

46

MENU CUSTOMIZATION

Chapter Objectives

After completing this chapter you should:

1. understand the menu structure;
2. be able to use name_tags, item labels, and menu macros;
3. be able to use special characters in menu macros;
4. be able to use line terminators;
5. understand the menu sections;
6. understand menu swapping;
7. understand the base menu versus partial menus.

CONCEPTS

One of the special features of AutoCAD is the ability to customize the menu system. AutoCAD enables you to change the menu system by providing the primary menu, ACAD.MNU, as an ASCII text file. You can use a word processor or text editor to change the entries in the ACAD.MNU file to include your own combinations of commands, command options, and system variable settings. The AutoCAD menu file is a detailed list of the various commands available to you. The menu file is organized in sections; each section controls a separate function, such as the pointing device buttons, the pull-down menus, the screen menu, the digitizing tablet menu, and the toolbars.

The ability to modify the existing menu file and to create new menu files gives you a surprising amount of power to control the way AutoCAD operates. However, customizing the menus requires knowledge of the menu organization and special codes (characters) that are understood by AutoCAD for performing specific tasks. Although specific knowledge is required for customizing the AutoCAD menus, it is easy to learn how to control the menus by making small changes to the existing menus or creating simple menus “from scratch.”

A menu file is an ASCII text file that executes commands based on the tasks you select on the menu. By customizing part or all of the menu system, you can tailor AutoCAD to work for you. Customizing menus is a means to automate repetitive tasks and maintain consistency on drawings throughout an organization. Customizing the menus for your specific tasks can make you much more productive.

MENU FILE TYPES AND MENU LOADING

Menu File Types

A menu file is usually a group of files working together to define the appearance and functionality of the particular menu you have loaded with your drawing file. A number of file extensions are associated with the basic AutoCAD menu. The following table defines the various file extensions that may be associated with the menu:

<u>File Extension</u>	<u>Description</u>
MNU	Menu template file: This ASCII file contains the AutoCAD command strings and menu macros.
MNC	Compiled menu file: This binary file contains the command strings and menu syntax that defines the appearance and functionality of the menu.
MNR	Menu resource file: This binary file contains the bitmaps used by the menu.
MNS	Menu source file: The source menu file is created by AutoCAD.
MNT	Menu resource file: This file is generated when the MNR file is unavailable, for example, when the MNR is read-only.
MNL	Menu LISP file: This ASCII file contains AutoLISP expressions that are used by the menu file and are loaded into memory when a menu file with the same name is loaded.

Before you begin creating custom menus, you should make a backup of the ACAD.MNU and the ACAD.MNL files. You can then make alterations in these files to create your own custom menu files. Each new menu you create should have its own MNL file (using the same filename) to evaluate and load the necessary LISP routines to interact with the other menu files.

Loading Menu Files

MENU

Pull-down Menu	COMMAND (TYPE)	ALIAS (TYPE)	Short-cut	Screen (side) Menu	Tablet Menu
...	<i>MENU</i>

The *Menu* command loads a menu file. Normally this command is used to load a base menu, complete with all menu areas such as pull-down menus, toolbars, cursor menu, image tile menus, screen menus, etc. The *Menu* command produces the *Select Menu File* dialog box (Fig. 46-1). Enter or select a menu file name. When *FILEDIA* is set to 0, the command line issues the following prompt.

Enter menu file name or [. (for none)] <C:\Program Files\Acad2000\Support\acad>:

The name of the most recently loaded menu file is stored in the system registry in AutoCAD 2000. The name of this menu is also stored with the drawing file in the *MENUNAME* system variable for backward compatibility. The last menu used is loaded each time you restart AutoCAD. In order to speed up initial drawing loads, beginning with R14, AutoCAD no longer reloads the menu file each time you open a drawing file in any one AutoCAD session.

Figure 46-1



MENULOAD

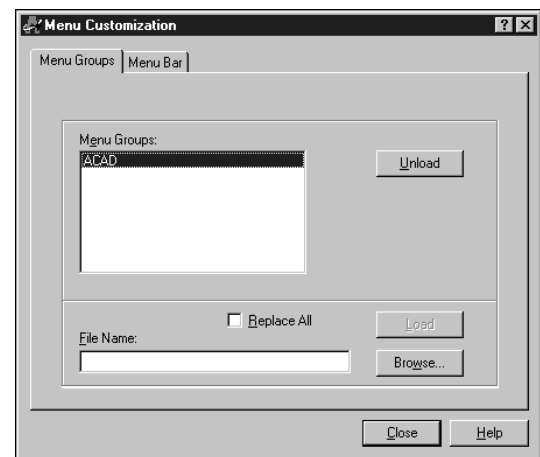
Pull-down Menu	COMMAND (TYPE)	ALIAS (TYPE)	Short-cut	Screen (side) Menu	Tablet Menu
<i>Tools Customize Menus...</i>	<i>MENULOAD</i>	<i>TOOLS 2 Menuload</i>	<i>Y,9</i>

Each time you restart AutoCAD or load a specific menu file with the *MENU* command, AutoCAD systematically searches for the menu file based on the library search path (see Library Search Path).

The *Menuload* command is used to load additional menus (called “partial menus”) or add and remove individual pull-down menus from the menu bar. The *Menuload* command produces the *Menu Customization* dialog box (Fig. 46-2). Use the *Menu Groups* section of this dialog box to *Load*, *Unload*, and *Browse* partial menu files. The *Menu Bar* tab is used to select pull-down menus to *Insert* into or *Remove* from the menu bar at the top of your screen. When *FILEDIA* is set to 0, the command line issues the following prompt:

Enter name of menu file to load:

Figure 46-2



MENUUNLOAD

Pull-down Menu	COMMAND (TYPE)	ALIAS (TYPE)	Short-cut	Screen (side) Menu	Tablet Menu
<i>Tools Customize Menus...</i>	MENUUNLOAD	TOOLS 2 Menuload	Y,9

The *Menuunload* command also produces the *Menu Customization* dialog box (see Fig. 46-2). Here you may also unload partial menus and menu bars. When *FILEDIA* is set to 0, the command line issues the following prompt:

Enter name of the MENUGROUP to unload:

Library Search Path

When AutoCAD attempts to locate a menu specified by the system registry, *MENUNAME* variable, or the *Menu* or *Menuload* commands, the following search path order is used:

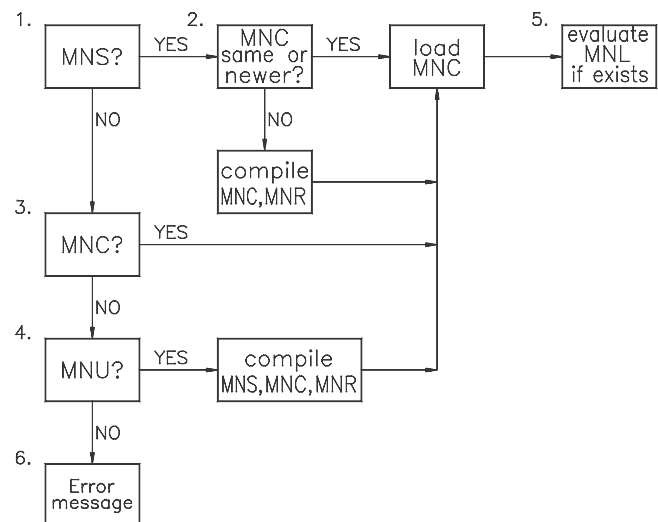
1. Current directory
2. Directory that contains the current drawing file
3. Directories listed in the Support path (for more information, see Support File Search Path in the *Installation Guide*)
4. Directory that contains the AutoCAD program files

NOTE: Two or more of these directories may be the same, depending on your drawing environment.

Steps for Loading Menu Files

The default AutoCAD menu files are named “ACAD” with the appropriate file extension, for example, ACAD.MNU, ACAD.MNS, etc. You can use any name (8 character limit) when you create a new custom menu. Assuming the menu file you are loading is called ACAD menu, AutoCAD looks for the menu using the following steps. Figure 46-3 visually describes the process.

1. AutoCAD looks for the ACAD.MNS (menu source) file following the library search procedure shown above. If found, it proceeds to step 2; otherwise, it continues with step 3.
2. After locating the ACAD.MNS (menu source) file, AutoCAD looks for the ACAD.MNC (compiled menu) file in the same directory. If it finds the ACAD.MNC (compiled menu) file with the same or later date and time as the ACAD.MNS (menu source) file, AutoCAD loads the ACAD.MNC (compiled menu) file. Otherwise, AutoCAD compiles the ACAD.MNS (menu source) file, generating a new ACAD.MNC (compiled menu) file in the same directory, and loads that file. Next AutoCAD goes to step 4.
3. If AutoCAD doesn’t find the ACAD.MNS (menu source) file during step 1, it looks for the ACAD.MNC (compiled menu) file, following the library search procedure shown above. If AutoCAD finds the ACAD.MNC (compiled menu) file, it loads that file and continues with step 5; otherwise, it moves to step 4.

Figure 46-3

4. If AutoCAD fails to locate either the ACAD.MNS (menu source) or the ACAD.MNC (compiled menu) file, it searches the library path for the ACAD.MNU (menu template) file. If the ACAD.MNU file is found, it compiles ACAD.MNC (compiled menu) and ACAD.MNS (menu source) files, then loads the ACAD.MNC (compiled menu) file and continues with step 5; otherwise, it moves to step 6.
5. When AutoCAD is finished locating, compiling, and loading the ACAD.MNU (menu template) file, AutoCAD looks for a ACAD.MNL (menu LISP) file, following the library search procedure shown above. If AutoCAD finds a ACAD.MNL (menu LISP) file, it evaluates the AutoLISP expressions within the ACAD.MNL (menu LISP) file. The ACAD.MNL file is optional.
6. If AutoCAD doesn't find any of the ACAD menu files, an error message is displayed and you are prompted for another menu file name. You need to specify the full path and file name for AutoCAD to locate the ACAD menu.

