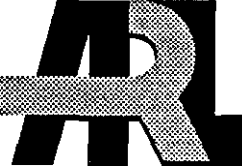


REFERENCE COPY
DOES NOT CIRCULATE

ARMY RESEARCH LABORATORY



XMGED - An X1 1 Interface to MGED

Robert Parker
Phillip Dykstra

ARL-TR-868

September 1995

AUG 1996

REFERENCE COPY
DOES NOT CIRCULATE

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1995	3. REPORT TYPE AND DATES COVERED Final, Jan 93 - Jæ 95	
4. TITLE AND SUBTITLE XMGED - An XI 1 Interface to MGED		5. FUNDING NUMBERS WO: 4B081501H2A094 PR: 612783	
6. AUTHOR(S) Robert Parker and Phillip Dykstra			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRL-SC-CC Aberdeen Proving Ground, MD 21005-5067		8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-868	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Funding was provided by the Survivability/Lethality Analysis Directorate (SLAD).			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release: distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report describes enhancement and changes made to the BRL-CAD MGED solid model editor. While called XMGED, it has been implemented primarily as a new display manager in MGED, and as such all of the old interfaces still operate. Besides offering a more modern user interface, there are also numerous new features including command history and editing, journaling and script tiles, multiple views, alternate mouse modes, position (in addition to rate) based knobs/sliders, user definable menus, etc. The description contained here is for a beta release of the code. The Motif widget set was used, so at present a machine with Motif is needed to compile this software (included with all Sgis). The software is also known to work on Suns and SunOS 4.x with Motif. We encourage user feedback both for bug reports and criticism of current features or suggestions of desired features.			
14. SUBJECT TERMS BRL-CAD, MGED, GUI		15. NUMBER OF PAGES 45	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

INTENTIONALLY **LEFT** BLANK.

TABLE OF CONTENTS

	<u>Page</u>
LIST OF FIGURES	v
LIST OF TABLES	vii
1. Introduction	1
2. About This Release	2
3. Getting Started	2
3.1 Single Display	2
3.2 Separate Drawing Area	2
4. Startup File	5
4.1 General Information	5
4.2 User Menus	5
4.3 Key Binding	6
4.4 Mouse Button Binding	7
4.5 Dial Binding	7
4.6 Button Box Binding	9
5. Multiple Drawing Areas	9
6. Multi-Views	9
7. Command Window	11
8. New Commands	11
8.1 AIP	11
8.2 ALIAS	11
8.3 BINDKEY	13
8.4 BUTTON	13
8.5 CLOSEW	15
8.6 CUE	15
8.7 HISTORY	15
8.8 IKNOB	16
8.9 IROT	16
8.10 ITRAN	16
8.11 JOURNAL	16
8.12 LIGHT	16
8.13 OPENDB	16
8.14 OPENW	17
8.15 PS	17
8.16 SAVEDIT	17
8.17 SLIDER	17
8.18 SOURCE	17

	<u>Page</u>
8.19 SV	18
8.20 TRAN	18
8.21 UNALIAS	18
8.22 ZBUFFER	18
8.23 ZCLIP	18
9. Modified Commands	18
9.1 MATER	18
9.2 PRESS	18
9.3 SET	20
10. New Buttons	23
10.1 SAVE EDIT	23
10.2 ALT MOUSE MODE	23
10.3 MULTI VIEWS	23
10.4 SWAP VIEWS	23
10.5 INPUT WIN	23
10.6 RATE BASED	23
10.7 HELP	23
11. The FILE Menu	24
11.1 MGED DATABASE FILE	24
11.2 POSTSCRIPT FILE	24
11.3 PIX FILE	24
11.4 OPEN	24
11.5 INSERT	24
11.6 SHOW INFO	24
11.7 MANUAL	24
11.8 EXIT	24
12. New Environment Variables	24
12.1 MGED_JOURNAL	24
12.2 MGED_SRC_DIR	24
12.3 XMGEDRC	25
13. Future Enhancements	25
APPENDIX A: STARTUP DEFAULTS	27
APPENDIX B: KEYSYM LIST	33
APPENDIX C: SAMPLE SOURCE FILE	39
APPENDIX D: X11 RESOURCES	43
DISTRIBUTION LIST	49

