

MENUS

[FAQ](#)

[Function Key Menus](#)

[Tool Boxes](#)

Historical Info

[Paper Menus](#)

[Cursor Button Menus](#)

[Side Bar Menus](#)

Frequently Asked Questions

All my task menus are no longer docked or opened correctly, how can I get them back to their default location?

If you have closed the RWD-Main Menu you can use the F1 key to open it back up. The DZine and Geopak menus can be opened with shortcut icons on the RWD-Main Menu or by pressing F2, F3, and F4. If the docking has been changed you can manually redock them.

If you are having problem docking the menus you can actually "re-install" the template RWD Preference files. Doing this will re-dock them into their default locations and also open menus that have been closed. To install the RWD Preference template file, close Microstation, and go to the desktop folder called "RWD APPS" and find the icon labeled "Replace Preference Files". Run this program and then re-open Microstation. The menus should be back in their default location. If you do not have the RWD APPS folder on your desktop, you can locate the folder and application ...\\rwddata\\apps\\newupf.exe and run this application.

Roadway Design Custom Menus don't seem to be showing up, what wrong?

Most likely you have entered Microstation without selecting our workspace. You have probably started Microstation in the untitled workspace. This occurs many times when starting Microstation by double-clicking a .dgn file in windows explorer, or when opening a .dgn within a email, or when entering Microstation from the Window Start Bar. The Windows OS is not configured to use our workspace automatically. This is why we recommend always starting Microstation from one of the icons available on the custom desktop folder called "Projects".

We have provided a program that can alter the default Windows OS command for starting Microstation. It is designed to modify the command to use one of our custom workspaces. It can be found in our custom desktop folder called RWD Apps.

Can I set up and use my own function key menu?

Yes, but is is highly discouraged. This is due to Roadway Design already having a standard function key menu installed and configured to provide some very handy commands and procedures.

Function Key Menus

What are Function Key Menus?

Function Key menus let you map the "F" Keys on your keyboard to Microstation commands. The function key menus are saved as ASCII text files. If you look at one of these text files, you can see the assigned commands listed in the file one line per command. Below is a partial example of the ASCII version of a function key file that indicates the settings for the function key:

```
$FK5.0$ 12/15/2008 11:25:13 AM  
30c,inputmanager home  
30b,accudraw dialog  
30a,dialog toolsettings  
309,inputmanager cmdbrowse
```

It is possible to edit the text file if you can keep track of the orders of which function key numbers the commands are associated with. Since the function key numbers are not listed in the text file it can be difficult to keep up with which command goes with which function key. This is why we recommend using the function key dialog editor to create or edit these text files.

Using your own custom function key menu is highly discouraged because we have some handy commands already available on our custom version.

Roadway Design Function Key Menu

rwdfunckey.mnu

This function key menu contains commonly used Microstation commands and is located in our group Function Key directory.

This menu is attached automatically by default when entering a design file. This is done with the setting of the Microstation environment variable **ms_fkeymenu** to equal the directory location of the file **rwdfunckey.mnu** in our workspace configuration file. It can also be manually attached with the key in: **am=rwdfunckey.mnu,fk** or by using the function key dialog open option.

The Roadway Design function menu command assignments are shown below:

F1	null;tasktoolbox open RWD-Main
F2	null;tasktoolbox open RWD\Dzine
F3	null;tasktoolbox open Road
F4	null;tasktoolbox open Survey MDOT
F5	choose element;accusnap autolocate toggle
F6	inputmanager menu
F7	inputmanager menu modify
F8	inputmanager menu manipulate
F9	inputmanager cmdbrowse
F10	dialog toolsettings
F11	accudraw dialog
F12	inputmanager home
Ctrl+F1	dmsg sinkall toggle
Ctrl+F2	macro clear
Ctrl+F4	mdl load fontutilities;font utilities dialog
Ctrl+F8	inputmanager popupitem primary tools,8
Ctrl+F9	inputmanager popupitem primary tools,9
Ctrl+F10	inputmanager popupitem primary tools,10
Ctrl+F11	inputmanager popupitem primary tools,11
Ctrl+F12	choose element;accusnap autolocate toggle
Shift+F1	null;set parse on;exp k newfile \$(_USTN_WORKSPACE...\DzineInterface.dgnlib
Shift+F2	null;set parse on;exp k newfile \$(_USTN_WORKSPACERO...\gpkInterface.dgnlib
Shift+F3	null;set parse on;exp k newfile \$(_USTN_WORKSPACEROOT)In...\RWDMain.dgnli
Shift+F4	null;set item toolsettings
Shift+F5	dialog cmdbrowse
Shift+F8	inputmanager popupitem attributes,8
Shift+F9	inputmanager popupitem attributes,9
Shift+F10	inputmanager popupitem attributes,10

Tool Boxes

What are Dialog Tool Boxes and Dialog Boxes?

