu instead of defining it in each section.

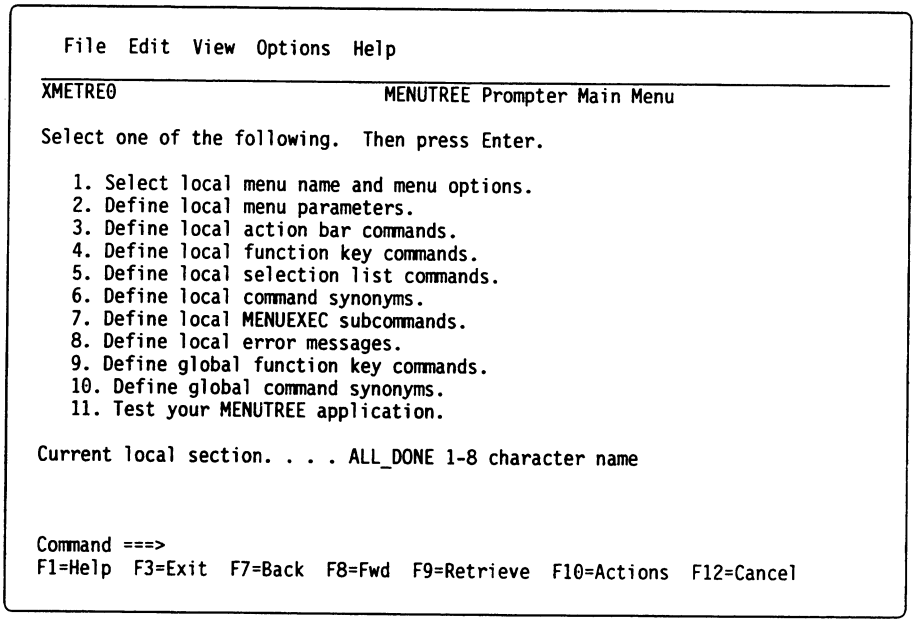
You only need to use this screen to enter command synonyms since XTREE automatically calls EXECs, CMS, and CP commands. You can also define command line commands to cause "fast path" calls or returns to specific menus in the hierarchy.

See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

Specifying MENUEXEC Subcommands

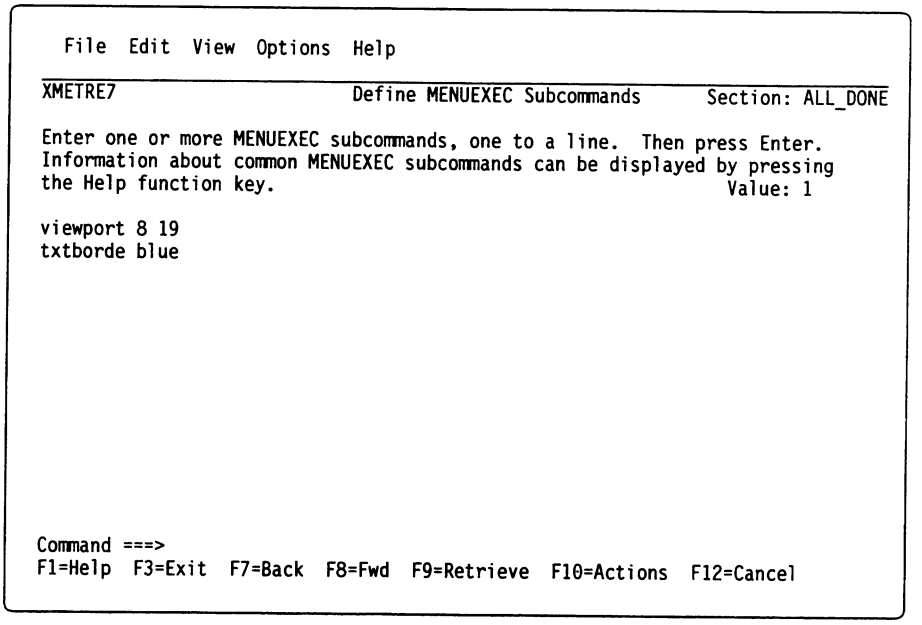
You can provide a list of MENUEXEC subcommands that will be processed whenever this section's menu is displayed. Commonly, you use these to position a pop-up or pull-down window, but you can use these for any purpose you wish.

In our example, the top section doesn't have any MENUEXEC subcommands. Therefore, we are going to display the subcommands from another section. First, we scroll to the section *ALL_DONE*. This is done from the main menu by pressing PF08 or PF20 until we reach this section (or by typing the section name and pressing enter):



Then, we enter *7* from the selection list.

The following menu is displayed:



Enter one subcommand per line. In this example, two are shown—one causes the pop-up window to be displayed starting in row 8, column 19. The second causes the borders of the window to be solid blue lines.

There are many MENUEXEC subcommands—normally you won't need more than a few at most.

You usually want the minimum two subcommands shown above if this section is going to display a pop-up or pull down window, so that it is positioned on the screen properly.

XTREE doesn't check the correctness of these subcommands; you should use the XTREE test facility to be sure that you have entered these correctly.

Information about MENUEXEC can be found in the *XMENU/REXX Interface User's Guide and Reference.*

Defining Error Messages

You can specify the custom messages displayed if a user enters an invalid selection, presses an undefined function key, or enters an unknown command.

The message text for each of these conditions is entered on the menu below:

```
  File  Edit  View  Options  Help
XMETRE8                    Define MENUTREE Error Messages        Section: TOP

Enter appropriate error message text.  Then press Enter.

Message for Invalid Selection
  Text. . . . You must make a selection.

Message for Invalid Function Key
  Text. . . . You have pressed an undefined function key.

Message for Invalid Command Entry
  Text. . . . You have entered an unknown command.




Command ===>
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel  F21=Command
```

You get to this menu when you type *8* in the main menu selection list.

If no messages are specified for this section, any error entry made when the hierarchy runs is simply ignored.

When you are finished, press PF12 or PF24 to return to the main menu.

Defining Global Function Key Commands

This menu is used to define commands issued when program function keys are pressed from any hierarchy section.

You get to this menu when you type *9* in the main menu selection list.

```
  File  Edit  View  Options  Help
-------------------------------------------------------------------------------
XMETRE9              MENUTREE Global Function Key Commands

Enter a function key and command for each global key.  Then press Enter.

Key             Command
PF01
PF02
PF03            Call All_Done
PF04
PF05
PF06
PF07
PF08
PF09
PF10
PF11
PF12



Command ===>
F1=Help  F3=Exit  F7=Back  F8=Fwd  F9=Retrieve  F10=Actions  F12=Cancel
```

The valid function keys are displayed to the left. Type the command to be issued when this key is pressed in the column marked *Command.*

Unless a key is specifically defined for a section, these commands take effect for all sections of the menu hierarchy.