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.	Utility Windows	1
2.	Combined User Interface	3
3.	Separate User Interface	4
4.	General Menu Specifications	6
5.	Sample Menu	6
6.	General Key Specification	7
7.	General Mouse Button Specification	7
8.	General Dial Specification	8
9.	General Button Box Specification	9
10.	Multiple Views	10
11.	Partially Raised Window Pane	12
12.	Edit Axis	20
13.	View Axis, center	21
14.	View Axis, lower left	22

INTENTIONALLY LEFT BLANK.

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Dial Keywords	8
2. Command Line Editing	11
3. Command History	16
4. Slider Numbers	17
5. New Press Command Options	19
6. Arb Edit Specifiers	19
7. NMG Edit Specifiers	19
8. Original Press Options	20
9. View Axis Values	21
D-1. X Resources	45
D-2. X Classes	46

INTENTIONALLY **LEFT** BLANK.

1 Introduction

The primary motivation to design a new GUI (graphical user interface) for MGED was to take advantage of XT (X Toolkit) and the look and feel of the widget sets that are layered above that. In particular, the X display manager has been rewritten to make use of the Motif¹ widget set. The usual assortment of display managers are still available; however, this document concerns itself primarily with the X display manager. Any deviation from this will be noted.

In addition to modernizing the GUI, many new features have also been added. Among these are command history and editing, command aliasing, journaling and script files, multiple views, multiple drawing windows, online documentation, alternate mouse modes, position (in addition to rate) based sliders/knobs, viewing axes, binding of keys and mouse buttons, support for dials and buttons, user defined menus, color support, etc. Figure 1 depicts XMGED with an assortment of utility windows displayed.