Dialog boxes and Tool Boxes can be made with MDL programming or with the Customization Tools in Microstation. Most of the menus used in Roadway Design are done using the customization tools in Microstation.

Microstation has custom interface tools that allows users to create tool boxes and modify the appearance of tools and menus in Microstation.

DZINE TASK DIALOG NAVIGATION TOOLS

As this interface is used and tools and menus are modified or created, the changes are saved in a dgn library (.dgnlib). This file is used as a resource when you select a user interface when entering a design file.

We have created a directory under the Microstation directory structure ...rwwdata\workspace\interfaces\microstation and have called it RWD. Under the RWD directory is where the custom changes are saved. The files located in this directory are called:

RWDMAIN.DGNLIB : this file contains the custom Roadway Design standard Microstation commands menus.

DZINEINTERFACE.DGNLIB : this file contains most of the custom Roadway Design Microstation menus.

SIGNS.DGNLIB : this file contains all the custom tools to place sign cells.

Other dgnlib files located here are for the various Geopak interfaces.

If you make changes to this custom interface, when updates are sent out you will receive a new version. This means that any changes you make to this interface would be lost. We provide a user interface for your customizations. See the Microstation Tips Chapter for information.

Roadway Design has added many new tool tasks. No existing standard Microstation tools have been modified.

The following section documents the tools that have been added in our custom interface:

RWD Main

The RWD Main menu is meant to be a replacement to the standard Main menu that comes with Microstation and has additional tools.

Here is a picture of the current RWD Main Tool.

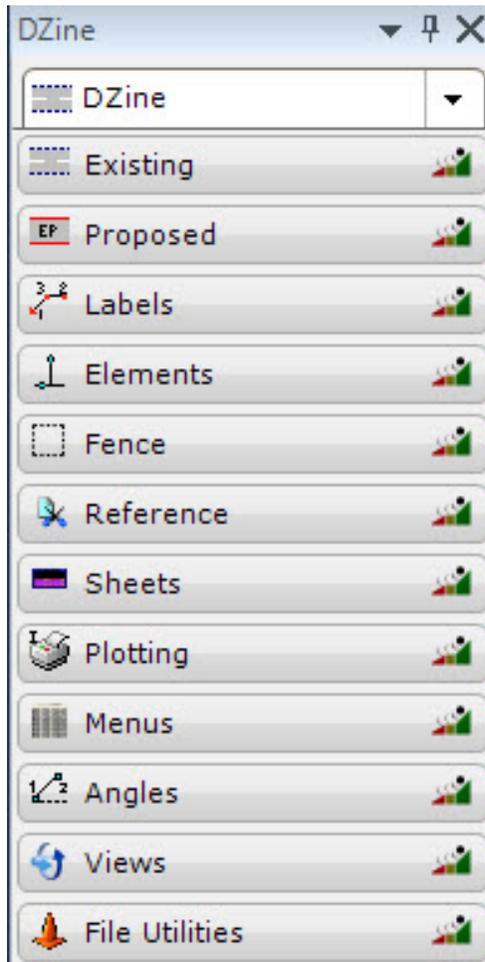
PRESS Function Key F1 to display this menu.



DZine

It can be used to open all the custom tasks created in Roadway Design . These custom tools are located in DZineInterface.dgnlib and can be activated/open from RWDMain

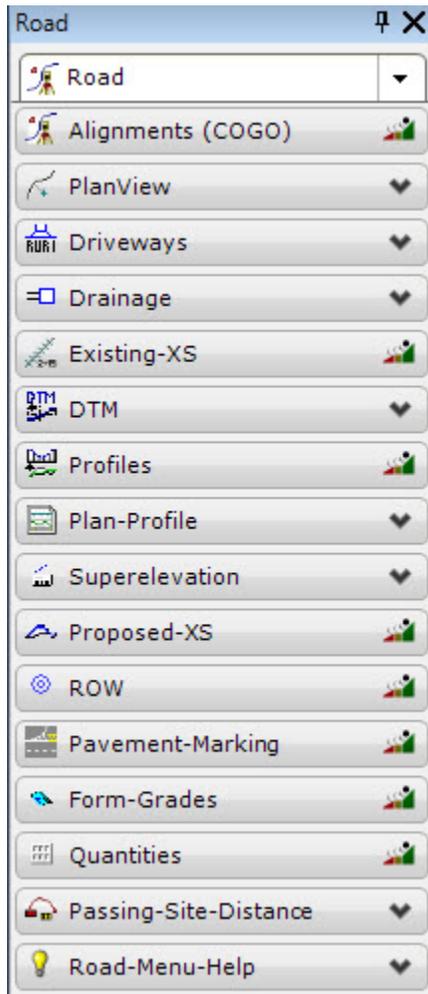
PRESS Function Key F2 to open this dialog.