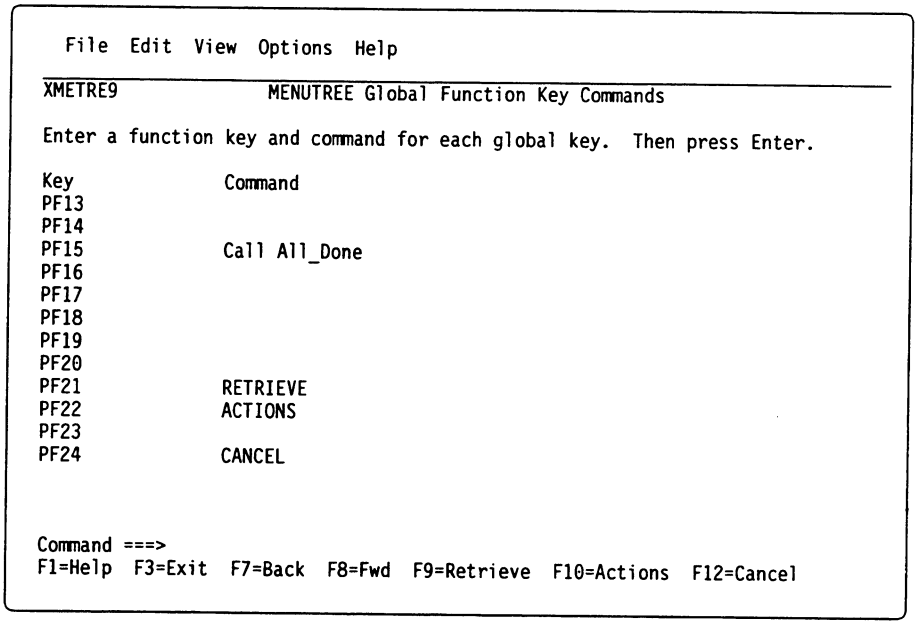
You can press PF08 or PF20 to move forward to another set of keys, or press PF07 or PF19 to move back to a previous set of keys. The first set of keys that are displayed are PF01 through PF12. The second set of keys that are displayed are PF13 through PF24. The last set of keys that are displayed are PA1 through PA3, and Enter.

See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

Any command defined for the enter key is only executed if the cursor is outside of the command line field. If the cursor is inside the command line field (CMDLINE) the enter key is taken as a request to execute the command line command.

When you are finished, press PF12 or PF24 to return to the main menu.

This menu shows an alternate set of function key definitions:



You get to this menu by pressing PF08 or PF20 from the first function key menu, or by pressing PF07 or PF19 from the third function key menu.

Defining Global Commands

This menu is used to define command line commands:

```
  File  Edit  View  Options  Help
XMETRE10              MENUTREE Global Command Strings

Enter a command name, minimum abbreviation and command string for each global
command.  Then press Enter.

Command  MinSize  Command                                          Value: 1
msg2jeff 5        cp msg xmenu2r1
lunch    5        cp msg xmenu2r1 ready for lunch?









Command ===>
F1=Help  F3=Exit  F7=Back  F8=Fwd  F9=Retrieve  F10=Actions  F12=Cancel
```

You get to this menu when you type *10* in the main menu selection list.

You must specify three things for each command:

1. The name of the command—one to eight characters.
2. The minimum number of characters the user must type for the command to be recognized. If you don't enter this, XTREE sets it to the number of characters in the command name.
3. The command to be issued if this command name is typed on the command line.

You can press PF08 or PF20 to enter additional commands, or press PF07 or PF19 to move back to a previous set of commands.

Unless a command is specifically defined for a section, these commands take effect for all sections of the menu hierarchy.

When your application runs, if the user enters anything following the command name, it is appended to the command string defined in this menu. For example, using the sample data above, if the user enters

```
MSG2J How are you doing?
```

the command

```
cp msg xmenu2r1 How are you doing?
```

will be issued to CP (sending a message to userid xmenu2r1). Note that the minimum abbreviation was typed in this example (five characters).

You only need to use this screen to enter command synonyms since XTREE automatically calls EXECs, CMS, and CP commands. You can also define command line commands to cause "fast path" calls or returns to specific menus in the hierarchy.

See "Menu Hierarchy File Format and Commands" on page 95 for a list of valid commands that can be specified in these fields. If a command isn't recognized, it is passed to CMS or CP when the application is run.

When you are finished, press PF12 or PF24 to return to the main menu.

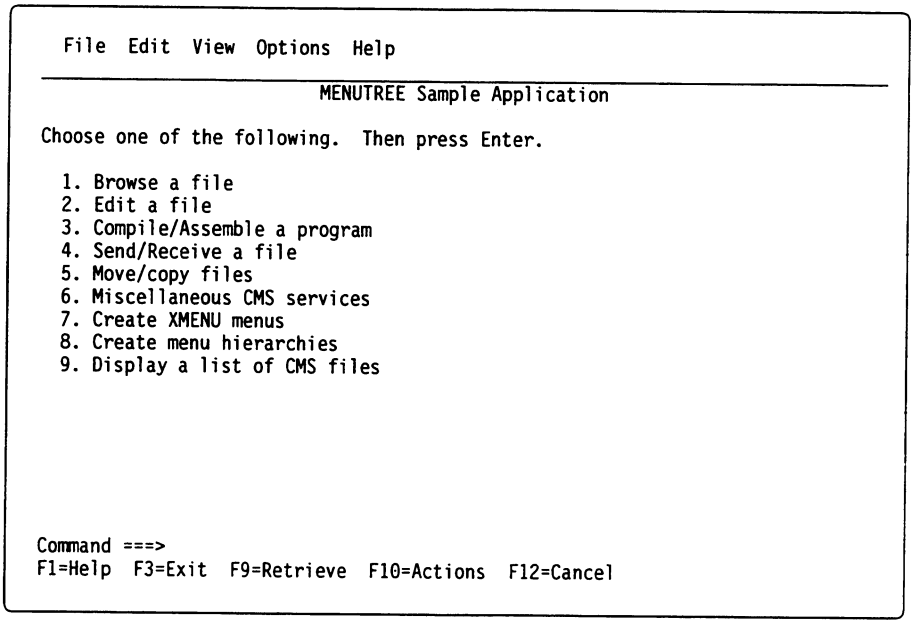
Testing your Menu Hierarchy

Enter *11* in the selection area of the main menu to test your menu hierarchy.

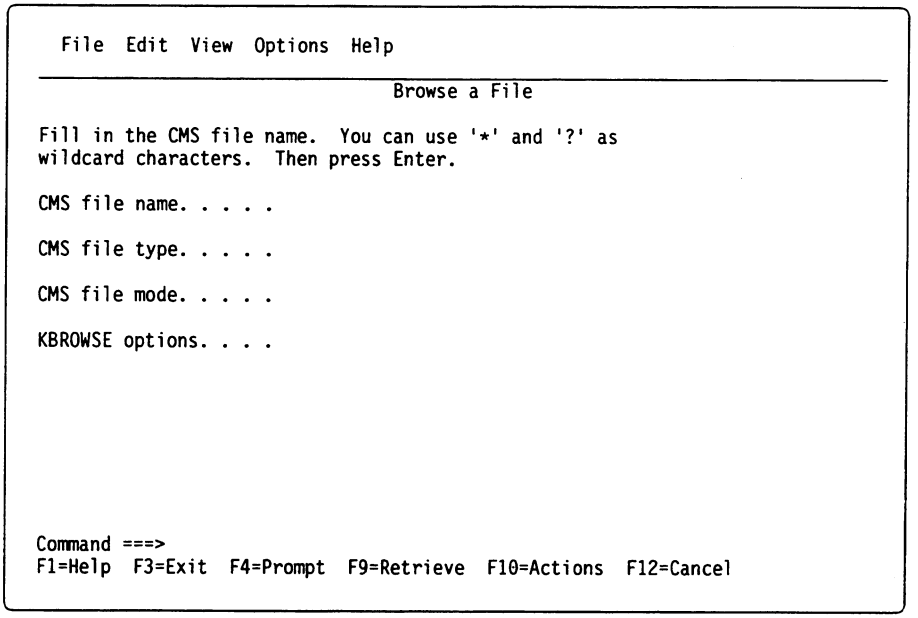
The menu hierarchy testing facility displays all menus used in your application, but does not allow the invocation of external commands. You use this to verify that your hierarchy is complete, and to test various input elements, such as the action bar and selection lists.