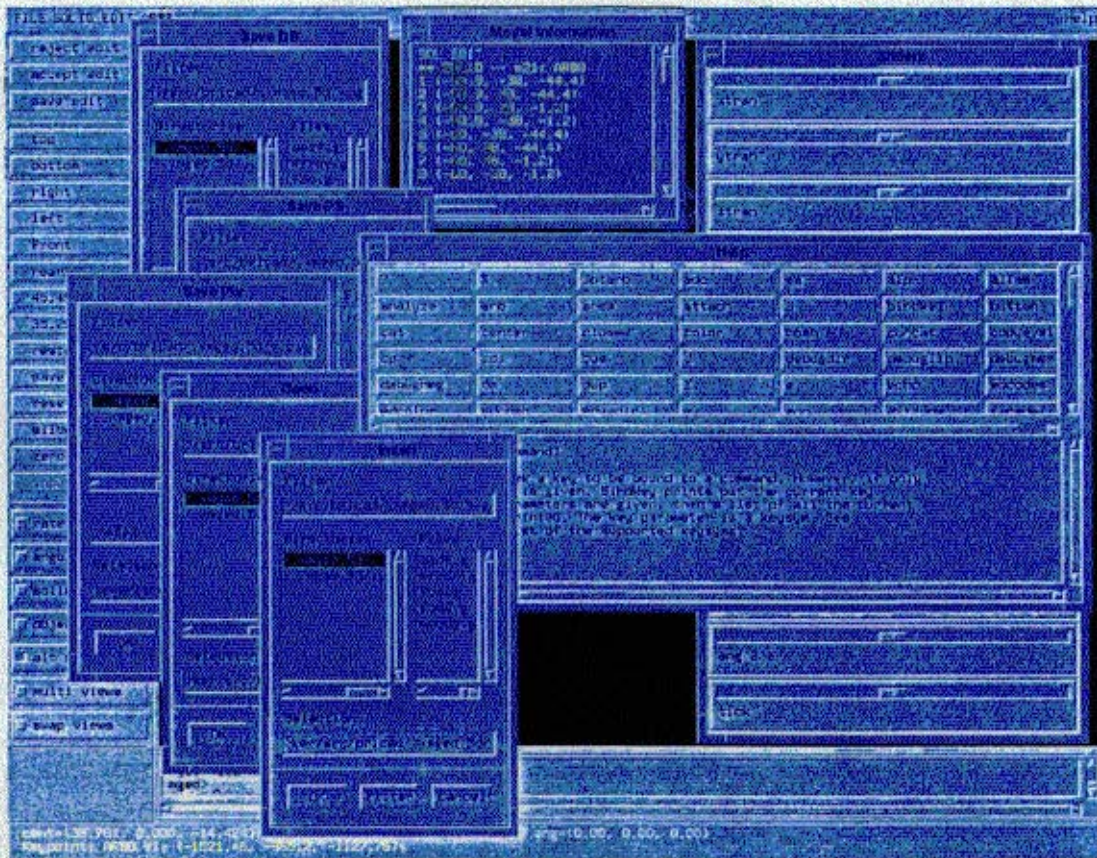


Figure 1: Utility Windows

Many new commands and buttons have been added. The buttons pertain only to the X display manager, while the new commands have been designed to work with all display managers. However, in the case where the command is inherently display manager dependent, hooks have been provided to allow support for a particular display manager. In commands of this type, no support has yet been added for other display managers, so the command simply returns. To add support for a particular display manager, one would need to write a new function within that display manager, declare the appropriate hook, and initialize the hook to point to the newly created function. One would also have to set these hooks to null when releasing the display.

¹Motif is a trademark of the Open Software Foundation, Inc.

The new X display manager is more than just a replacement for the old X display manager. It can also be considered as a replacement for the Sgi display manager because when running on an Sgi, the drawing is done in GL (Sgi's Graphics Library). As with the Sgi display manager, access to the dials and buttons is also provided. It should be noted that the customary Sgi display manager is still available.

For convenience, listed at the end of this document are the startup defaults, a list of the supported keysyms, a sample source file, pertinent X resource names, and the fallback X resources.

2 About This Release

Source code for XMGED is included with BRL-CAD release 4.3 or later. You will need a machine with X11 and the Motif widget set. All Sgi machines come with Motif. On Suns and other machines, this may be an extra cost option. If there is enough interest, a version using a public domain widget set will be developed.

3 Getting Started

3.1 Single Display

Here is an example of starting XMGED in its single window mode

```
% xmgcd file.g
BRL-CAD Release 4.3   Graphics Editor (MGED)
Fri Jan 14 17:13:46 EST 1994, Compilation 12
bparker@admii:/scivis/phil/cad4.3/.xmgcd.5d

attach (nu|tek|tek4109|ps|plot|sgi|X)[nu]? X
X Display(User Interface) [:0.0]?
X Display(Drawing Area 0) [:0.0]?
X Display(Drawing Area 1)
ATTACHING X (X Window System (X11))
```

You should attach to the "X" interface. Even if you are on an Sgi, "X" must be selected to get the new interface. The code will automatically sense that you are using an Sgi and use the fast GL graphics routines for the drawing area.

You are given the option of entering several X DISPLAY strings. The first is where you would like the user interface to be displayed, i.e., the part with the buttons, menus, command window, etc. The second prompt (for Drawing Area 0) allows you to specify a different location for the drawing area. If the specified string is identical to that for the user interface, you will get the combined form, i.e., the drawing area will be embedded into the user interface as seen in Figure 2. If these displays differ, the drawing area will be placed into a separate window.

Finally it prompts you for an optional Drawing Area 1. If a DISPLAY specification is entered here, XMGED will open a second drawing area. It will continue to prompt for additional drawing areas until an empty line (return) is entered. In this way, you can open up any number of drawing areas on any number of X servers.

3.2 Separate Drawing Area

Here is an example of starting XMGED with the user interface on a separate machine from the drawing area. The advantage of this is that you can dedicate an entire display to the drawing area and get a large command window on the user interface machine as seen in Figure 3.