Geopak-Road

A main task menu has been created for Custom Geopak-Road commands. These custom tools are located in GPKInterface.dgnlib and can be activated/open from RWDMain.

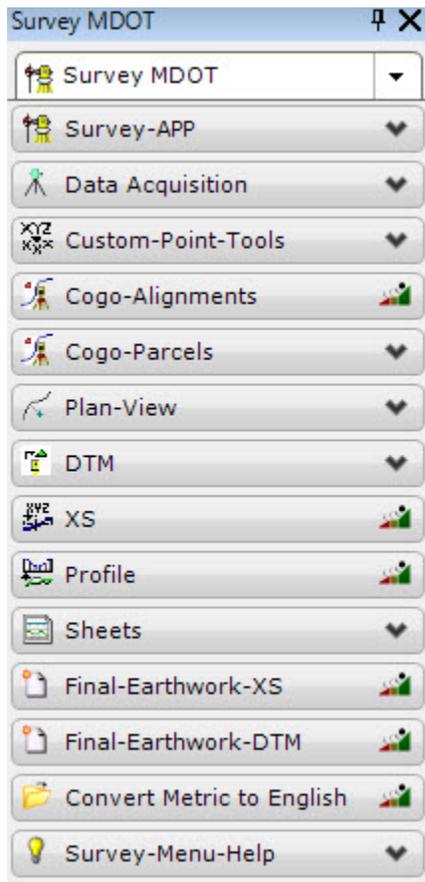
PRESS Function Key F3 to open this dialog.



Geopak-Survey

A main task menu has been created for Custom Geopak-Survey commands. These custom tools are located in GPKInterface.dgnlib and can be activated/open from RWDMain.

PRESS Function Key F4 to open this dialog.