Each time you load the standard AutoCAD menu (ACAD.MNU), the standard AutoCAD LISP menu file (ACAD.MNL) is also evaluated. AutoCAD then compiles an ACAD.MNS and an ACAD.MNC file and generates the ACAD.MNR which contains the bitmaps used by the menu. The generation of the ACAD.MNR file occurs any time AutoCAD needs to compile the ACAD.MNC. (According to the *AutoCAD 2000 Customization Guide*, if AutoCAD for some reason cannot access or generate the ACAD.MNR, the ACAD.MNT will be generated.) The ACAD.MNU file is an ASCII file which contains comments and special formatting.

Sample Menu Files

Excerpt from the AutoCAD Standard Menu Template File (ACAD.MNU)

Listed below is the first section of the standard AutoCAD ACAD.MNU menu template file:

```
//
//   AutoCAD 2000 Menu
//   28 April 1997
//
//   Copyright (C) 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1994, 1996, 1997, 1998
//   by Autodesk, Inc.

.....
//   NOTE: AutoCAD looks for an ".mnl" (Menu Lisp) file whose name is
//         the same as that of the menu file, and loads it if
//         found. If you modify this menu and change its name, you
//         should copy acad.mnl to <yourname>.mnl, since the menu
//         relies on AutoLISP routines found there.
//
//
//   Default AutoCAD NAMESPACE declaration:
//
***MENUGROUP=ACAD

//
//   Begin AutoCAD Digitizer Button Menus
//
***BUTTONS1
// Simple + button
// if a grip is hot bring up the Grips Cursor Menu (POP 500), else send a carriage return
// If the SHORTCUTMENU sysvar is not 0 the first item (for button 1) is NOT USED.
$M=$(if,$(eq,$(substr,$(getvar,cmdnames),1,5),GRIP_),$P0=ACAD.GRIPS $P0=*);
$P0=SNAP $p0=*
^C^C
^B
```

A pair of forward slashes (//) indicates comments in a menu file. The ACAD.MNS file is initially the same as the ACAD.MNU without the comments or special formatting. The first 15 lines of the standard AutoCAD menu (ACAD.MNU) are comments. When AutoCAD compiles the ACAD.MNS file, only the first three and the last three lines of the file are kept as comments. AutoCAD modifies the ACAD.MNS file each time you make changes to the menu using the AutoCAD interface (such as creating custom toolbars).

Whenever you modify the toolbars (using the AutoCAD interface) and you want to keep the new changes, you should cut and paste the modified sections of the MNS file to the MNU file prior to deleting the MNS file or specifically reloading the MNU file.

Excerpt from the AutoCAD Standard Menu Source File (ACAD.MNS)

Listed below is a section of the standard AutoCAD ACAD.MNS menu source file:

```
//
// AutoCAD menu file - R:\NTEXE\RELEASE\support\acad.mnu
//

***MENUGROUP=ACAD

***BUTTONS1
$M=$(if,$(eq,$(substr,$(getvar,cmdnames),1,5),GRIP_),$P0=ACAD.GRIPS $P0=*);
$P0=SNAP $p0=*
^C^C
^B
```

The MNU and the MNS files contain the initial positioning of the toolbars. Changes to the status (docked/floating and show/hide) are stored in the system registry. When you modify the MNU file after the MNS file is generated, you must specifically load the MNU to generate new MNS and MNR files to make your changes recognized.

SHORTCUT MENUS, POINTING DEVICES, AND RELATED SYSTEM VARIABLES

Before we discuss customizing menu structure, button, auxiliary and pull-down menus, we should discuss the shortcut (right-click) menu system in AutoCAD 2000, pointing devices, and the two new system variables that affect the way menus perform. The *SHORTCUTMENU* system variable controls the right-click menus and the *MBUTTONPAN* system variable affects the action of the mouse wheel. Your understanding of the uses and settings for these system variables is essential to your ability to write customized AutoCAD menus.

Shortcut Menus and the *SHORTCUTMENU* System Variable

To display a shortcut menu in AutoCAD 2000, you can right-click your pointing device. Although most pointing devices have right and left buttons, other devices provide different or additional methods to accomplish a right-click. Some keyboards have a Shortcut Menu key you can use to display shortcut menus. See “Using Pointing Devices” in the *AutoCAD 2000 Users Guide* for information about how to right-click each pointing device.

Shortcut menus are context-sensitive. In other words, the shortcut menu that is displayed and the options that appear on it are dependent on several conditions current at the time you invoke the menu. The shortcut menu can depend on the pointer location, whether an object is selected, or if a command is

in progress. The possible areas in the AutoCAD window that can cause a specific shortcut menu to appear are listed below.

- Drawing area
- Command line
- Dialog boxes and windows such as DesignCenter
- Toolbars
- Status bar
- Model tab and layout tabs

Using Shortcut Menus in the Drawing Area

When your pointing device is located in the drawing area and you right-click, one of six shortcut menus appears: the Default menu, Edit-mode menu, Command-mode menu, Object Snap menu, Hot Grips menu, or the OLE menu (see Chapter 1, “Shortcut Menus”). The following table describes how to access the shortcut menus that are displayed when you right-click in the drawing area.

<u>Shortcut menu</u>	<u>How to access</u> <u>Menu options</u>
Default	Cancel active command, deselect all objects, and right-click <i>Copy, Paste, Pan, Zoom</i>
Edit	Cancel active command, select desired objects, and right-click Options are specific to the kind of object selected
Command	While a command is active, right-click Options specific to the command in progress and any options currently displayed on the command line
Object Snap	Press Shift and right-click Object snap options, object snap settings, and point filters
Hot Grips	Select a grip on an object, then right-click Grip editing options
OLE	Right-click an OLE object OLE object editing options

You can control the action of the Default menu, Edit-mode menu, and Command-mode menus using the *User Preferences* tab in the *Options* dialog box (explained next in “Controlling Default-mode, Edit-mode, and Command-mode Shortcut Menus”). For information about additional shortcut menus, see “Using Other Shortcut Menus.”

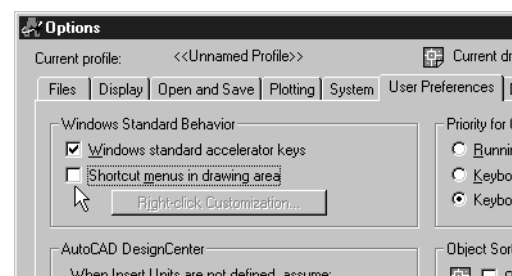
Controlling the Default-mode, Edit-mode, and Command-mode Shortcut Menus

The Object Snap, Hot Grips, and OLE shortcut menus are always turned on. However, you can control the display of the Default-mode, Edit-mode, and Command-mode shortcut menus using the *Options* dialog box. Changes you make in the dialog box to control the display of these menus are automatically stored in the *SHORTCUTMENU* system variable.

Alternately, you can make changes directly to the *SHORTCUTMENU* system variable instead of using the *Options* dialog box.

To turn off the display of the Default-mode, Edit-mode, and Command-mode shortcut menus in the drawing area, open the *Options* dialog box. In the upper-left corner of the *User Preferences* tab (Fig. 46-4), remove the check from *Shortcut menus in drawing area*.

Figure 46-4



If you want to control the Default-mode, Edit-mode, and Command-mode menus individually, select the *Right-click Customization* button in the *User Preferences* tab (see Figure 46-4, grayed-out option). This action produces the *Right-Click Customization* dialog box (Fig. 46-5). A brief explanation of the options is given here.

Default Mode and Edit Mode

Repeat Last Command

Selecting this option turns off the Default-mode and Edit-mode shortcut menus. Right-clicking has the same effect as pressing the Enter key; that is, right-clicking repeats the last command.

Shortcut Menu

This option turns on the Default-mode or Edit-mode shortcut menu when you right-click in the drawing area.

Command Mode

Enter

This option turns off the Command-mode shortcut menu. Right-clicking in the drawing area while a command is in progress has the same result as pressing the Enter key.

Shortcut Menu: always enabled

Right-clicking in the drawing area while a command is in progress always displays the Command-mode shortcut menu.

Shortcut Menu: enabled when command options are present

This option displays the Command-mode shortcut menu only when options are currently available in the command line prompt [shown in enclosed square brackets like this]. If no options are available, right-clicking has the same effect as pressing Enter.

SHORTCUTMENU

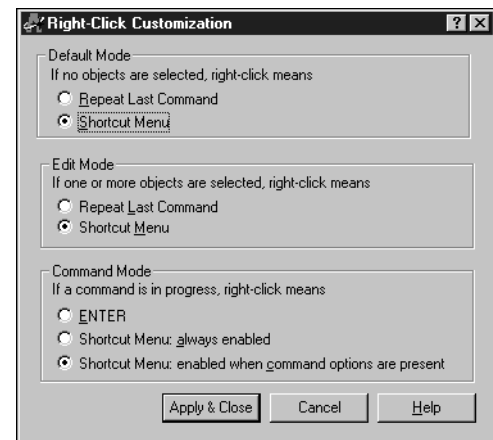
The *SHORTCUTMENU* system variable stores right-click customization settings that you specify in the *Options* dialog box as described in the previous section. You can, however, specify the right-click customization settings directly using the *SHORTCUTMENU* variable. See *AutoCAD 2000 Instructor Appendix A*, pages 1120 & 1121, for the legal bit code settings you can use. Keep in mind that understanding the purpose of the *SHORTCUTMENU* system variable is essential to your ability to write customized menus in AutoCAD.

In addition to specifying your preferences for the Default-mode, Edit-mode, and Command-mode shortcut menus using the *Options* dialog box and the *SHORTCUTMENU* system variable, you can customize the individual options that are displayed on them by writing customized menu (.MNU) files. For example, you may want to add options to the Edit shortcut menu that appears only when a certain type of object is selected. For information about writing customized shortcut menus, see “Pull-Down and Shortcut Menus.”

Using Shortcut Menus Outside the Drawing Area

In addition to right-clicking in the drawing area to display shortcut menus, you can right-click in other AutoCAD window areas to display shortcut menus. The following table describes ways to access these shortcut menus and the options that appear. These shortcut menus cannot be turned off by using the *Options* dialog box or the *SHORTCUTMENU* system variable. In other words, settings in

Figure 46-5



the *SHORTCUTMENU* system variable do not affect these menus. You can, however, customize these shortcut menus by writing customized menus (.MNU files).