The XTREE test facility verifies that all commands are complete and that all sections and menus are defined properly. You should save your work before using this testing facility.

If you are testing with the supplied example XTRSAMPL, XTREE would display the XTRSAMPL application's initial menu:



If you select item *1* from the tested main menu, you should display this menu:



If you press PF12 or PF24, you will return to the tested main menu.

You should continue to test all action bar, selection list, function key, and command line commands you have defined in your menu hierarchy file.

Issuing external commands

If you attempt to run an external command from a tested display's command line as in this example:

```
File  Edit  View  Options  Help
──────────────────────────────────────────────────────────────────────
                        MENUTREE Sample Application

Choose one of the following.  Then press Enter.

  1. Browse a file
  2. Edit a file
  3. Compile/Assemble a program
  4. Send/Receive a file
  5. Move/copy files
  6. Miscellaneous CMS services
  7. Create XMENU menus
  8. Create menu hierarchies
  9. Display a list of CMS files




Command ===> msg2jeff hi did you get this?
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel
```

a pop-up will appear showing you the actual text of the external command that would have been invoked:

```
File  Edit  View  Options  Help
──────────────────────────────────────────────────────────────────────
                        MENUTREE Sample Application

Choose one of the following.  Then press Enter.
   ┌──────────────────────────────────────────────────────────┐
 1 │                  XTREE Menu Hierarchy Tester             │
 2 │                                                          │
 3 │  Since this is only a test of your menu hierarchy, XTREE will
 4 │  not actually execute any external commands.  The command
 5 │  that would have been executed at this point appears below.
 6 │  Press PF12/PF24 to continue your testing.
 7 │                                                          │
 8 │  Executed command: MSG2JEFF hi did you get this?
 9 │                                                          │
   │  F12=Cancel                                              │
   └──────────────────────────────────────────────────────────┘




Command ===> msg2jeff hi did you get this?
F1=Help  F3=Exit  F9=Retrieve  F10=Actions  F12=Cancel
```

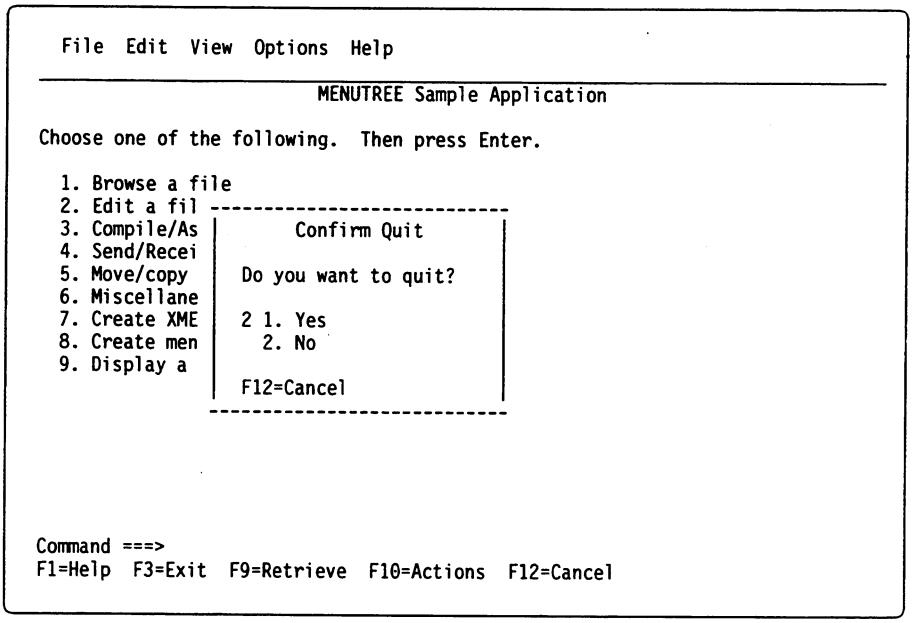
Leaving test mode

Use a function key or command that you've defined in your menu hierarchy to leave test mode and return to XTREE. Most applications use PF03 or PF15 as standard EXIT keys. Alternatively, you could enter *exit* on the command line.

Only in test mode, XTREE treats PA2 as a special *hot key* to exit the tested hierarchy and return to the XTREE main menu.

You should always save your hierarchy to disk before testing, in case your application goes into a loop or has other fatal problems. For example, while XTREE attempts to protect you from calling outside applications, it cannot prevent these calls from any MENUEXEC REQUIRE EXECs that you have asked the hierarchy to call.

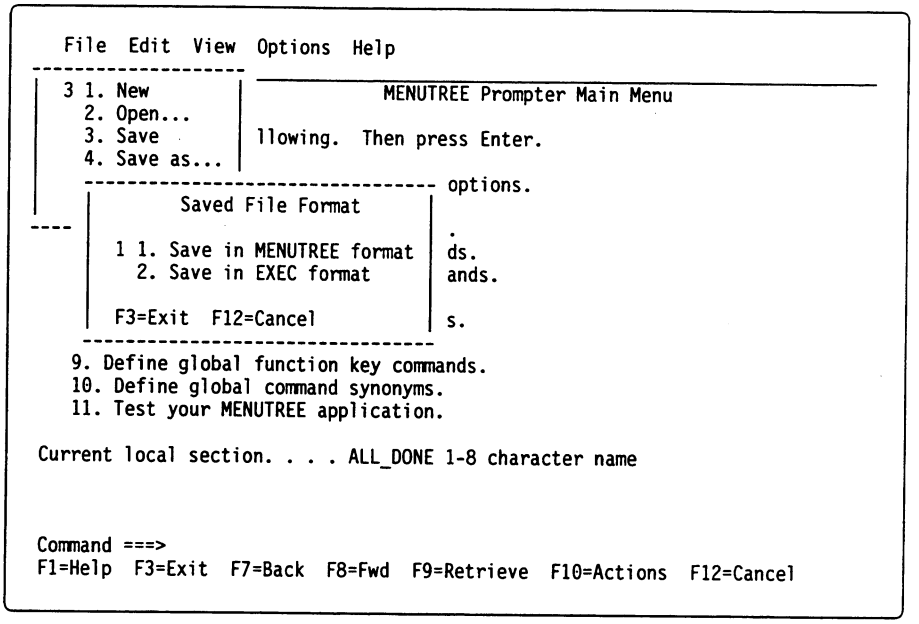
Our sample application displays a verification pop-up before exiting:



Saving the Menu Hierarchy

We haven't saved our work yet.

Select *File* from the action bar, and select *Save as*. You should see the following pop-up windows:



This menu allows you to choose the format of the data to be saved.