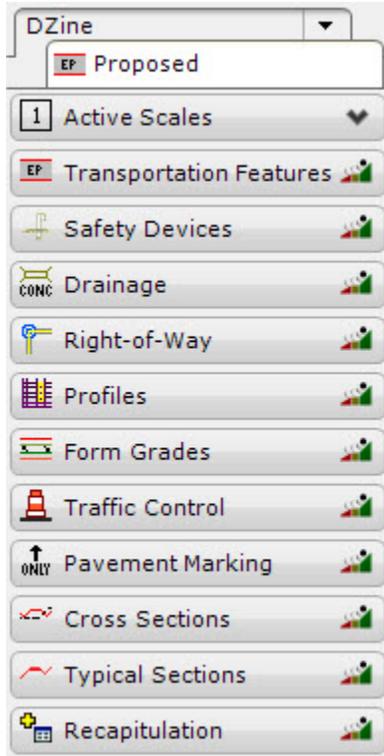
EXISTING

Dzine-Existing



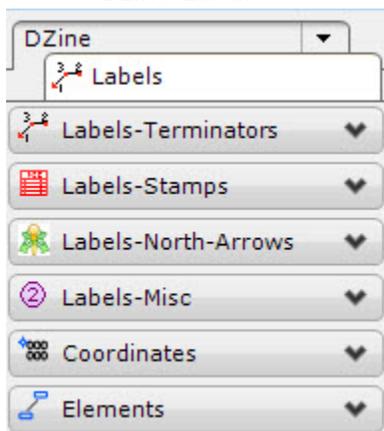
PROPOSED

Dzine-Proposed



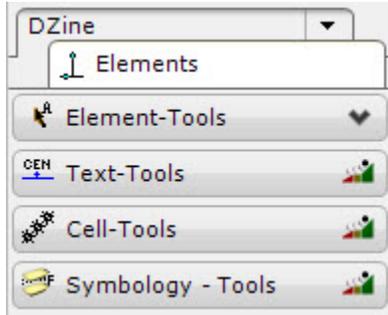
LABEL

Dzine-Labels



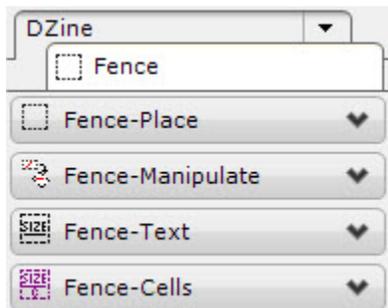
ELEMENTS

Dzine-Elements



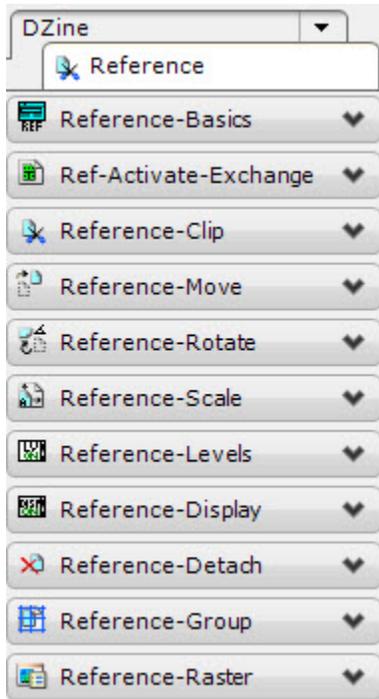
FENCE

Dzine-Fence



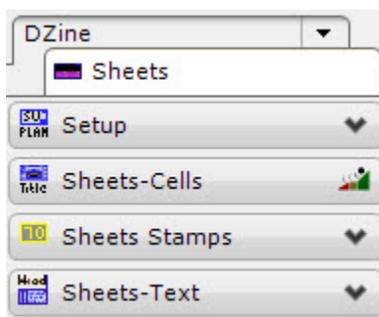
REFERENCE

Dzine-Reference



SHEETS

Dzine-Sheets



PLOTTING

The Plot Task menu gives access to tools to make it a little easier to Plot. Quick access Microstation Print, IPlot Apps, Color Tables, Plot Border, Scales, and View Manipulations.



MENUS

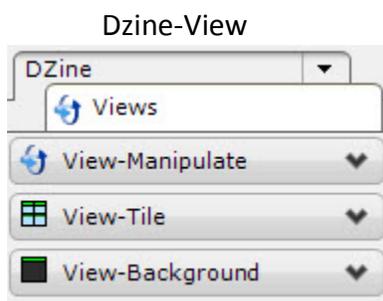


ANGLES

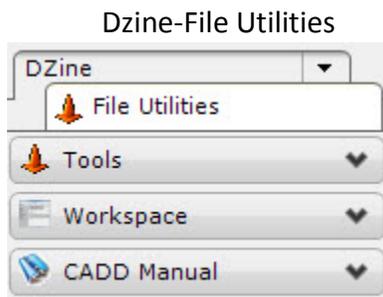


VIEWS

The View Task menu gives access to tools to help you quickly change your view settings



FILE UTILITIES



Paper Menus

What are Paper Menus?

The following information on paper menus contained here is to retain the historical methods used to create and use menus. We no longer have the proper hardware to use paper menus and the menu cells are no longer provided.

Paper menus are tablet-based menus containing blocks. These blocks contain information to perform a specific task. They consist of drawing a graphical representation of the menu and placing action strings in the blocks to tell it what specific task to perform. It is then created as a cell and stored in a cell library. The cell is created by putting an 'm' after the description of the cell or by choosing menu from the option menu in the create new cell dialog box. You must have a hardware menu-table attached to your workstation and the Microstation Tablet drivers installed to use these menus.

Create Paper Menu User Commands

There are user commands that were designed to help create paper menus.

These commands no longer come with Microstation. Roadway Design has modified these commands and they are still located in the Group directories.

See the Cadd Section for information on their use.

Information Not Documented by Microstation

1. Menu cell libraries once delivered with Microstation are no longer available. If you want a paper menu. You must draw it, and create the menu cell.
2. When creating *matrix menus* the command blocks do not have to be square. However, if created as rectangles they will all have to be the same size.
3. When the creating user command places an action string text line, that contains underbars. You will have to edit the text later and put in the real underbars. Microstation thinks that the original underbars are *enter data fields*.
4. If placing an action string that performs a UNIX command you will have to go back and edit the text string to make the characters lower-case.
5. You can have only one command menu and up to 3 matrix menus attached at one time.
6. You will have to delete the existing menu cell from a cell library before adding back a modified menu as a cell in the same library. That is, if you do not change the menu cell name.