<u>Shortcut menu</u>	<u>How to Access</u> <u>Menu Options</u>
Toolbar	Right-click on any toolbar Hide, display, or <i>Customize</i> (Right-click the empty area to the right of the Standard toolbar to display a list of toolbar menu groups)
Command Line	Right-click the command line or the text window <i>Recent commands, Paste to Cmd line, Copy, Copy History, Paste, Options</i>
Dialog Box or Window	Right-click individual items in most dialog boxes or windows Depending on the type of dialog box or list: <i>Rename, Delete, Copy, Paste</i> , and other file management commands and context-specific options
Status Bar	Right-click the coordinate display or any Status bar buttons <i>On, Off, Settings</i>
Model/Layout	Right-click the Model tab or any of the layout tabs <i>New layout, From template, Delete, Rename, Move or Copy, Select all layouts, Page Setup, Plot</i>

Pointing Devices and the *MBUTTONPAN* and *ZOOMFACTOR* System Variables

In AutoCAD, you can specify points in the drawing area and select commands using a pointing device such as a standard mouse, an IntelliMouse®, or a digitizing tablet. Some pointing devices, such as a mouse, may have only two buttons, while others, such as a digitizing puck, may have many buttons. AutoCAD assigns the actions of the first 10 buttons automatically, but you can reassign the action of the buttons (except the pick, or first, button) by modifying the menu (.MNU) file. The following sections explain the action of the buttons and assumes that the ACAD.MNU file contains the default settings supplied with AutoCAD.

Two-Button Mouse

On a two-button mouse, the left button is the “pick” button used to specify points on the screen or select menu options. The right button can either produce a shortcut menu or act as an Enter, depending on your pointer location and the right-click settings you specify in the *Options* dialog box or *SHORTCUT-MENU* system variable (see “Controlling Default-mode, Edit-mode, and Command-mode Shortcut Menus”). Also with a two-button mouse, holding down the Shift key while right-clicking displays the Object Snap shortcut menu.

Three-Button Mouse and *MBUTTONPAN*

With a three-button mouse, holding down the middle button and dragging the mouse accomplishes a real-time *Pan* by default. However, the middle button can either activate real-time panning or display the Object Snap shortcut menu. You can control which action the middle button has using the *MBUTTONPAN* system variable. Possible settings for *MBUTTONPAN* are as follows:

- 0 Supports the action defined in the AutoCAD menu (.MNU) file
- 1 Supports panning by holding and dragging the button or wheel

Since the middle button carries out the action as defined in the .MNU file when *MBUTTONPAN* is set to 0, you can use this button to produce the Object Snap shortcut menu (default .MNU file setting), or you

can customize the .MNU file to define some other action for the button. System pointing devices such as a mouse use the AUX1-4 sections of the .MNU file to define the button actions.

IntelliMouse, MBUTTONPAN and ZOOMFACTOR

The IntelliMouse is a two-button mouse with a small wheel between the buttons. The left and right buttons have the same functions as a standard two-button mouse. The wheel, however, can be rotated to real-time *Zoom* in your drawing and can be held down to real-time *Pan* in the drawing without using any commands.

The wheel is designed to rotate in small increments rather than rotating smoothly. Each increment in the wheel rotation changes the *Zoom* by some regular amount (about 10% by default). The *ZOOMFACTOR* system variable controls the amount of this incremental change. However, the amount of change is inversely proportional to the value. In other words, the higher the number, the smaller the change.

On the IntelliMouse, you can click the wheel as well as rotate it. If you click the wheel, it has the same action as the middle button on a three-button mouse; that is, it performs a real-time *Pan*. As with a three-button mouse, you can change the *MBUTTONPAN* system variable setting to 0 to cause the wheel button to display the Object Snap shortcut menu instead. Also, you can customize the .MNU file to define another action for the wheel button when *MBUTTONPAN* is set to 0 (see “Three-Button Mouse and *MBUTTONPAN*”).

Tablet

You can use a digitizing tablet to accomplish several tasks, such as selecting commands from the tablet or selecting and drawing objects on the screen. The tablet’s pointing device, which you use to select an object, can be a puck or a stylus. The “crosshairs” cursor on the screen follows the movement of the pointing device in the drawing area of the tablet.

The digitizing tablet pointing device uses a different section of the .MNU file to define its actions than that used by the system mouse. Up to 10 buttons can be defined. By default, the first button, as with any other device, is the pick button, which is used to specify points on the screen. The second button either displays a shortcut menu or issues an Enter, depending on the pointer location and the right-click settings you specified in the *Options* dialog box for the Default-mode, Edit-mode, and Command-mode shortcut menus. Pressing Shift+button 2 displays the Object Snap shortcut menu.

The actions of the digitizer pointing device are defined in the *BUTTONS* section of the .MNU file. You can assign other actions to the buttons by customizing the .MNU file. You can also customize the *TABLET* sections of the .MNU file to define specific actions when a cell (particular row and column) on the tablet is selected with the pointing device.

MENU STRUCTURE

Menu Sections

The menu is made up of several sections. Section labels are in the format of ****section_name*. The following table shows the standard AutoCAD menu section labels and their associated menu areas.

<u>Menu Section</u>	<u>Use of Label</u>
***MENUGROUP	Menu file group name
***BUTTONS1-4	Digitizer or other Non-System Pointing Device menu
***AUX1-4	System Pointing Device menus

***POP0	Cursor menu
***POP1-11	Pull-down menus
***POP500	Grips Cursor menu

Note: The following menu sections, POP501 through POP511, are initiated when the *SHORTCUTMENU* system variable is set to a value other than 0. Refer to the previous section titled "Shortcut Menus, Pointing Devices, and Related System Variables."

***POP501	Shortcut menu Default Mode
***POP502	Shortcut menu Edit Mode
***POP503	Shortcut menu Command Mode
***POP504	Shortcut menu Edit Mode - Dimensions
***POP505	Shortcut menu Edit Mode - Viewports
***POP506	Shortcut menu Edit Mode - Xrefs
***POP507	Shortcut menu Edit Mode - Mtext
***POP508	Shortcut menu Edit Mode - Text
***POP509	Shortcut menu Edit Mode - Hatch
***POP510	Shortcut menu Edit Mode - Polylines
***POP511	Shortcut menu Edit Mode - Splines
***TOOLBARS	Toolbar menu
***IMAGE	Image Tile menus
***SCREEN	Screen menus
***TABLET	Tablet menu area
***HELPSTRINGS	Text that is displayed in the Status bar when your cursor passes over a pull-down menu item, cursor menu item, or a toolbar icon
***ACCELERATORS	Accelerator key definitions

Each section of the menu controls the way AutoCAD responds to that particular device or area of the Drawing Editor. When you make a menu selection in AutoCAD from one of the menu areas, AutoCAD responds by following the instructions (commands) listed in that particular line of the menu.

Menu Item Syntax

A menu item is generally one line in a menu that performs one action in AutoCAD when selected from the menu. In general, menu item syntax is consistent across all sections of the menu that allow menu items. Each menu item normally is comprised of these elements: a name tag, a label, and a menu macro. A menu item is generally on one line in the following format.

```
name_tag      label      menu_macro
```

Name Tags

A name tag is generally the first component listed in a menu item. It consists of a combination of alphanumeric characters and an underscore (_) character. The name_tag should be a unique string within the menu file. The functionality provided by the name tags are:

Name_tags link menu items and toolbar buttons to their associated Help strings which are displayed on the Status bar line.

Name_tags link accelerator keys to the corresponding POP menu macros.

Name_tags enable menu items to switch from one state to another. Menu items can be enabled/disabled and checked/unchecked with menu macros and AutoLISP.

Name tags are not allowed in the BUTTONS, AUX, and IMAGE sections of the menu. They serve no purpose in the SCREEN and TABLET menu sections.

Item Labels

Item labels define the words that are displayed in AutoCAD for a pull-down or screen menu. Each menu section uses a different format for item labels. In the menu file, item labels are enclosed in brackets [label]. Menu sections that have no interface for displaying information to the user, such as the BUTTONS, AUX, and TABLET sections, do not require item labels. You may wish to use item labels in the BUTTONS, AUX, and TABLET sections for programming notes, but they serve no other purpose. The use of each of the menu section item labels is shown in the following table. This table is reprinted from the AutoCAD 2000 *Customization Guide*.

<u>Menu Section</u>	<u>Use of Item Label</u>
POP n	Defines the content and formatting of pull-down and shortcut menu selections.
TOOLBARS	Defines the toolbar name, status (floating or docked and hidden or visible), and position. Also, it defines each button and its properties.
IMAGE	Defines the text and image displayed in the image tile menus.
SCREEN	Defines the text that is displayed in the screen menus.
HELPSTRINGS	Defines the Status line help related to POP and TOOLBAR menu items.
ACCELERATORS	Associates keyboard action with menu macros.

The proper syntax for each of the labels is described in the particular menu section descriptions.

Menu Macros

Menu macros are single commands or a string of commands in a menu that complete tasks for you. When you include menu macros in the menu item, you must use the proper syntax, otherwise you may get unexpected results. Every character in a menu macro is significant. Even blank spaces serve a vital purpose and must be located properly. With each revision and upgrade of AutoCAD, command syntax might change and may require you to make minor changes to your customized menus. Sometimes with new releases, command names change. One example of this is the LWPOLYLINE object, or "optimized polyline," which was new with Release 14.

Menu macros often involve using special characters. The special characters, which AutoCAD recognizes, are shown in the following table. The table is reprinted from the AutoCAD 2000 *Customization Guide*.