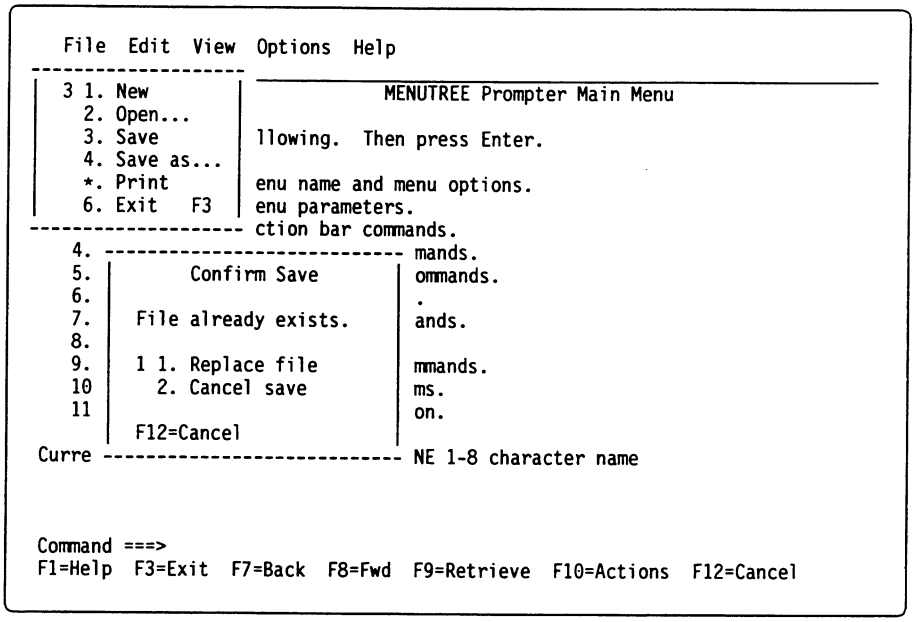
If you select *1* (the default), the file is saved in MENUTREE format. This format is discussed in the next chapter.

If you select **2**, a REXX EXEC is built that will run the menu hierarchy in a more efficient manner. You probably don't want to do this until your application is complete. Also, this function does not let you append custom code to your REXX EXEC—use XTRECOMP to do this. XTRECOMP is described in "XTRECOMP—The MENUTREE Compiler" on page 105.

You should always save your work in MENUTREE format, as this is the only way that you can make subsequent modifications to your work.

Press *Enter* to continue the saving process.

If the menu hierarchy file already exists, you will see this:



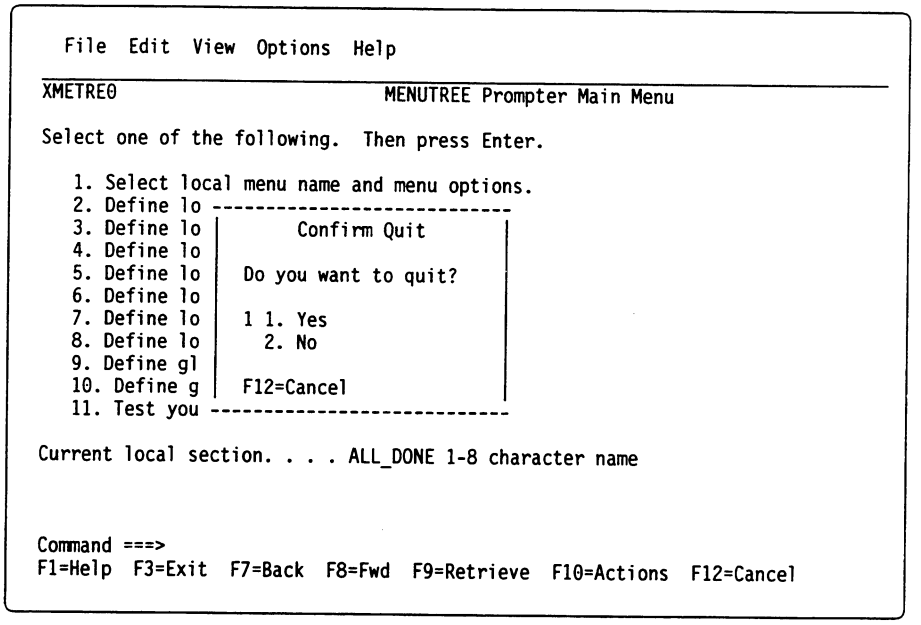
Press *Enter* to save the MENUTREE file. If you don't want to overwrite this file, type *2* to cancel this save, then run another save, and select a different CMS filename.

You can also use the Prompt function key to select a saved filename.

Leaving XTREE

If you want to leave XTREE now, press the *Exit* key.

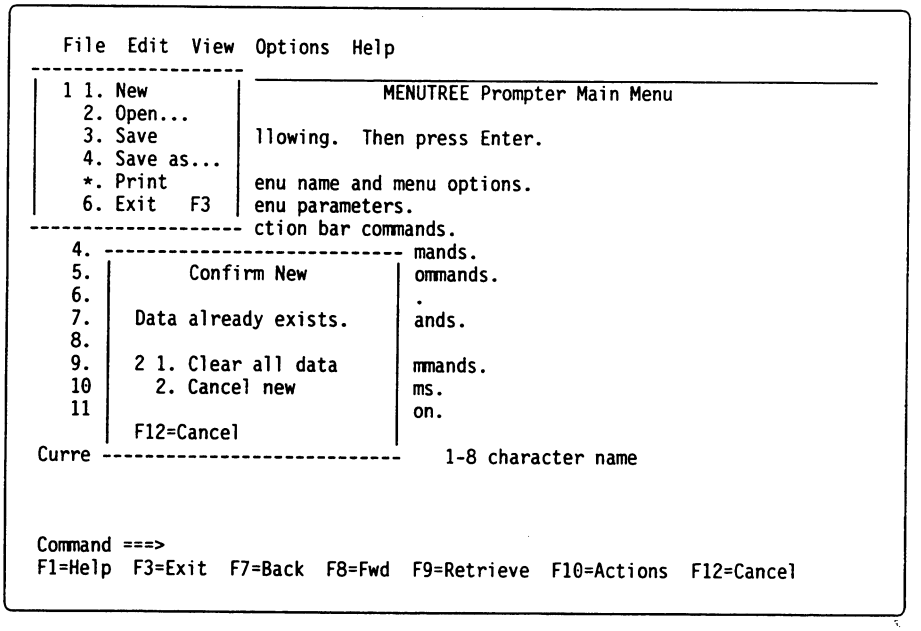
You should get the following:



Select *1* to quit, then press *Enter*. Otherwise, just press *Enter* to continue using XTREE.

If you want to start a new hierarchy, select *File* from the action bar and select *New* from the pull-down window.