7. You do not have to plot the menu representation at a 1 to 1 scale. It can be any size as long as it fits on the digitizing tablet or table.

8. Sometimes after attaching a menu, it may slip on the tablet. To check the alignment of the hand-held cursor to the paper menu command locations. Type in: *mc* and follow the prompts.

Cursor Button Menus

The following information on cursor button menus contained here is to retain the historical methods used to create and use menus. We no longer have the proper hardware to use these menus and the menu cells are no longer provided.

What are Cursor Button Menus?

Cursor button menus can be created for users who have a 12-Button Cursor mouse. Commands can be assigned to correspond to locations on the mouse. The buttons are internally labeled 1 to 8. The 'C, D, R, and T' buttons cannot be reassigned and button's 1 and 2 cannot be reassigned in a 3-D file -- they are reserved for 3D tentative and data points. You must have a menu table attached and setup, and the table must have a 12 button cursor puck to use these menus.

If you need help determining the locations of the button numbering relationship contact Roadway Design's CADD Support. Creation of Cursor Button menus require a cell library and special menu cell that can be created with a user command. Contact Roadway Support for information.

Side Bar Menu

NOTICE : SIDE BAR MENUS ARE NO LONGER AVAILABLE

What are Side Bar Menus?

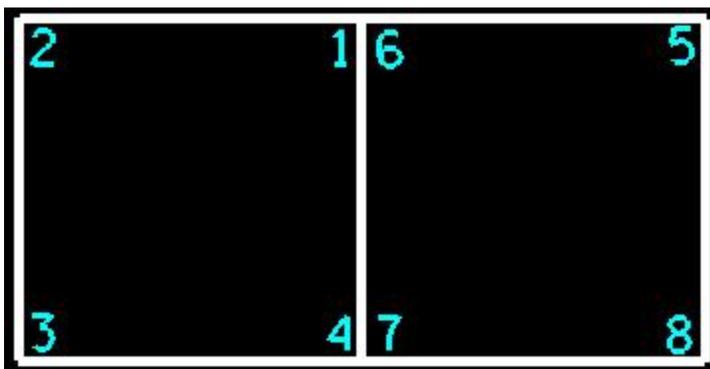
These are older type screen menus that can still be used in Microstation. This section has two parts, one tells you which side bars are available and the other gives instructions on how to modify or create them.

Creating and Modifying Side Bar Menu

A side bar menu is a set of screen menus, usually consisting of a main menu and several subordinate menus. Side bar menus are constantly displayed in a window on the screen. They can be oriented either horizontally or vertically. Since the side bar is displayed in a window, it can be moved around in the screen area.

The advantage of side bar menus over tablet-based or function key menus is with side bar menus, your attention is not diverted from the screen when selecting commands. The disadvantage is the screen space used to display the menu permanently reduces the screen space for the design file. It also can require more selections to get to the desired command over just one selection required on tablet or function key menus.

Side bar menus are attached with the key in: `am=<sidebar_name>, sb#`. The # sign is used to determine the initial position on the screen to attach the menu. The numbers map to the following positions on the screen:



Window 1 Window 2

The diagram above is for users with dual-screen workstation. Use Window 1 only for hardware with only one screen. With later versions of Microstation Window 1 and 2 can be reversed. In

other words, Window 1 can be put on the left and Window 2 can be put on the right. The numbers shown inside of each screen move with the window.

Commands are selected by positioning the screen cursor over the desired command or category and tapping the Data button. They often consist of several sub menus that contain additional selections related to the highest level of selection. When the lower-level selections are available, a default entry can be provided.

Side bar menus are defined in ASCII text files with the '.sbm' extension and may be created and edited using any available text editor. The text file is organized as a sequence of records that defines the root or main menu, which must be the first menu in the file, and all related sub menus. Each menu in the menu file consists of a header record that defines the appearance of the menu on the screen. It is then followed by the menu entry records that define the actions performed by the individual menu selection.

The following general syntax rules apply to all records in the side bar files:

1. Blank spaces and tab characters are ignored except within quotes.
2. Literal strings that define the action to be performed must be enclosed in single quotation marks (').
3. Any characters entered after a semicolon (;) and not enclosed in a literal string are considered a comment and are ignored.
4. Records may be up to 128 characters long. Long records may be divided into multiple records by inserting a backslash (\) as the last character in the record.