Special Characters for Menu Macros

<u>Character</u>	<u>Description</u>
;	Issues Enter
^M	Issues Enter
^I	Issues Tab
Spacebar	Enters a space; blank space between command sequences in a menu item is equivalent to pressing the Spacebar
\	Pauses for user input
-	Translates AutoCAD commands and key words that follow
+	Continues menu macro to the next line (if last character)
=*	Displays the current top level image, pull-down, or cursor menu
*^C^C	Prefix for a repeating item
\$	Special character code that loads a menu section or introduces a conditional DIESEL macro expression (\$M=)
^B	Toggles <i>Snap</i> on or off (Ctrl+B)
^C	Cancels command (Esc)
^D	Toggles <i>Coords</i> on or off (Ctrl+D)
^E	Sets the next isometric plane (Ctrl+E)
^G	Toggles <i>Grid</i> on or off (Ctrl+G)
^H	Issues a backspace
^O	Toggles <i>Ortho</i> on or off (Ctrl+O)
^P	Toggles <i>MENUECHO</i> on or off
^Q	Echoes all prompts, status listings, and input to the printer (Ctrl +Q)
^T	Toggles <i>Tablet</i> on or off (Ctrl+T)
^V	Changes current viewport (Ctrl+V)
^Z	Null character that suppresses the automatic addition of Spacebar at the end of a menu item

Interpreting a Macro

Built into AutoCAD is the ability to add a space at the end of a menu macro:

```
ID_Line    [&Line]^C^C_line(blank space)
```

AutoCAD interrupts the menu macro shown above as:

```
name_tag    = ID_Line
```

```
item label  = Line (with the letter L underlined for use as an accelerator key combination)
```

<code>^C^C</code>	=	(cancel any currently active command)
<code>underscore (_)</code>	=	automatic translation for foreign language support
<code>line</code>	=	Standard AutoCAD <i>Line</i> command
<code>blank</code>	=	AutoCAD automatically adds a blank space at the end of a menu macro line

AutoCAD automatically adds a blank space (like pressing the Spacebar) at the end of a line when no other menu macro item terminator is at the end of the line.

Special Menu Macro Line Terminators

Some of the special menu macros line terminators are the semi colon (;), the backslash (\), all of the control characters, and a plus sign (+). These terminators appear at the end of a menu macro line. AutoCAD does not add a space after any of these line terminators.

`;` = A semicolon is interpreted as pressing the Enter key on the keyboard.

The command to execute pressing the Enter key would simply be `ID_Enter [;&Enter];`

`\` = A backslash causes the menu macro to pause for user input.

A sample of good use of the backslash is in the `**TABLET3` section of the AutoCAD menu where you can reconfigure your digitizing tablet simply by picking the corners of each area of the tablet menu. The following section of code is taken from the `ACAD.MNU`.

```
**TABLET 3
...
[Config _]_cfg 4 \25 9 \11 9 \9 13 \25 7 _y \_y
```

Each one of the backslashes in the menu macro is a pause for you to select one of the corners on the particular tablet menu or screen pointing area. Note the number of rows and the number of columns are automatically input for you. The first “y” is in response to the prompt: “Do you want to respecify the Fixed Screen Pointing Area?” The second “y” is in response to the prompt: “Do you want to specify the Floating Screen Pointing Area?”

NOTE: Since the backslash (\) pauses for user input, whenever you need to specify a directory path you should use a forward slash (/) for any and all directory path delimiters in a menu file (for example, /directory/filename).

When you use the backslash to pause for user input, continuation of the macro will be delayed under the following conditions:

1. When input of a point is expected, the user may set Object Snap modes prior to entering the data to the prompt for a point.
2. When *X/Y/Z* point filters are used, the macro is suspended until the entire point is entered.
3. For the *Select* command only, the menu item does not resume until object selection is completed.
4. When the user responds with a transparent command, the menu macro remains suspended until the transparent command is completed and the original requested input is received.
5. When the user responds by choosing another menu item to supply options, the menu macro remains suspended until the options portion is processed and the original requested data is received.
6. When the user responds by selecting a completely different command, most likely the menu macro will be terminated.

Canceling a Command

Canceling a command is controlled by two Ctrl+Cs (^C^C) or the Escape key (Esc). Typically, a ^C^C is at the beginning of all menu macros to make sure that no previous incomplete commands are active when a new command is executed. Two Ctrl+Cs (^C^C) are common in the AutoCAD standard menu. The reasoning is that it takes two Cancels to exit the *DIM* command; therefore, it has become standard practice to use a pair of ^C^C to verify all previous commands have been terminated.

Menu Echoes and Prompts

Menu echoes and prompts are controlled by Ctrl+P (^P). When you enter commands at the keyboard, the characters you type are echoed to the Command: prompt line. The same is true when the commands are read from a menu. AutoCAD uses the special character Ctrl+P (^P) to toggle the menu echoing feature On or Off to the command line. The echoing of commands can also be controlled by the system variable *MENUECHO*.

Foreign-Language Support

The underscore (_) is the special character to enable foreign-language support for a menu. The standard AutoCAD menu makes extensive use of an underscore placed immediately before the command names to enable automatic translation between languages of AutoCAD. The portion of the *File* pull-down menu shown below has two examples: the underscore preceding the word *new* and the underscore preceding the word *open*.

```
***POP1
**FILE
ID_MnFile  [&File]
ID_New     [&New...\tCtrl + N]^C^C_new
ID_Open    [&Open...\tCtrl + O]^C^C_open
```

Control Characters in Menu Macro Items

You can enter control characters to the command string by placing items on the menu that send ASCII control characters to the system. An example of this is the Ctrl+C (^C). This sequence issues a Cancel or Escape command to AutoCAD. When entered through the keyboard the Ctrl+C (^C) is a *Copyclip* as defined by the Windows system interface.

AutoCAD recognizes the following non-alphabetic control characters:

```
^@    (ASCII code 0)
^[    (ASCII code 27)
^\    (ASCII code 28)
^]    (ASCII code 29)
^^    (ASCII code 30)
^_    (ASCII code 31)
```

The caret character (^) maps to the Ctrl key on the keyboard when used in a menu macro. You can use this feature in your menu macros to toggle settings *On* or *Off*:

```
[GridFlip]^G
[OrthoFlip]^O
[SnapFlip]^B
```

The caret character (^) is also used to cancel a command, that is, ^C^C. Remember it takes two ^Cs to exit a *DIM* command:

```
[*Cancel*]^C^C
```

You may need to send one or more characters to the command prompt line, but do not want AutoCAD to send them as final input. You can use the special characters Ctrl+H to accomplish this delay for addi-

tional user input. Ctrl+H is the ASCII code for a Backspace character. AutoCAD uses a numeric keypad in the following two examples:

```
[1]1x^H
[2]2x^H
[3]3x^H
```

When you choose one of these items, the appropriate digit is entered. Another character follows (the letter x in this case), and that character is removed by ^H (backspace). Each of these menu items ends with a control character, and AutoCAD does not add a space or Enter to such items. Thus, you can choose [2], [2], [3], [1] to construct the input 2231. Press Enter to enter the completed number.

The following sample menu items perform the function as shown above using the backslash:

```
[1]1\
[2]2\
[3]3\
```

The second method is easier to implement and in most cases produces the same result. A command issued while a menu pause is active might not function as expected, so the first method is recommended.

Support for Long Menu Macros

You can add support for long menu macros by using the Plus (+) symbol. When you have a menu macro that extends beyond one line, you can use the Plus (+) symbol as the last character in the line to continue the macros to the next line. There is no limit to the number of lines to use for a single macro. Line continuations are not preserved in the AutoCAD MNS file. The following example of a line continuation shows the toolbar *Copy* command in the **TB_MODIFY menu section of the AutoCAD menu:

```
**TB_MODIFY
ID_TbModify
ID_Erase
ID_Copy [_Button("Copy Object",....., ICON_24_COPYOB)]$M=$(if,$(eq,+
$(substr,$(getvar,cmdnames),1,4),grip),_copy,^C^C_copy)
```

Repeating Menu Macros

Sometimes it is useful to allow the user to repeat a menu macro. To have a menu macro repeat until the user intentionally cancels the macro, put an asterisk (*) immediately following the item label. When used in combination with the *Single* option of object selection, it makes for a very friendly repeating command. The *Move'em* menu macro below enables the user to move one object or a selection set of objects to a new location in the drawing. The command automatically repeats until the user intentionally cancels the command:

```
[Move'em]*^C^Cmove Single
```

The same syntax can be beneficial for many other commands like *Copy*, *Erase*, *Stretch* (use with *Single Crossing*), *Rotate*, *Scale*, *Trim*, and *Extend*.

Menu Groups

The term ***MENUGROUP refers to a label. The MENUGROUP section contains no menu items. The MENUGROUP label must precede all menu sections in the menu. Typically, a menu has one MENUGROUP. The AutoCAD menu has one MENUGROUP that labels all other menus in the AutoCAD menu with the following syntax:

```
***MENUGROUP=ACAD
```

A simple way to refer to the POP2 (*Edit*) menu of the standard AutoCAD menu would be by MENU-GROUP=ACAD and the MENU-NAME=POP2 or by MENU-GROUP=ACAD and MENU-ALIAS=EDIT.

Now that AutoCAD uses a MENUGROUP label, a name_tag must be unique only within the menu file in which the name_tag is defined.

Menu group names combined with POP menu names or aliases provide the following functionality:

Interactive loading and unloading of partial menus
(see the *Menuload* command)

Control of pull-down menu display and layout from menu macros or AutoLISP
(see the *Visual LISP Developer's Guide*)

Menu Swapping

AutoCAD supports menu swapping of the same type menu (AUX n for another, BUTTONS n for another, etc.) in the menu sections of an active BUTTONS, AUX, POP, SCREEN, or TABLET menu. The resulting menu content can be that from another section or submenu in the base menu, or it can come from a currently loaded partial menu (see Buttons and Auxiliary Menus).

BUTTON AND AUXILIARY MENUS

Whether you are using the Button (**BUTTONS n) or the Auxiliary (**AUX n) menus depends on your pointing device. The system mouse uses the Auxiliary menus. A digitizer puck or any other input device uses the Buttons menus. The format of the two menus is identical:

<i>Menu_Section</i>	=	Required
<i>Section_Alias</i>	=	Optional, one or more allowed
<i>Section_Label</i>	=	Not allowed
<i>Name_tags</i>	=	Not allowed
<i>Labels</i>	=	Not required but are allowed and used for short comments
<i>Menu Macros</i>	=	Needed to execute commands

Each line in the BUTTONS n or AUX n menu corresponds to a button on your puck/input device or on the mouse. A three-button mouse typically has button # 1 on the left, button # 2 on the right, and button # 3 in the middle. Button # 1 is always the PICK button and cannot be reassigned to another menu macro; therefore, no line in the menu defines the button # 1 function.