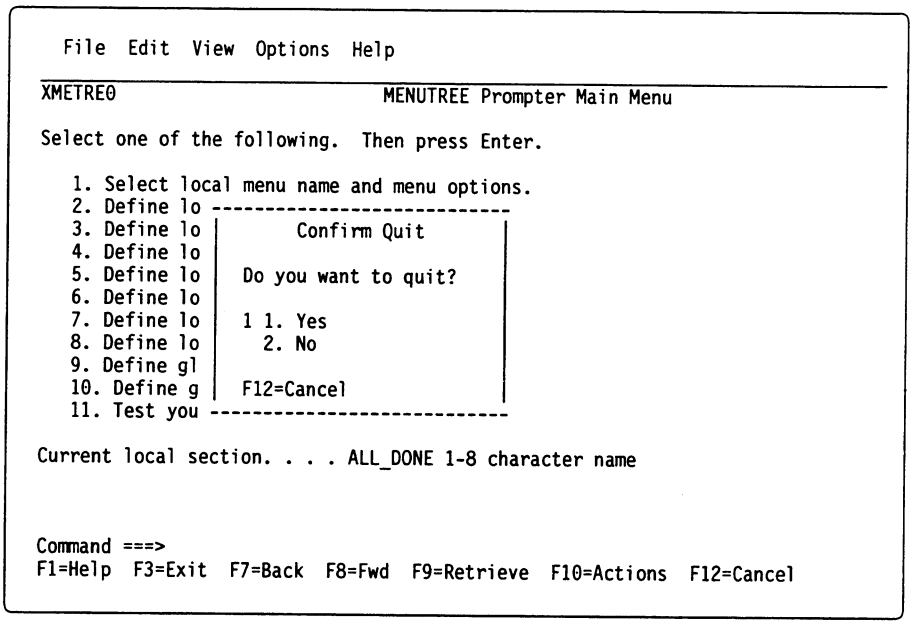
The following should appear:



This allows you to confirm that you want to start a fresh new editing session.

Select *1* to start a new panel, then press *Enter*. Otherwise, just press *Enter* to continue using the existing data.

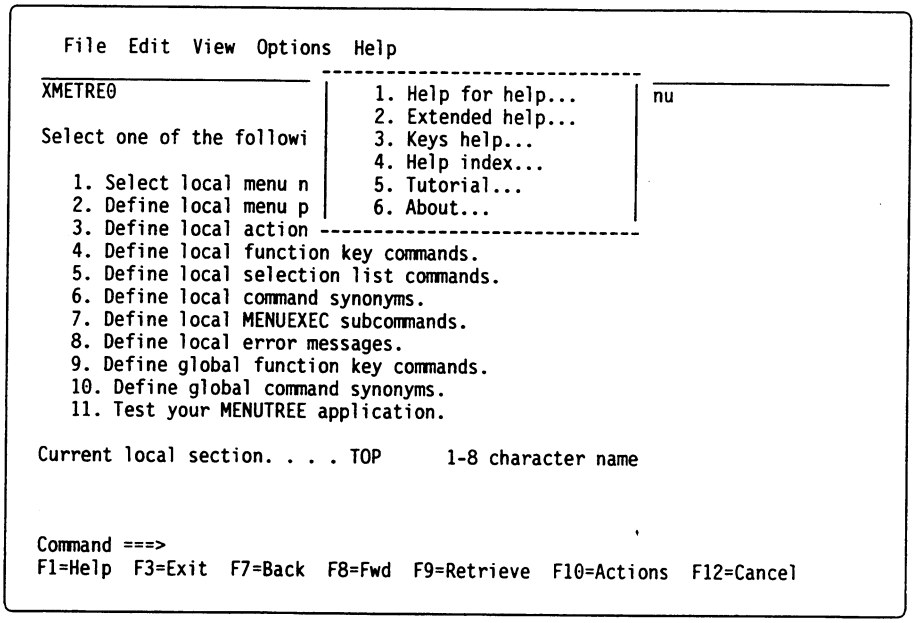
If you attempt to leave XTREE before you have edited anything, you will see this display:



Just press *Enter* to exit XTREE. Otherwise, select *2* to continue using XTREE.

Displaying XTREE Help

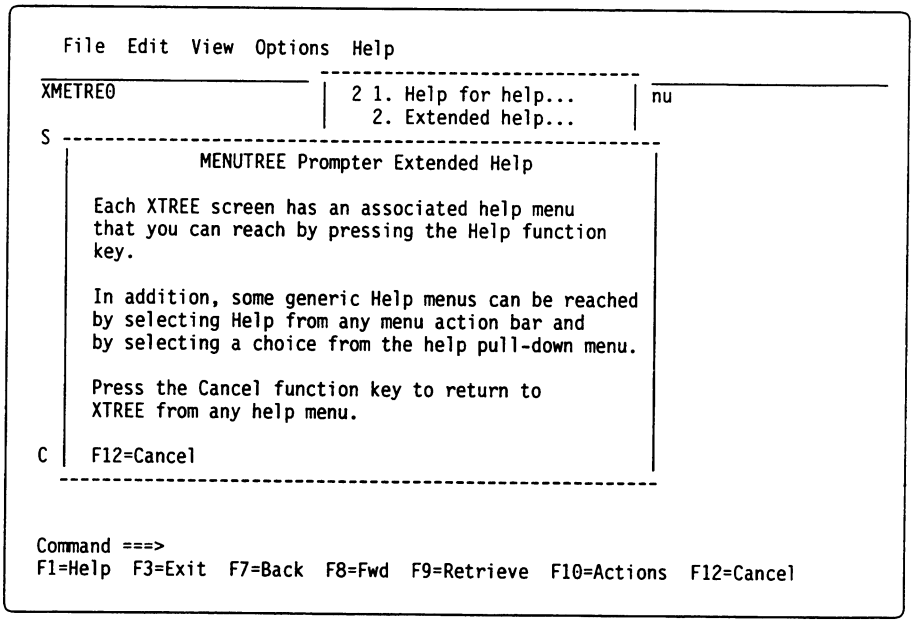
A certain amount of online help data is provided with XTREE. If you select *Help* from the action bar you will get this pull-down menu:



You can choose the kind of help you want by selecting an item from this pop-up window.

You can use the *Cancel* key to return to XTREE.

If you selected *6* from the Help pull-down window you will get this display:



The MENUTREE Program

This chapter describes how you run your menu hierarchy using the MENUTREE program.

To run MENUTREE, type:

```
MENUTREE [filename] [parameters [(options]]
```

Filename refers to the name of the MENUTREE file we wish to process. Any parameters and options entered on the command line are saved so that your application can process them.

A certain amount of overhead is incurred while reading and processing the MENUTREE control file. In addition, since MENUTREE only processes the control file, internal calls (ICALL statements) are not executed. You can convert your MENUTREE control file into a standalone REXX EXEC by using XTRECOMP (described in "XTRECOMP—The MENUTREE Compiler" on page 105).

MENUTREE can be called from the CMS command line, other EXECs or application programs capable of calling CMS commands.

Menu Hierarchy File Format and Commands

This chapter describes the format of the statements in the MENUTREE hierarchy file. If you don't wish to use the XTREE prompter, you may create these statements directly with XEDIT, then test them with the XTREE test facility.

Overview

A MENUTREE file contains statements that direct the flow of a menu hierarchy.

The file consists of a group of *sections*, each of which describe a single display—a full screen menu, a pull-down menu, a help panel, or a pop-up window—and its relationships with other displays.

MENUTREE File Format

A MENUTREE file can be variable or fixed format, with a logical record length of up to 255 characters.

Each command must be completely contained on a single file line. Operands must be separated by one or more blanks.

Usage Notes

Upper case letters show the minimum abbreviation for keywords.