For example:

```
'cir 2 in','t,co=1;p,pcirr;2;\
```

```
'/d;t,nulcmd' ... is the same as ...
```

```
'cir 2 in','t,co=1;p,pcirr;2;/d;t,nulcmd'
```

5. All keywords may be entered in either upper- or lower-case.
6. Blank or comment-only records are ignored.

Header Records

Each menu in a side bar file must have a header record that defines the parameters for displaying and processing the menu. The first header record in the file must be the header for the main menu. This header defines the maximum screen area for the display of the menu. All sub menus are contained within the main menu area. The main menu header also defines the orientation of the menus as horizontal or vertical and defines the size of entries.

Submenu headers have a format similar to that of the main header. Some keywords (described below) are specific to either the main menu or the submenu headers.

Header records have the following syntax rules:

1. The first keyword in the record must be either *main_menu* or *sub_menu*
2. Following rule '1' keyword is a series of keywords and values. These keywords may appear in any order and should be separated with commas.

KEYWORD	OPERATION
border	Draws a border around the menu.
color=(, <f>)	Defines the default colors used. =background, <f> foreground
title= <name>	Defines the name of the menu, up to 8 characters long.
vlines	Causes vertical lines to be drawn between the columns of the menu.

Valid **main_menu** keywords:

KEYWORD	OPERATION
by_row	Organize the menu by rows rather than columns. Rows are filled from left to right beginning with the top row.
columns= <n>	Defines how many columns will be used in the menu. Typical values for <n> is 2 for vertical menus and <10> for horizontal menus.
height= <n>	Defines the height of the menu entries. Valid values for <n> are 1 and 2.
rows= <n>	Defines the number of rows used in the menu. Typical values for <n> are 20 for vertical menus and 2 for horizontal menus.
width= <n>	Defines the width of the menu entries in characters. Valid values for <n> are 1 to 20.

Valid **sub_menu** keywords:

KEYWORD

OPERATION

depth=<n> Defines the nesting depth of the menu. When a menu's activated, all existing menus of equal or higher depth are erased, but existing menus of lower depth are not affected. This is useful so menus may call submenus but do not effect existing menus. The main menu is at depth zero and all submenus should be a larger depth.

start=<n>,<m> Defines the starting position for the submenu. If the menu is oriented horizontally, the first number <n> indicates the starting row and the second number <m> indicates the starting column. If the menu is oriented vertically <n> and <m> are reversed.

Examples:

main_menu title=wards, color=(0,7), width=7, height=1, rows=20, column=2, border, vline

sub_menu title=exist, color=(0,12), start=1,0, depth=1, border

Main Entry Records

The menu entries displayed on the screen are defined in the menu entry records that follow the menu header records.

Menu entry records follow the general syntax rules described earlier and follow the general form of:

'<menu line>' , '<action>' [/keyword 1]...[/keyword 2]

The <menu line> is the character string that will display on the screen. The number of string characters is determined by the value assigned to the 'width' keyword in the header record.

The 'action' is the menu action string describing the action to be taken when the menu entry is selected. See the Microstation manuals for more information on menu action strings.

Possible keywords in the menu entry record:

KEYWORD

OPERATION

color=(****,<f>) Defines the default colors used. The first number represents the background and the second the foreground (text) color.

default Indicates the entry that is to be used as the default selection the first time the menu is used. Only one entry can have the default for each menu

line Causes a horizontal line to be drawn under the entry.

nosave Indicates that the entry should not be saved as the default entry after it has been selected. An entry cannot have both the 'nosave' and 'default' keyword.

Examples:

```
'area',t>window area', /line /default /color=(0,2)
```

```
'zoom in',t,zoom in', /line /nosave
```

```
'zoom o',t,zoom out', /line /nosave
```

General Guidelines for Building Side Bar Menus

Side bar menus are simple and very flexible and allow you to create and customize menus for your environment and applications. By following a few general rules, the menus you create will be consistent and easier to use.

- All keep the main part of your menu active at all times. If your menu has sub menus, try to divide the menu screen space into no fewer than two columns.
- Conserve screen space. Although more commands can be more easily accessed with large spacious menus, keep in mind that space is reduced on the screen.
- Make the layout of your menu as logical as possible.
- Select the default menu entry carefully. Default entries can save time by eliminating an input step and can help the novice user by indicating the most common menu selection.
- Use colors wisely. Color can add visual cues to the menu, but too many colors can be distracting.
- A side bar can only have up to 150 sub menus.
- A total of 650 menu entries can be included in the side bar.
- The total number of characters that can be included in a side bar is 10,000.

I recommend that you use the provided side bar menu files and make modifications, instead of trying to create a brand new menu. Save any sidebars you create under your personal folders