Since a mouse is considered a system pointing device, it uses the AUX n menus. Button # 1 (left) is the PICK button. Button # 2 (right) would execute the menu macro one line below the menu section label **AUX1. Button # 3 would execute the menu macro on the second line below the menu section label **AUX1.

Using a Three-Button Mouse as an Input Device

// AUX1 are executed when you press the buttons on your system input device:

```

***AUX1                                // The menu section label
[Button # 2];                          // Press Button # 2 and send an Enter
[Button # 3]^C^C                        // Press Button # 3 and send an Escape/Cancel
[Button # 4]^B
[Button # 5]^O
[Button # 6]^G
[Button # 7]^D
[Button # 8]^E
[Button # 9]^T

```

Using a Four-Button Digitizer Puck as an Input Device

// BUTTONS1 are executed when you press the buttons on your input device:

```

***BUTTONS1                // The menu section label
[Button # 2 ];             // Press Button # 2 and send an Enter
[Button # 3 ]$P0=SNAP $p0=* // Press Button # 3 and call the OSNAP Pop menu
[Button # 4 ]^C^C         // Press Button # 4 and send an Escape/Cancel
[Button # 5 ]^B
[Button # 6 ]^O
[Button # 7 ]^G
[Button # 8 ]^D
[Button # 9 ]^E
[Button # 10]^T

```

AutoCAD ships with menu sections labeled BUTTONS1–4 and AUX1–4. To access the BUTTONS2–4 or AUX2–4 menu areas, use the key combinations shown in the table below.

Menu Sections and Associated Button Combinations to Execute the Menu Macro

<u>Menu Section</u>	<u>Key/Button Combination</u>
AUX1 and BUTTONS1	Button
AUX2 and BUTTONS2	Shift + Button
AUX3 and BUTTONS3	Ctrl + Button
AUX4 and BUTTONS4	Ctrl + Shift 1 Button

Under the Key/Button Combination column, the word “Button” refers to any single button on your input device (e.g., button #1 or button # 2 or button # *n*). Your pointing device can recognize as many lines in the BUTTONS or AUX menus as the number of assignable buttons.

Button and Auxiliary Menu Swapping

Button and Auxiliary menus allow menu swapping. BUTTON and AUX menu swapping gives you access to more commands at your fingertips by allowing you to dynamically swap the commands assigned to your input device. BUTTON and AUX menus recognize what are called “menu section aliases.” The format for a menu section alias is ***alias*. In previous versions of AutoCAD, the double asterisk (**) designated menu subsections, which are no longer supported. The following sample of code shows two methods to swap the Auxiliary menus in and out of ***AUX2. The ***AUX3 section swaps the menu sections ***AUX2 and ***AUX5 by using menu section aliases. The ***AUX4 section swaps the ***AUX2 and ***AUX5 menu sections by using the menu section names.

```

***MENUGROUP=ACAD

***AUX1
// Simple button
// if a grip is hot bring up the Grips Cursor Menu (POP 17), else send a carriage return
$M=$(if,$(eq,$(substr,$(getvar,cmdnames),1,5),GRIP_),$P0=ACAD.GRIPS $P0=*)
$P0=SNAP $p0=*
^C^C
^B
^O
^G

```

```

***AUX2
**DEFAULT
// Shift + button
[Shift+Right]^C^Cline
[Shift+Middle]^C^Cline \

***AUX3
[CTRL+Right]^C^C^P(menucmd "a2=acad.video") ^P // Alias for ***AUX5
[CTRL+Middle]^C^C^P(menucmd "a2=acad.default") ^P // Alias for ***AUX2
//$P0=SNAP $p0=*

***AUX4
[CTRL+Shift+right]^C^C(menucmd "A2=acad.aux5") // Uses menu section name
[CTRL+Shift+Middle]^C^C(menucmd "A2=acad.aux2") // Uses menu section name

***AUX5
**VIDEO
[Button #2 Right]^C^C'_zoom ;
[Button #3 Middle]^C^C'_pan
$P0=SNAP $p0=*

```

Since all of these menu sections are in the *****MENUGROUP** called ACAD, you do not have to list the **MENUGROUP** name in the commands. The **MENUGROUP** name is in the macros because you can swap menu sections from the same **MENUGROUP** or a different **MENUGROUP**. Only like menu sections can be swapped (**BUTTON1** with **BUTTON n** or **AUX1** with **AUX n**). Another reason for including the **MENUGROUP** name is because when you or someone else needs to revise the menu in the future, it will be much easier to understand the macros. You can add as many uniquely named *****BUTTON** and *****AUX** sections as you wish in a given menu. One thing to keep in mind is that each added section takes up additional memory and system resources. Excessive menu sections in a single menu file could require more resources than some systems have available, which could lead to unpredictable results.

Special Use of the Backslash by **BUTTON** and **AUX** Menus

AutoCAD receives both the button number and the cursor screen coordinate values when you select a button on a multi-button pointing device. You can use this to your advantage when writing menu macros. The *Line* command shown previously in *****AUX2** is an example of how you can use the backslash (\) to your advantage. The syntax “[Shift+Right]^C^Cline” issues a *Line* command and waits for the user to select the beginning point of the line (From point:). The “[Shift+Middle]^C^Cline\” issues a *Line* command and uses the coordinates from the cursor location on the screen as the first point (From point:) of the *Line* command. By using the backslash to capture coordinates and pass them to the system, you can make the menu macros work for you.

PULL-DOWN AND SHORTCUT MENUS

Shortcut Menus

The shortcut menus are displayed as cascading menus (also known as walking or hierarchical menus). Using shortcut menus, you can create a logical layout of menus without swapping menu areas. The shortcut menu can provide quick access to often-used menu items such as Object Snap modes, and switches to the Grips Cursor (*****POP500**) menu when a grip is “hot” (see Chapter 23, Grip Editing). Shortcut menu items are similar to items in other menu sections, so you can define menu macros similar to the way you define standard screen or tablet menus.

Shortcut menus are defined in the *****POP0** and *****POP500** through *****POP999** sections. The shortcut menus in the upper range are also referred to as context menus. A shortcut menu can contain up to 499 menu items. Limits include all menus in the hierarchy. If menu items in the menu file exceed these limits, AutoCAD ignores the extra items. If a shortcut menu is longer than the available space on the graphics screen, it is truncated to fit on the screen. The shortcut menu is always displayed at or near the crosshairs or cursor on the graphics screen, text window, command line, or toolbar areas. The syntax for this **POP n** menu section is the similar to the pull-down **POP n** menu section except that the shortcut menu title is not included in the menu bar. The shortcut menu title is not displayed at all, but you must still enter a dummy title. Access to the shortcut menu is through the **\$P0=* menu command**, which can be issued by another menu item (such as a **BUTTONS n** menu item) or by an AutoLISP or ObjectARX program. While the shortcut menu is active, the menu bar is not available.

Pull-Down Menus

The pull-down menus are displayed as cascading menus (also known as walking or hierarchical menus). Thus, you can create a logical layout of menus without swapping menu areas. Pull-down menu items are similar to items in other menu sections, and you define menu macros similar to the way you define standard screen or tablet menus. Pull-down menus are defined in the *****POP1** through *****POP499** menu sections. A pull-down menu can contain up to 999 menu items. Limits include all menus in the hierarchy. If menu items in the menu file exceed these limits, AutoCAD ignores the extra items. Pull-down menu items are always “pulled-down” from the menu bar. If the pull-down menu is taller than the available graphics screen space, it is truncated to fit on the graphics screen. Access to the pull-down menu is through the **\$P n =* menu command**.

Creating Pull-Down and Shortcut Menus

The shortcut (*****POP0** & *****POP500** -**POP999**) and the pull-down (*****POP1**-**POP499**) menu sections control the pull-down menus. These menus are known as cascading menus. Presentation of these menus is generally in a logical order. When AutoCAD loads a menu file, it uses the *****POP1**-*****POP16** menu sections to construct the menu bar at the top of the graphics screen. The *****POP16+ n** (where $n \geq 1$) menu sections are loaded, but are not active on the menu bar. You can use the *Menuload* command to load any additional Pull-down menus. If no ****POP n** sections are defined, AutoCAD inserts the default *File* and *Help* menus.

Creating Shortcut Menus

The following section is reprinted from the *AutoCAD 2000 Users Guide*.

Shortcut menus are defined using the same syntax as pull-down menus. The **POP0** menu section defines the default Object Snap shortcut menu, and the menu sections from **POP500** through **POP999** are used for context shortcut menus.

AutoCAD references the context shortcut menus by their alias (as in ****GRIPS**) and uses them in specific situations. The actual **POP n** number is not important, but the alias names must follow the proper naming conventions in order to be used. The following aliases are reserved for use by AutoCAD.

GRIPS

The content of this menu defines the Hot Grip shortcut menu (right-click in the drawing area while an object's grip is hot [selected]).

CMDEFAULT

The content of this menu defines the Default-mode shortcut menu (right-click in the drawing area while no command is active and no objects are selected).

CMEDIT

The content of this menu defines the Edit-mode shortcut menu (right-click in the drawing area while one or more objects are selected [but no grips are hot], and no command is active). In addition to the content of the CMEDIT menu, the appropriate object menu (if it exists) is inserted into this menu when one or more of a specific object type are selected. Object menus use the following naming convention:

OBJECT(S)_objectname

If a single object is selected, the OBJECT_objectname menu is used, and if more than one of the same object is selected, the OBJECTS_objectname is used. If no OBJECT_objectname is available, AutoCAD uses the OBJECTS_objectname menu (if it exists).

The object name is the DXF name of the object in all cases except the insert object. To differentiate between a block insertion and an xref, use the names BLOCKREF and XREF. The following AutoLISP code defines the command OTYPE, which reports the selected object's DXF name.

```
(defun C:OTYPE() (cdr (assoc 0 (entget (car (entsel))))))
```

CMCOMMAND

The content of this menu defines the Command-mode menu (right-click in the drawing area while a command is active). In addition to the content of the CMCOMMAND menu, the command line options (keywords within the square brackets) are inserted into this menu.