Optional statements are surrounded in brackets ([ ]).

A choice of one or another operand is specified by the use of the vertical bar (|) between the operands.

Global statements must precede all section definitions.

Section statements can be in any order but must be kept together within a section definition.

While it isn't a requirement, we suggest that you indent section statements to make your file easier to read.

Since this facility is implemented in REXX, it is advisable to avoid using double quotes in character or command strings.

All definitions are given in alphabetical order.

Summary of MENUTREE File Format

The following pages give a BNF-like summary of a MENUTREE file's format.

```
File contents: [Global definitions]
Section definitions
Global definitions: [KEYS key-name command-string]
[GLOBAL_Command command-name minimum-size command-string]
Section definitions: section-name: menu-name [( menu-parameters]
[section-commands]
key-name = PF1-PF24|PA1-PA3|Enter
menu-name = 1-8 valid CMS filename characters
section-name = 1-8 valid REXX symbol name characters
section-commands = ACTion act-name command-string
                   AFTer restricted-command-string
                   BEFore restricted-command-string
                   COMMand command-name minimum-size command-string
                   CURsor field-name
                   DEFSelection sel-number
                   INITially restricted-command-string
                   key-name command-string
                   MESSAGE|MSG msg-type message-string
                   ORDER A|F|D A|F|D A|F|D
                   SELection sel-number sel-field command-string
                   SUBCmd menuexec-subcommand
                   TRACE
A = process action bar
act-name = 1-20 character action bar name
command-name = 1-8 characters
command-string = command-verb command-parameters [( command-options]
D = process selection list and user data
F = process function key
field-name = 1-7 valid XMENU field name characters
key-name = PF1-PF24 PA1-PA3 Enter
minimum-size = decimal value <= command-name-length
msg-type = COMMAND|SELECTION|PFKEY
restricted-command-string = restricted-command-verb parameters [( options]
sel-field = 1-7 valid XMENU field name characters
sel-number = decimal digit

command-verb = ACTIONS
               * comments
               CALL section-name
               CANCEL|RETURN
               COMMAND
               EXIT
               GOTO section-name
               ICALL exec-subroutine
               NOOP
               =|REPEAT
               ?|RETRIEVE
               Valid CMS|CP command
               Valid EXEC name
restricted-command-verb = ICALL exec-subroutine
                          Valid CMS|CP command
                          Valid EXEC name
exec-subroutine = Valid, existing internal REXX subroutine name
section-name = 1-8 valid REXX symbol name characters
```

MENUTREE File Commands

These commands appear in a MENUTREE file.

Global_commands

Defines a global command name, that is, one valid in all sections.

The format of this command is:

```
GLOBAL_command command-name minimum-size command-string
```

Command-name must be 1-8 characters. Minimum-size is the minimum allowed abbreviation for command-name.

Command-string can be any valid run-time MENUTREE command.

If this command is specified more than once, the last copy found is used.

For a MENUTREE command line to function properly, it must have a field name of CMDLINE, and be defined with its Modified Data Tag (MDT) set.

As MENUTREE automatically passes commands to CMS and CP, this facility is mostly intended to define "fast path" command line commands to immediately go to a specific section of a menu hierarchy, or to allow experienced users to quickly enter sophisticated commands.

KEYS—Global Key Definitions

Specifies a command to issue if a program function key is pressed. This takes effect for all sections.

The format of this command is:

```
KEYS key-name command-string
```

Key-name can be PF1 (or PF01) through PF24, PA1 through PA3 or Enter. If key-name is pressed, command-string is executed.

Any command defined for the enter key is only executed if the cursor is outside of the command line field. If the cursor is inside the command line field (CMDLINE) the enter key is taken as a request to execute the command line command.

This command can be specified more than once.

Section Heading

Defines the start of a section.

The format of this command is:

```
Section-name: menu-name [( menu-parameters]
```

Section-name must be 1-8 valid REXX symbol name characters. Menu-name must be an existing XMENU menu in MENU or XMENULIB format. Menu--parameters are passed to MENUEXEC as MENUEXEC command line parameters.

See the *XMENU/REXX Interface User's Guide and Reference.* for information on MENUEXEC command line parameters.

ACTION

Specifies a command to issue if an action bar item is selected.

The format of this command is:

```
ACTION act-name command-string
```

The case of act-name is not significant. If act-name is returned as the action bar choice, command-string is executed.

This command can be specified more than once, and can appear anywhere within a section.

AFTER

Specifies a command to issue after a menu is displayed.

The format of this command is:

```
AFTer restricted-command-string
```

Only ICALL, CMS, and CP commands can be passed. This command is normally used to retrieve and process user input data.

This command is processed before selection data. If the user pressed Enter, it is also processed before any Enter key command is processed.

If this command is specified more than once in a section, the last copy found is used.

BEFORE

Specifies a command to issue before a menu is displayed.

The format of this command is:

```
BEFore restricted-command-string
```

Only ICALL, CMS, and CP commands can be passed. This command is normally used to move output data to a menu.

If this command is specified more than once in a section, the last copy found is used.

COMMAND

Defines a command name.

The format of this command is:

```
COMMand command-name minimum-size command-string
```

Command-name must be 1-8 characters. Minimum-size is the minimum allowed abbreviation for command-name.

Command-string can be any valid run-time MENUTREE command.

If this command is specified more than once in a section, the last copy found is used.

This command overrides any identically named global commands.

For a MENUTREE command line to function properly, it must have a field name of CMDLINE, and be defined with its Modified Data Tag (MDT) set.

As MENUTREE automatically passes commands to CMS and CP, this facility is mostly intended to define "fast path" command line commands to immediately go to a specific section of a menu hierarchy, or to allow experienced users to quickly enter sophisticated commands.

CURSOR

Defines the initial cursor position.

The format of this command is:

```
CURsor field-name
```

Field-name must be a valid, existing XMENU field on the menu.

If this command isn't specified, the cursor is displayed where it was defined when the menu was created.

If this command is specified more than once in a section, the last copy found is used.

DEFSELECTION

Defines the default selection list selection.

The format of this command is:

```
DEFSelection sel-number
```

Sel-number must be a valid decimal number.

If this command isn't specified, the user must choose a selection. If it is specified, sel-number appears in the selection list input area, and is selected if the user doesn't modify it.

If this command is specified more than once in a section, the last copy found is used.

INITIALLY

Specifies a command to issue the first time a menu is displayed.

The format of this command is:

```
INITially restricted-command-string
```

Only ICALL, CMS, and CP commands can be passed. This command is normally used to initialize a section or the entire application.

If this command is specified more than once in a section, the last copy found is used.

Key Definitions

Specifies a command to issue if a program function key is pressed.

The format of this command is:

```
key-name command-string
```

Key-name can be PF1 (or PF01) through PF24, PA1 through PA3 or Enter. If key-name is pressed, command-string is executed.

Any command defined for the enter key is only executed if the cursor is outside of the command line field. If the cursor is inside the command line field (CMDLINE) the enter key is taken as a request to execute the command line command.

This command can be specified more than once, and can appear anywhere within a section.

This command overrides any global function keys defined.

MESSAGE

Specifies a message that is displayed if the user selects an invalid choice, selects an invalid action bar item, or enters an unknown command.