Like the CMEDIT menu, the CMCOMMAND menu can have context-sensitive information added to it. Any menu named COMMAND_commandname is appended to the CMCOMMAND menu. The text of commandname can be any valid AutoCAD command, including any custom-defined or third-party commands.

To make this work with a hyphen-prefixed command (such as -INSERT), you need to name the menu COMMAND_-INSERT.

Special Characters for Pull-down and Shortcut Menus

<u>Character</u>	<u>Description</u>
--	Item label that expands to become a separator line in the pull-down and shortcut menus (when used with no other characters).
+	Continues the macro to the next line (if last character).
.->	Label prefix that indicates that the pull-down or shortcut menu item has a submenu.
<.-	Label prefix that indicates that the pull-down or shortcut menu item is the last item in the submenu.
<.-<.-	Label prefix that indicates that the pull-down or shortcut menu item is the last item in the submenu and terminates the parent menu. (One <.- is required to terminate each parent menu.)
\$(Enables the pull-down or shortcut menu item label to evaluate a DIESEL string macro if \$(are the first characters.

~	Label prefix makes a menu item unavailable.
!	Label prefix that marks a menu item with a check mark.
&	An ampersand placed directly before a character specifies that character as the menu accelerator key in a pull-down or shortcut menu label. For example, S&le displays as <u>S</u> ample.
/c	Specifies the menu accelerator key in a pull-down or shortcut menu label. For example, /aSample displays as <u>S</u> ample.
\t	Specifies that all label text to the right of these characters is pushed to the right side of the menu.

NOTE: The only non-alphanumeric characters that can be used as the first character in a menu label are those listed above. Non-alphanumeric characters not listed in the previous table are reserved for future use as special menu characters.

Pull-down and Shortcut Menu Syntax

<i>Menu_Section</i>	=	Required
<i>Section_Alias</i>	=	Optional, One or more allowed
<i>Section_Label</i>	=	Required (For Pull-down menus this is the heading shown in the menu bar. Shortcut menus do not display this required label).
<i>Name_tags</i>	=	Not required, but adds functionality (See Name Tags)
<i>Labels</i>	=	Command Label the user sees on the menu
<i>Menu Macros</i>	=	Needed to execute commands

The following example illustrates the syntax that is used to create a pull-down or cursor menu:

```

***POP6
**TOOLS
ID_MnTools      [&Tools]           // No Menu Macro Associated with Label
ID_Spell        [&Spelling]^C^C_spell
ID_MnOrder      [->Display &Order]
ID_DrawordeF    [Bring to &Front]^C^C^P(ai_draworder "_f") ^P
ID_DrawordeB    [Send to &Back]^C^C^P(ai_draworder "_b") ^P
                [--]
ID_DrawordeA    [Bring &Above Object]^C^C^P(ai_draworder "_a") ^P
ID_DrawordeU    [<-Send &Under Object]^C^C^P(ai_draworder "_u") ^P

```

In the example above, ***POP6 is the *Menu_Section* label, **TOOLS is the *Section_Alias*, and ID_MnTools is the *Name_tag* for the *Section_Label*. The [&Tools] entry is the heading shown in the menu bar. The rest of the lines are of the following format:

<u>Name Tag</u>	<u>Label</u>	<u>Menu Macro</u>
ID_Spell	[&Spelling]	^C^C_spell

The name_tags in the Pull-down menu can be used to enable and disable the entire menu or various commands within the menu. The label [&Spell] causes *Spell* to be displayed as the menu bar heading, and the letter S is underscored to indicate that Alt+S is the accelerator key for this menu item. As shown in the sample above, menu macros cannot be associated with Pull-down menu labels.

Cascading Submenus

The special characters used in Pull-down and Shortcut menu labels to indicate the beginning and end of submenus are `->`, and `<.`. These characters control the cascading of the menus and terminate all parent menus. They are always the first characters in the item label. Review of the **TOOLS** menu above shows that a submenu begins with the label `[->Display &Order]` and ends with the label `[<.-Send &Under Object]`.

You must have one returning special character (`<.`) for each and every submenu level you create; therefore, you may need to have multiple special characters (`<.-<.`) on one menu item label as shown below.

```
ID_PurgeShp      [<.-<.&Shapes]^C^C_purge _sh
```

Create Menu Label Separator Lines with Two Hyphens

In order to make reading of menus easier, AutoCAD uses the following format and two hyphens to create menu item separators `[-]`. The line created by the hyphens extends the width of the menu. You cannot select separator lines from the menu. Any menu macro assigned to a separator line is ignored.

Base and Partial Pull-down Menus

AutoCAD supports the theory of base (e.g., AutoCAD menu) and partial menus (e.g., your menu) working in combination. What this means to you is that you can create a menu with only your menu macros and commands, then use it in combination with any other menu that is supported by AutoCAD. In the past, if you wanted a complete working menu, you had to create your commands and cut and paste them to a copy of the AutoCAD menu. Now you can add or remove your partial menus anytime you wish without affecting the standard AutoCAD menu.

You can insert Pull-down menus with the following syntax:

```
$Pn=+menugroup.menuname
```

In the following example, the menu group name is `MYGROUP` and the menu name is `MYPOP`; therefore, the code to insert the partial menu on the standard AutoCAD menu bar between the `MODIFY` and `HELP` menu sections would be:

```
$P9=+mygroup.mypop
```

The same thing can be done with the AutoLISP function `menucmd`:

```
(menucmd "P9=+mygroup.mypop")
```

Similar syntax can be used to remove `MYPOP` from the menu bar:

```
$P9=-
```

The syntax `$P9=-` in a menu macro removes whatever menu is in the ninth position on the menu bar. A safer method to remove the expected menu from the menu bar is to use an AutoLISP command to do the job:

```
(menucmd "Gmygroup.mypop=-")
```

The above command searches for the menu group `MYGROUP` and the menu `MYPOP` and removes only this menu if it is found; otherwise, it takes no action.

AutoCAD uses the ***POP1 through ***POP16 for menu swapping. You may wish to always have your menus placed in the same location on the menu bar. One method that works is to always place your partial menus with the same statement using a *Pn* larger than the location where your menu will be placed. In the following command, any partial menus are loaded in the farthest position to the right on the menu bar.

Place partial menu on menu bar: (menucmd "P100=+mygroup.mypop")
 Remove specific partial menu from menu bar: (menucmd "Gmygroup.mypop=-")

By using a combination of the two commands above you can safely place and remove partial menus to and from the menu bar.

IMAGE TILE MENUS

Although not used much in AutoCAD 2000, image tile menus make use of individual slides and slide libraries to allow you to see an image of the items you are selecting. AutoCAD displays image tiles in groups of 20, along with a scrolling list box containing the associated slide names or label text. Image tile submenus may be of unlimited length. When the image tile submenu contains more than 20 images, AutoCAD provides *Next* and *Previous* buttons to enable you to scroll through the images.

One use for image tile menus is for placing blocks. You may not know the name of a *Block* you need to place in the drawing file, but often you will know what the item looks like. By combining an image tile menu of the *Blocks* you often place in drawing files with the menu macros to *Insert* the *Blocks*, block insertion becomes easy.

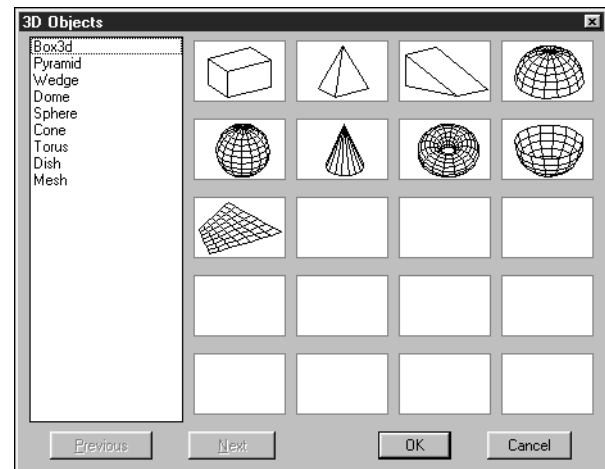
One of only two examples of the use of the image tile menus in AutoCAD 2000 can be found by selecting the *Draw* pull-down, then *Surfaces*, then *3D Surfaces...* This selection displays an image tile menu titled *3D Objects* (Fig. 46-6). When you select one of the images, the related command is invoked.

Slide Preparation

Any slide generated by AutoCAD can be used for an image tile (see Chapter 44, "Basic Customization," for more information). In order to present the best image and for slides to be consistent, you should follow these guidelines:

- | | |
|--------------------|---|
| Keep it simple | Images should be easily recognizable. If the image is to represent a complex object, you may wish to create a simple form of the object to use for the image tile. |
| Fill the image box | It is easier to recognize the images that fill the image box. Images are displayed with an aspect ratio of 1.5:1 (1.5 units wide by 1 unit high). If you create your slide in a floating viewport that has this aspect ratio, you will be assured the images will look the same on your slides. |
| Shade solids | Solid filled areas like wide <i>Plines</i> and filled solids display as outlines unless you issue a <i>Shade</i> command before making your slides. |

Figure 46-6



To set up a correctly proportioned viewport for making a slide, start with a drawing with no views:

```

Command: tilemode
Enter new value for TILEMODE <1>: 0
Regenerating layout.
Command: mview
Specify corner of viewport or
[ON/OFF/Fit/Hideplot/Lock/Object/Polygonal/Restore/2/3/4] <Fit>: 0,0
Specify opposite corner: 3,2
Regenerating model.
Command: zoom
Specify corner of window, enter a scale factor (nX or nXP), or
[All/Center/Dynamic/Extents/Previous/Scale/Window] <real time>: e
Command: mspace

```

You can create consistent slides for your image tile menu by issuing the commands from a script file. Open an ASCII text editor (for example, Windows Notepad) and type the following lines as shown:

```

undo
mark
tilemode
0
mview
0,0
3,2
zoom
e
mspace
mslide

undo
back

```

The line after the command *Mslide* is blank to save the slide file using the same name as the drawing file with an extension of SLD. Save this text file to a name (MAKESLD.SCR, for example). Copy the script file to your AutoCAD support directory. Start AutoCAD and open a drawing. To create a slide, type the command *Script* at the command prompt. At the prompt for a script file to load, type in the name of your script file: MAKESLD. AutoCAD issues the commands and creates a slide file with the same name as your drawing file, then places it in the same directory where your drawing file is located. Your drawing file will be returned to the state it was before you ran the script file.