The format of this command is:

```
MESSAGE|MSG msg-type message-string
```

Msg-type can be:

COMMAND message for unknown commands.

PFKEY message for undefined function keys.

SELECTION message for invalid selections.

The message is displayed in the XMENU menu field named *ERRMSG*.

To avoid problems with XTRECOMP converted control files, do not use double quotes in your messages (or place two wherever you want one to appear).

If this command is specified more than once in a section, the last copy found is used.

ORDER

Specifies the order of user input processing.

The format of this command is:

```
ORDER A|F|D A|F|D A|F|D
```

The order type can be:

A process action bar.

F process function keys.

D process selection list.

The default sequence is action bar, function keys, user-input and selection lists.

This command isn't checked at run-time—if you specify the same letter three times, that function will be run three times.

If this command is specified more than once in a section, the last copy found is used.

SELECTION

Specifies a command to issue if a selection list choice is chosen.

The format of this command is:

```
SELection sel-number sel-field command-string
```

Sel-number must be a valid decimal number. Sel-field must be a valid XMENU field name. If the cursor is placed within this field, and the selection input area is blank, this command will be executed.

This command can be specified more than once, and can appear anywhere within a section.

SUBCMD

Specifies a MENUEXEC subcommand used to display this menu.

The format of this command is:

```
SUBCmd menuexec-subcommand
```

Menuexec-subcommand must be a valid menuexec-subcommand. It is not syntax checked—if invalid, the MENUTREE EXEC will fail at run-time.

Subcommands are invoked in the order that they are found in the MENUTREE control file.

This command can be specified more than once, and can appear anywhere within a section.

See the *XMENU/REXX Interface User's Guide and Reference.* for information on MENUEXEC subcommands.

TRACE

Causes the REXX code invoking this section to be traced.

The format of this command is:

```
TRACE
```

This command is for debugging purposes only.

This command can be specified more than once, and can appear anywhere within a section.

MENUTREE Run-Time Commands

These are the commands that are executed by the menu hierarchy run-time processor as a result of user input.

ACTIONS

Moves the cursor to the action bar. A second use restores the original cursor position.

The format of this command is:

```
ACTIONS
```

* Command

Has no effect. This can be used to validate a command action without doing anything (such as silently accepting undefined selection choices).

The format of this command is:

```
* comments
```

CALL

Stacks the current menu and displays a new menu. The calling menu is returned to if a subsequent RETURN command is invoked.

The format of this command is:

```
CALL section-name
      or
CALL ( variable-name )
```

If the second form is used, the contents of the variable specified in variable-name is used as the section-name. This form can be used to modify the order of your hierarchy at run-time by setting variable-name based on current conditions. If you use this form, there must be at least one space between the parenthesis and the variable name.

CANCEL|RETURN

Returns to the previously displayed menu, that is, to the section where the most recent CALL was invoked. If this is called from the top section, the application exits.

The format of this command is:

```
CANCEL   or
RETURN
```

COMMAND

Moves the cursor to the command line. A second use restores the original cursor position.

The format of this command is:

```
COMMAND
```

EXIT

Causes the application to exit immediately.

The format of this command is:

```
EXIT
```

GOTO

Displays a new menu. The calling menu is not stacked.

The format of this command is:

```
GOTO section-name
      or
GOTO ( variable-name )
```

If the second form is used, the contents of the variable specified in variable-name is used as the section-name. This form can be used to modify the order of your hierarchy at run-time by setting variable-name based on current conditions. If you use this form, there must be at least one space between the parenthesis and the variable name.

ICALL

Calls a REXX subroutine within the run-time EXEC.

The format of this command is:

```
ICALL routine-name
```

This command is used to call internal REXX subroutines. You must add this routine name to the REXX subroutine called *local_subroutine*. See "XTRECOMP—The MENUTREE Compiler" on page 105 for more information about ICALL.

If the routine-name does not exist, an external REXX call is made instead.

NOOP

Performs no function.

The format of this command is:

```
NOOP
```

REPEAT

Repeats the last entered command line command.

The format of this command is:

```
REPEAT or
=
```

RETRIEVE

Redisplays the last entered command line command. Subsequent calls display up to ten previously entered commands.

The format of this command is:

```
RETRIEVE  or
?
```

XTRECOMP—The MENUTREE Compiler

XTRECOMP converts a MENUTREE file into a runnable REXX EXEC. This can result in much greater performance than simply running MENUTREE and you will be able to execute internal calls (ICALL statements) once you have inserted them into the resultant EXEC.

The XTREE utility can also save hierarchy files in REXX format, but does not have the NOCOPY and APPEND options. Use XTRECOMP when you need to use these options.

To run XTRECOMP, type:

```
XTRECOMP [filename] [( NOCOPY|APPEND aname]
```

If *NOCOPY* is specified, the run-time code is not appended to the generated REXX assignment statements. This option is only used for special purposes.

If *APPEND aname* is specified, the file *aname EXEC* is appended to the generated EXEC. This is how you add your own specialized code to a XTRECOMP-converted EXEC.

Note that we use the term *compiled* to refer to the MENUTREE data. The resulting EXEC is not compiled REXX. You could compile this REXX code to further improve performance.

ICALL—Adding your own run-time code

The REXX EXEC generated by XTREE and XTRECOMP contains the logic necessary to 'walk' through specific menus. For this DIALOG to be fully functional, you may need to add your own code to handle your application needs, such as retrieving and saving data. Processes invoked with the ICALL statement can be either external EXECs or internal subroutines. To incorporate an EXEC into the resultant XTRECOMP-generated code:

1. Create an exec—SAVEIT EXEC in this example— starting with the following code:

```
/*  Call locally defined subroutines... */
local_subroutine:
rc = 0
parse var temp_parms temp_cmd temp_parms
parse upper var temp_cmd temp_cmd
/*  Place internal subroutine calls here   */
select
   when 0 then nop
   otherwise 'EXEC' temp_cmd temp_parms
end
return rc
/*  Place locally defined subroutines here... */
```

2. Create a MENUTREE file (hierarchy) using XTREE.

   When building the hierarchy using XTREE, you will need to specify when the subroutine (or exec) should be called. Let's assume that one of your menus has PF05 denoted as:

   ```
   PF05=SAVEIT
   ```

When a user presses PF05, you want the SAVEIT EXEC to be executed. To do this, define an ICALL statement for PF05 using selection number 4 from XTREE's main menu. The following ICALL statement should be defined for PF05:

```
Icall SAVEIT any_parms (any_options
```

(***any_parms*** define parameters you want passed to SAVEIT EXEC, and ***any_options*** define options you want passed to SAVEIT EXEC).

3. Convert the MENUTREE file to an EXEC by entering:

   ```
   XTRECOMP <filename> (APPEND SAVEIT
   ```

   where filename is the name of your MENUTREE control file.

4. XEDIT <filename> EXEC which was created in the previous step
5. Locate the ***local_subroutine:*** label
6. Insert:

   ```
   "when temp_cmd = 'SAVEIT' then call SAVEIT temp_parms"
   ```

7. Pull the SAVEIT EXEC into this exec:

   Locate the statement: ***Place locally defined subroutines here...***, and issue the XEDIT subcommand

   ```
   GETFILE SAVEIT EXEC
   ```

   and give it a label of SAVEIT:. (Remember to change any 'EXIT' statements to 'RETURN' as needed.

8. File and run the exec, verifying that PF05 executes SAVEIT.

External EXECs need no special preparation since the ICALL logic will attempt to execute an EXEC if no other action is defined. Simply create your dialog and reference the EXEC name as part of the ICALL statement, eg. ***ICALL MYEXEC***, and issue the appropriate XTRECOMP statement.

Changing Sections

Within your ICALL routine, you can call several REXX subroutines to force a section change.

ALL_DONE

This subroutine lets you return from a section.

Its syntax is:

```
call all_done(return-code)
```

CANCEL_LEVEL

This subroutine lets you return from a section. It does not cause an immeditate termination of the section, and is generally issued after all processing in a section.

Its syntax is:

```
call cancel_level
```

SET_LEVEL_RET

This subroutine lets you set the next level of the hierarchy to be executed upon exiting the current one, and cause the current level to be re-entered after its (the new level) completion.

Its syntax is:

```
call set_level_ret(level-name)
```

SET_LEVEL_NORET

This subroutine lets you set the next level of the hierarchy to be executed without returning to the current one. The level prior to the current one is entered after completion of the new level. This technique is used when a secondary window completes processing for a primary window.

Its syntax is:

```
call set_level_noret(level-name)
```

Other Techniques

By convention, you can cause the alarm to sound and/or the cursor to be moved by setting the *alarm* variable, for example:

```
alarm = 'ALARM'
```

would sound the alarm,

```
alarm = 'ALARM CURSOR X'
```

would sound the alarm and move the cursor to menu field x.

You can display error messages by placing your error message string into the variable called *errmsg*. for example:

```
errmsg = 'You must enter a valid social security number'
```

Menu Hierarchy Special Field Names

The following variables are maintained by XTREE and any REXX EXEC created with XTRECOMP. Certain other field names may be relevant to your application. These are described in "XMENU Field Names" on page 113.

If these field names are used in your menu(s), these fields will automatically be set to the values described below:

CMDLINE if this field exists, MENUTREE commands, CMS, and/or CP commands can be entered when the enter key is pressed.

For a MENUTREE command line to function properly, it must have a field name of CMDLINE, and be defined with its Modified Data Tag (MDT) set.

CMDPREF if this field exists, the resulting return code of the previous command entered on the CMDLINE is displayed. Otherwise, this field contains the string "Command ===>".

XAPPLN contains the name of this MENUTREE application (the MENUTREE CMS file name).

XBNAME contains the name of the hierarchy section being displayed.

XCALLER contains the name of the section calling this section if any.

XDAY contains the day of the month in DD format.

XDATE contains the date in MM/DD/YY format.

XDATEE contains the date in DD/MM/YY format.

XDATEF contains the date in dd mon year format.

XDATEO contains the date in YY/MM/DD format.

XDOW contains the day of the week in character format.

XJDATE contains the date in YY.DDD format.

XLEVEL contains the current MENUTREE recursion level (how many menus deep you are within MENUTREE).

XMNAME contains the name of the menu being displayed.

XMONTH contains the day of the month in character format.

XTIME contains the time of day in HH:MM:SS format.

XUSER contains your USERID.

XYEAR contains the year in YYYY format.

Menu Hierarchy Files

XTREE can create and use a number of CMS files. Here's a list of them and what they are used for:

Filetype	Function
MENUTREE	Contains a complete menu hierarchy definition. This file is used by XTREE to make subsequent alterations, by MENUTREE to display a hierarchy, and by XTRECOMP to create a converted EXEC.
XMENUSAA	Contains an XMENU SAA/CUA compatible definition file. This is the file generated by XMENUCUA. If this file exists, XTREE reads this file to fill in information about action bars, selection lists, and function keys.
MENU	Contains an XMENU menu. This is the file used by an XMENU application to present and retrieve terminal data from your end users.

XTREE comes with a few example hierarchy files.

XMENU Field Names

When XMENUCUA or XMEDIT creates an XMENU menu, it automatically names important fields so that they can be accessed by your application program.

While these names are not essential to run XTREE, the use of certain naming conventions makes programming with XTREE easier. If you create your own menus directly, we recommend using these names.

Also see "Menu Hierarchy Special Field Names" on page 109 for a list of special variables displayed by XTREE.

XMENUCUA names the title, parts of selection lists and entry areas, the command line, the error message line, and the function key area. All other menu fields are unnamed.

The following names are used by XMENUCUA:

Field Name	Function
PANELID	The Panel I.D. if any.
TITLE	The Panel title if any.
CMDPREF	The prefix to the command line if any.
ERRMSG	The error message line if any.
CMDLINE	The command line(s) if any. For a MENUTREE command line to function properly, it must have a field name of CMDLINE, and be defined with its Modified Data Tag (MDT) set.
PFKLINE	The function key line(s) if any.
En	The input field for entry area *n*. The number *n* is based on the order in which areas are created, not on their relative position in the menu.
Sn	The input area for single selection list *n*. The number *n* is based on the order in which areas are created, not on their relative position in the menu.
SnSx	Input area for choice *x* in multiple selection list *n*. The number *n* is based on the order in which areas are created, not on their relative position in the menu. *X* is line *x* in the list—this number does not necessarily correspond to numbers you've given selection list choices.
SnCx	Choice *x* in selection list *n*. The number *n* is based on the order in which areas are created, not on their relative position in the menu. *X* is line *x* in the list—this number does not necessarily correspond to numbers you've given selection list choices.

XMENU Hierarchy Utilities

KVARXFER

KVARXFER transfers REXX symbols from one invocation of REXX to another.

The syntax for this command is:

```
KVARXFER [FWD|BACK [ALL]]
```

If FWD or nothing is specified, then KVARXFER transfers the contents of all REXX variables from the most recently invoked REXX EXEC to the running REXX EXEC.

If BACK is specified, KVARXFER transfers the contents of all REXX variables from the currently running REXX EXEC to the REXX EXEC that called this one (that is, the previous REXX EXEC).

All existing REXX variables are copied, except for variables starting with the string *SYS$_*. If *ALL* is specified, these are copied also.

This command gives you a fast way to transfer variables between two running REXX EXECs. However, it can take a large amount of time if you have many EXEC variables.

Return codes

```
1 - Not running under REXX.
2 - No previous invocation of REXX found.
8 - Insufficient storage to run this command.
```

KREXLINE

KREXLINE converts REXX symbols found in character strings.

The format of this command is:

```
KREXLINE var-name1 var-name2
```

KREXLINE is passed the name of two REXX variables. As the name, rather than the contents is passed, be sure to enclose these names in quotes.

KREXLINE reads the contents of the first variable, does a single pass substitution of any REXX variables found in this data, then sets the second REXX variable to the converted string.

For example:

```
a = 'String 1'
b = 'String 2'
c = 'Please convert a and b'
'KREXLINE c d'
say d
```

would display

```
Please convert String 1 and String 2
```

You can use this program instead of the REXX function INTERPRET, allowing you to use the REXX compiler to compile otherwise uncompilable EXECs.

Return codes

```
1 - Not running under REXX.
2 - No/insufficient variables passed.
3 - The variable name(s) passed are invalid.
4 - A dropped or empty REXX input variable was passed.
8 - Insufficient storage to run this command.
```