The format for the image tile menu section is:

```

Menu_Section = Required
Section_Alias = Optional, one or more allowed
Section_Label = Required

Name_tags = Not allowed
Labels = Not required, but are used for the text in the list box
Menu Macros = Needed to execute commands

```

Image tile item labels that appear in the list box may be the slide file name or text labels. The image displayed in each image tile box may be a single slide, part of a slide library, a combination of both, or can even be from different slide libraries. The slide file name in the menu macros should be the same as the slide file name used at the command line for the *Vslide* command.

A number of image labeling options are available. Often the scrolling list box has the name of each slide in the list. You can input up to 19 characters for labels in lieu of using the slide file names.

Image Tile Menu Labeling Options

[sldname]	The sldname is displayed in the list box and the sldname image is displayed.
[sldname,labeltext]	Labeltext is displayed in the list box and the sldname image is displayed.
[sldlib(sldname)]	Sldname is displayed in the list box and the sldname image from sldlib is displayed.
[sldlib(sldname,labeltext)]	Labeltext is displayed in the list box and the sldname image from sldlib is displayed.
[blank]	When you use the word blank as an image label, a separator line is displayed in the list box and a blank image is displayed.
[(space)labeltext]	When the item label begins with a space, the text supplied as labeltext is displayed in the list box and no image is displayed. This allows you to include simple commands (words) like EXIT in the image tile without having to create slides.

One image tile menu can display another image tile menu. Since image tile menus are called sequentially, and not nested, there is no limit to the complexity of the menus you can create. The asterisk (*) character, for repeating menu macros, is not supported in the image tile menus.

Sample of an Image Tile Menu from the AutoCAD Standard Menu

```
***IMAGE
**IMAGE_3DOBJECTS
[3D Objects]
[acad(Box3d,Box3d)]^C^Cai_box
[acad(Pyramid,Pyramid)]^C^Cai_pyramid
[acad(Wedge,Wedge)]^C^Cai_wedge
[acad(Dome,Dome)]^C^Cai_dome
```

You must first load an image tile menu before you can display it:

```
$I=menugroup.menuname
```

You can then display the currently loaded image tiles menu by:

```
$I=*
```

The syntax to load and display the image menu YOURBLKS from a partially loaded menu group YGROUP is:

```
$I=ygroup.yourblks $I=*
```

Below is an example of an image tile menu used to create office furniture layouts using individual slides located in a directory already listed in the search path:

```
***MENUGROUP=OFFICE

***IMAGE
**IFURN
[Furniture]
[Chair]^C^C_insert chair
[ArmChair]^C^C_insert armchair
[Desk]^C^C_insert desk
```

Another example follows of an image tile menu used to create office furniture layouts using slides compiled in a slide library called FURN.SLB located in a directory already listed in the search path:

```
***MENUGROUP=OFFICE

***IMAGE
**IFURN
[Furniture]
[furn(Chair)]^C^C_insert chair
[furn(armchair)]^C^C_insert armchair
[furn(Desk)]^C^C_insert desk
```

Image Tile Menus Support Pull-down Partial Menus

As of Release 13 c3, image tile menus are supported in pull-down partial menus. The format for loading a partial image tile menu from a pull-down menu is:

```
[Menu Item Label]^C^C$I=menugroup.menuname
```

The following menu macro loads and displays a partial image tile menu called IFURN, which is in the menu group file called OFFICE:

```
[Furniture]^C^C$i=office.ifurn $i=*
```

SCREEN MENUS

In early versions of AutoCAD, the screen menu was the main menu interface for AutoCAD users. The screen menus section controls the text listing of commands along the right side of your graphics screen. When you initially install AutoCAD 2000, the screen menu is disabled. You can enable the Screen menu by selecting the *Tools* pull-down menu, then *Options*, the *Display* tab, and check *Display screen menu*.

The format for the screen menus section of the menu is:

<i>Menu_Section Name</i>	=	Required
<i>Submenu Sections</i>	=	Used extensively
<i>Start Number</i>	=	Optional
<i>Section_Label</i>	=	Not allowed
<i>Name_tags</i>	=	Not required and serve no purpose
<i>Labels</i>	=	Command label the user sees in the screen menu
<i>Menu Macros</i>	=	Needed to execute commands

The following information is from the beginning of the screen menus section of the ACAD.MNU file.

```
//
// AutoCAD Screen Menus
//
// There are two types of screen menus: command menus and options menus
// Command menus provide access to the lists of AutoCAD commands.
// Options menus provide access to the options available for individual commands.
//
// There are 22 lines between menu titles. This is one method for assuring that each time
// that a menu is called that it overwrites the previous menu.
//
// The organization of the command menus generally follows the organization of the pull-down
// menus.
// A command has a screen menu item only if it has a pull-down menu item.
// Command menus have, as much as possible, the same name as the equivalent pull-down menu.
// Command menus have, as much as possible, the same items in the same order as the pull-down
// menus.
//
// Command menus generally use the command name while pull-down menus offer a more
// descriptive title.
// Items in command menus that call other command menus are in upper case.
//
// Command menu names start with a number and are located after the special menus.
//
***SCREEN                               Menu Section
**S                                     Submenu
[AutoCAD ]^C^C^P(ai_rootmenus) ^P
[* * * ]$$=ACAD.OSNAP
[FILE ]$$=ACAD.01_FILE
[EDIT ]$$=ACAD.02_EDIT
[VIEW 1 ]$$=ACAD.03_VIEW1
[VIEW 2 ]$$=ACAD.04_VIEW2
[INSERT ]$$=ACAD.05_INSERT
[FORMAT ]$$=ACAD.06_FORMAT
[TOOLS 1 ]$$=ACAD.07_TOOLS1
[TOOLS 2 ]$$=ACAD.08_TOOLS2
[DRAW 1 ]$$=ACAD.09_DRAW1
[DRAW 2 ]$$=ACAD.10_DRAW2
[DIMNSION]$$=ACAD.11_DIMENSION
[MODIFY1 ]$$=ACAD.12_MODIFY1
[MODIFY2 ]$$=ACAD.13_MODIFY2

[HELP ]$$=ACAD.14_HELP

[ASSIST ]$$=ACAD.ASSIST
[LAST ]$$=ACAD.
```

Screen submenu labels use the format:

Format for submenu label: **menuname [startnum]

Sample submenu label: **01_FILE 3

The maximum length of a screen menu name is 33 characters. It can include the special characters dollar sign (\$), hyphen (-), and underscore (_). A submenu label must be on a line by itself and cannot have embedded blanks. You can specify a starting line for your submenu using the optional start number. The "3" in the sample submenu label indicates the 01_FILE submenu will begin on line 3 of the screen menu area. When the word FILE is selected on the screen menu, the *FILE* menu, which begins on line 3 of the graphics screen, overwrites the AutoCAD main screen menu. The main menu selections (*AutoCAD*) and the second line (****) stay in their respective locations on the graphics screen.

If a label is provided, the first eight characters of the label are displayed in the appropriate screen menu box. Any additional characters can serve as comments.

NOTE: The maximum number of menu items depends on your system. You can retrieve the number of screen menu boxes with the *SCREENBOXES* system variable.

Your graphics screen has a fixed number of lines for you to work with when developing screen submenus. A good rule is not to exceed 22 items in a single screen submenu. This allows the menu to be presented properly on both low and high resolution monitors. AutoCAD starts all screen submenus at or below line 3 to retain the first 2 lines (*AutoCAD* and ****) of the screen menu. The menu macro associated with the first line recalls the AutoCAD main screen menu. The second line of the screen menu calls the OSNAP submenu. When you select a submenu item on the screen menu, AutoCAD generally replaces lines 3 through 24 of the screen menu with the referenced screen submenu. If you create submenus that exceed the visible area of your display monitor, the menu items that are below the bottom viewing edge of the screen are inaccessible.

Control Submenu Swapping with the *MENUCTL* System Variable

You can control the automatic swapping of screen submenus with the *MENUCTL* system variable. When *MENUCTL* is set to 1 (On), the screen menus swap corresponding to selections made on the menus.

TABLET MENUS

There are up to four configurable tablet menu areas. The following tablet menu areas are defined in the standard AutoCAD menu.

***TABLET1 = 25 columns x 9 rows	Reserved for your customization
***TABLET2 = 11 columns x 9 rows	Commands defined by AutoCAD
***TABLET3 = 9 columns x 13 rows	Commands defined by AutoCAD
***TABLET4 = 25 columns x 7 rows	Commands defined by AutoCAD

The format for the tablet menus section of the menu is:

<i>Menu_Section Name</i>	=	Required
<i>SectionsAlias</i>	=	Optional
<i>Section_Label</i>	=	Not allowed
<i>Name_tags</i>	=	Optional, but serve no purpose
<i>Labels</i>	=	Optional, may be used for comments
<i>Menu Macros</i>	=	Needed to execute commands

Tablet area 1 [A-1]–[I-25] has been reserved for you to customize. This means there are 225 command lines on the standard tablet area 1 for you to customize. If you need more boxes, you may define up to a maximum of 32,766 items for each of the tablet areas.

The tablet boxes are ordered from left to right across the rows of the tablet, then top to bottom; therefore, A-1 (row A, column 1) is the upper-left box in the tablet 1 area and I-25 is the lower-right box in the tablet 1 area. All of the boxes are of equal size in a particular tablet area.

In the standard AutoCAD menu file, the 225 labels following the *****TABLET1** Menu section name and the ****TABLET1STD** section alias correspond to the 225 template grid boxes in tablet area 1 on your digitizing tablet:

```
// Begin AutoCAD Tablet Menu
//
// This is the TABLET1 menu. You may put your own
// macros and menu items here in these spaces.
// All of the "blank" line items actually contain a
// backslash so that no command is issued when you pick any
// of them from the tablet. Remove them if you want an Enter
// to happen when they are selected, or place your own
// macros in their place.

***TABLET1
**TABLET1STD
[A-1]\
[A-2]\
[A-3]\
```

Modifying any lines following box [I-25] is not recommended.

STATUS LINE HELP

In the standard AutoCAD menu each command associated with the pull-down and toolbar menus echoes a Help message to the Status line. These messages occur when items are highlighted (when you pass your cursor over the various commands). When an item is highlighted, the `name_tag` for that item is queried for a matching entry in the *****HELPSTRINGS** section of the menu. When a match occurs, the Help string message contained within the label is displayed on the Status line of the graphics screen.

The menu has a special section for entering HELPSTRINGS. A sample of the ACAD.MNU menu *****HELPSTRINGS** sections follows:

```
// IDs are listed in alphabetical order.
// Helps strings end with a colon, two spaces and the name of the command.
//

***HELPSTRINGS
ID_2doptim      [Set viewport to 2D wireframe: SHADEMODE 2]
ID_3darray      [Creates a three-dimensional array: 3DARRAY]
ID_3dclip       [Starts 3DORBIT and opens the Adjust Clipping Planes window: 3DCLIP]
```

The format for entering HELPSTRINGS is:

```
name_tag [Help String: Command name]
```

By adding Help string messages to your commands as you create custom menus, you provide first line Help support.

Help strings can be extremely useful when you are searching toolbar buttons for a particular AutoCAD command.

ACCELERATOR KEYS

Two methods of user-defined accelerator keys are supported by AutoCAD. The format for the first method maps a key sequence to an existing menu item:

```
name_tag [Modifier(s)+accelerator key]
// Map CTRL+Z key combination to the menu item UNDO
ID_U [CONTROL+"Z"]
```

The format for the second method maps a key sequence to a command string:

```
[Modifier(s)+accelerator key] command string
// Toggle Orthomode
[CONTROL+"L"]^O
```

The *****ACCELERATOR** menu section from the ACAD.MNU is shown below:

```
//
// Keyboard Accelerators
//
// If a keyboard accelerator is preceded by an ID string that references a menu item
// in a pull-down menu, then the keyboard accelerator will run the command referenced
// by that menu item.
//
***ACCELERATORS
// Bring up hyperlink dialog
ID_Hyperlink[CONTROL+"K"]
// Toggle Orthomode
[CONTROL+"L"]^O
// Next Viewport
[CONTROL+"R"]^V
ID_Copyclip [CONTROL+"C"]
ID_New [CONTROL+"N"]
ID_Open [CONTROL+"O"]
ID_Print [CONTROL+"P"]
ID_Save [CONTROL+"S"]
ID_Pasteclip [CONTROL+"V"]
ID_Cutclip [CONTROL+"X"]
ID_Redo [CONTROL+"Y"]
ID_U [CONTROL+"Z"]
ID_Modify [CONTROL+"1"]
ID_Content [CONTROL+"2"]
ID_dbConnect [CONTROL+"6"]
ID_VBARun [ALT+"F8"]
ID_VBAIDE [ALT+"F11"]
```

The valid modifiers are the Control and/or Shift keys. AutoCAD does not distinguish between the left and right sides of the keyboard; therefore, the left Control or Shift keys are interpreted the same as the right Control or Shift keys. You can use Control+Shift key combinations to define accelerator keys.

Special Virtual Keys

The following table is reprinted from the AutoCAD Release 14 *Customization Guide*. (Functions keys must be enclosed in quotation marks.)

<u>String</u>	<u>Description</u>	<u>Exceptions</u>
F1	F1 key	Although the F1 key can be assigned a menu macro, this is discouraged because this key is generally associated with Help. Using a modifier with this key is acceptable.
F2	F2 key	Unmodified, this toggle key switches the state of the text window.
F3	F3 key	Unmodified, this key runs <i>OSNAP</i> .
F4	F4 key	Unmodified, this toggle key switches <i>TABMODE</i> on or off.
F5	F5 key	Unmodified, this toggle key switches <i>ISOPLANE</i> on or off.
F6	F6 key	Unmodified, this toggle key switches <i>COORDS</i> on or off.
F7	F7 key	Unmodified, this toggle key switches <i>GRIDMODE</i> on or off.
F8	F8 key	Unmodified, this toggle key switches <i>ORTHOMODE</i> on or off.
F9	F9 key	Unmodified, this toggle key switches <i>SNAPMODE</i> on or off.
F10	F10 key	Unmodified, this toggle key switches <i>Polar Tracking</i> on or off.
F11	F11 key	Unmodified, this toggle key switches <i>Object Snap Tracking</i> on or off.
F12	F12 key	None
INSERT	INS key	None
DELETE	DEL key	None
ESCAPE	ESC key	Although the ESC key can be assigned a menu macro, it is discouraged because this key is generally associated with Cancel. CONTROL+ESCAPE and CONTROL+SHIFT+ESCAPE cannot be assigned a menu macro; these sequences are controlled by Windows. Using the SHIFT modifier with this key is acceptable.
UP	UPARROW key	Must be used with the CONTROL modifier.
DOWN	DOWNARROW key	Must be used with the CONTROL modifier.
LEFT	LEFTARROW key	Must be used with the CONTROL modifier.
RIGHT	RIGHTARROW key	Must be used with the CONTROL modifier.
NUMPAD0	0 key	None
NUMPAD1	1 key	None
NUMPAD2	2 key	None
NUMPAD3	3 key	None
NUMPAD4	4 key	None
NUMPAD5	5 key	None
NUMPAD6	6 key	None
NUMPAD7	7 key	None
NUMPAD8	8 key	None
NUMPAD9	9 key	None

SUMMARY OF CHANGES TO THE MENUS OF AUTOCAD 2000

1. The .MNT menu file type is added. According to the AutoCAD Release 2000 *Customization Guide*, AutoCAD 2000 will generate an ACAD.MNT file if for some reason it cannot access or generate the ACAD.MNR.
2. “Bonus Tools” is changed to “Express Tools.” The menu name is AECTMAIN.
3. The *SHORTCUTMENU* system variable is new in AutoCAD 2000. This variable affects the way menus perform when a “right-click” is made with the pointing device. Customization changes for shortcut menus made in the *User Preferences* tab of the *Options* dialog box are stored in this variable.
4. The *MBUTTONPAN* system variable is new in AutoCAD 2000. This variable affects the action of the middle mouse button or wheel of the IntelliMouse. The setting determines whether the wheel operates as a “real-time” *Pan* or displays the Object Snap shortcut menu when pressed.
5. The *ZOOMFACTOR* system variable is new in AutoCAD 2000. The *ZOOMFACTOR* setting controls the amount of the incremental change in zoom when the wheel of the IntelliMouse is rotated.
6. “Cursor Menus” are now called “Shortcut Menus.” The AutoCAD Release 14 menu section ***POP17 for GRIPS is ***POP500 in AutoCAD 2000. Shortcut menu sections are from ***POP500 to POP999.
7. CMDEFAULT is the Default-mode shortcut menu alias. The Default-mode menu appears when you right-click in the drawing area while no command is active and no objects are selected.
8. CMEDIT is the Edit-mode shortcut menu alias. The Edit-mode shortcut menu appears when you right-click in the drawing area while one or more objects are selected (but no grips are hot) and no command is active.
9. CMCOMMAND is the Command-mode menu alias. The Command-mode menu appears when you right-click in the drawing area while a command is active.

CHAPTER EXERCISES

1. Buttons and Auxiliary Menus

Using a copy of the standard AutoCAD menu, create Buttons or Auxiliary menus to do the following:

Ctrl+Button # 2 executes a *Zoom Real Time* command

Shift+Button # 2 executes a *Zoom Extents* command

Ctrl+Button # 3 executes a *Copy* command

Shift+Button # 3 executes a *Move* command

Assign any menu macro you wish to the combination Ctrl+Shift+Button # 2

Assign any menu macro you wish to the combination Ctrl+Shift+Button # 3

(HINT: Remember Button # 2 on a three-button mouse is normally the right button.)

2. Shortcut Menus

- A. Add the *Explode* command to the shortcut context menu for *Mtext* objects.
- B. Add an option to the shortcut context menu to detach *Xref* objects. (HINT: Invoke the *-Xref* command, then the *Detach* option.)

3. Partial Pull-down Menus

Create a pull-down menu titled **Layouts**. In **Layouts**, create menu macros for the following AutoCAD commands:

<u>D</u> raw	<u>E</u> dit
<i>M</i> line	<i>C</i> opy
<i>C</i> ircle	<i>E</i> rase
<i>D</i> text	<i>M</i> ove
<i>M</i> text	<i>R</i> otate
<i>R</i> ectangle	

4. Partial Image Tile Menus

- A. *Open* the **OFFICE** drawing that you created in Chapter 21 Exercises. Use the *Wblock* command to create .DWG files of each of the following *Blocks* from the drawing:

CHAIR DESK FILECAB TABLE

- B. Create a slide of each of the furniture items (*Wblocks*). Next, create a slide library of the slides and name the file **FURN.SLB**. (HINT: Use the file \ACADR14\SUPPORT\SLIDELIB.EXE to create the slide library). The slide library file (FURN.SLB) must be in your AutoCAD 2000 Support directory to function correctly as image tiles.
- C. Add a menu item to the pull-down menu you created in Exercise 2 (this chapter) called **FURNITURE** and have that menu item display the four furniture slides in an Image Tile menu titled **Office Furniture**. Once the images are displayed, you should be able to select any of the office furniture objects and insert them in your drawing at a scale of 1:1. You should not be prompted for the X and Y scale factors—only the rotation angle and insertion point—as you insert the objects into the drawing.

5. Partial Tablet Menu

Create *Block* insertion menu macros for the four *Blocks* in Exercise 3. Add the menu macros to the drawing tablet menu area 1 in the following locations:

[A-5] CHAIR [B-5] DESK [B-6] FILECAB [C-5] TABLE

Be prepared to turn in your menu macros in ASCII text format and a plot of the TABLET1 menu area you would use to *Insert* the *Blocks* into a drawing file.