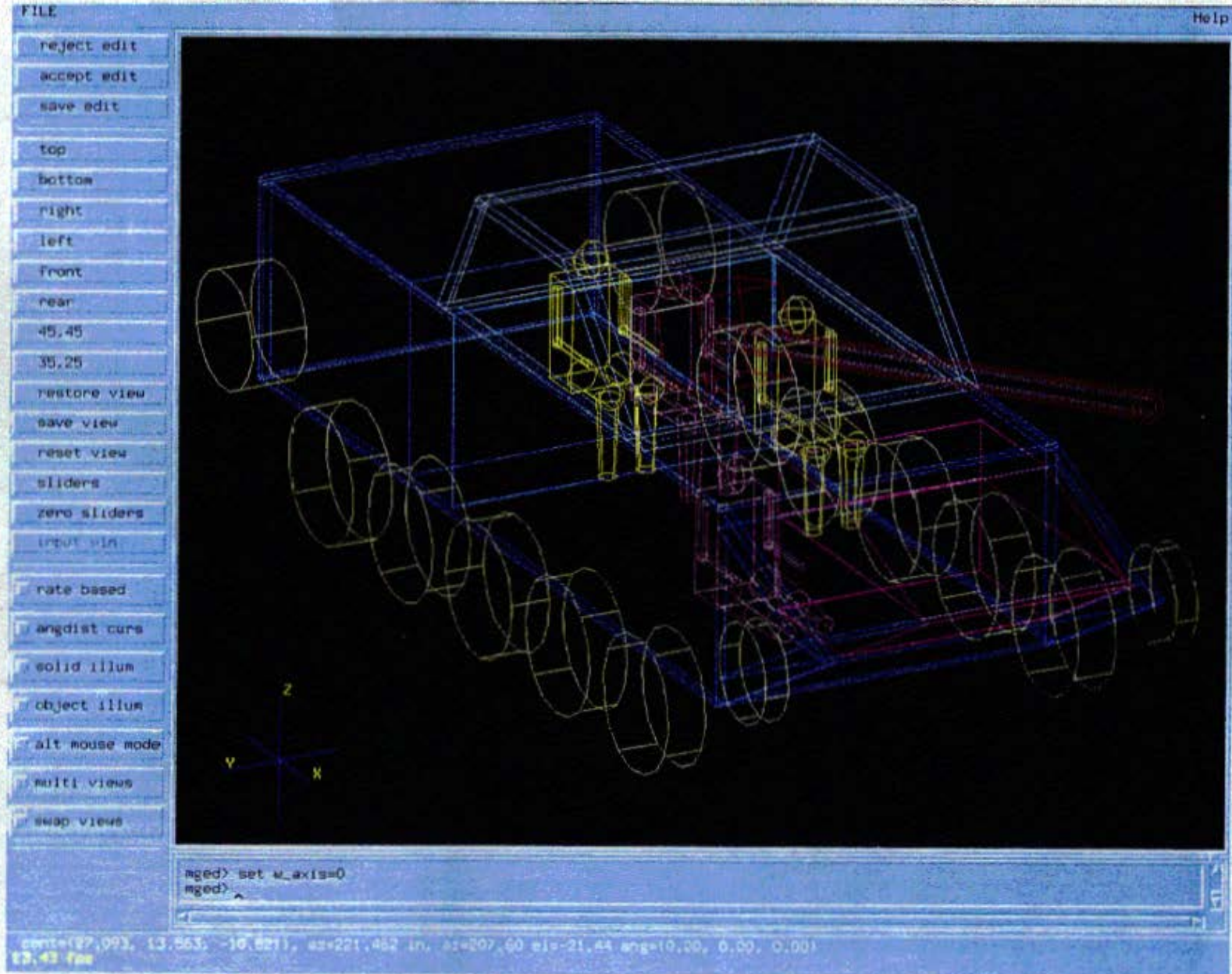


Figure 2: Cobmined User Interface

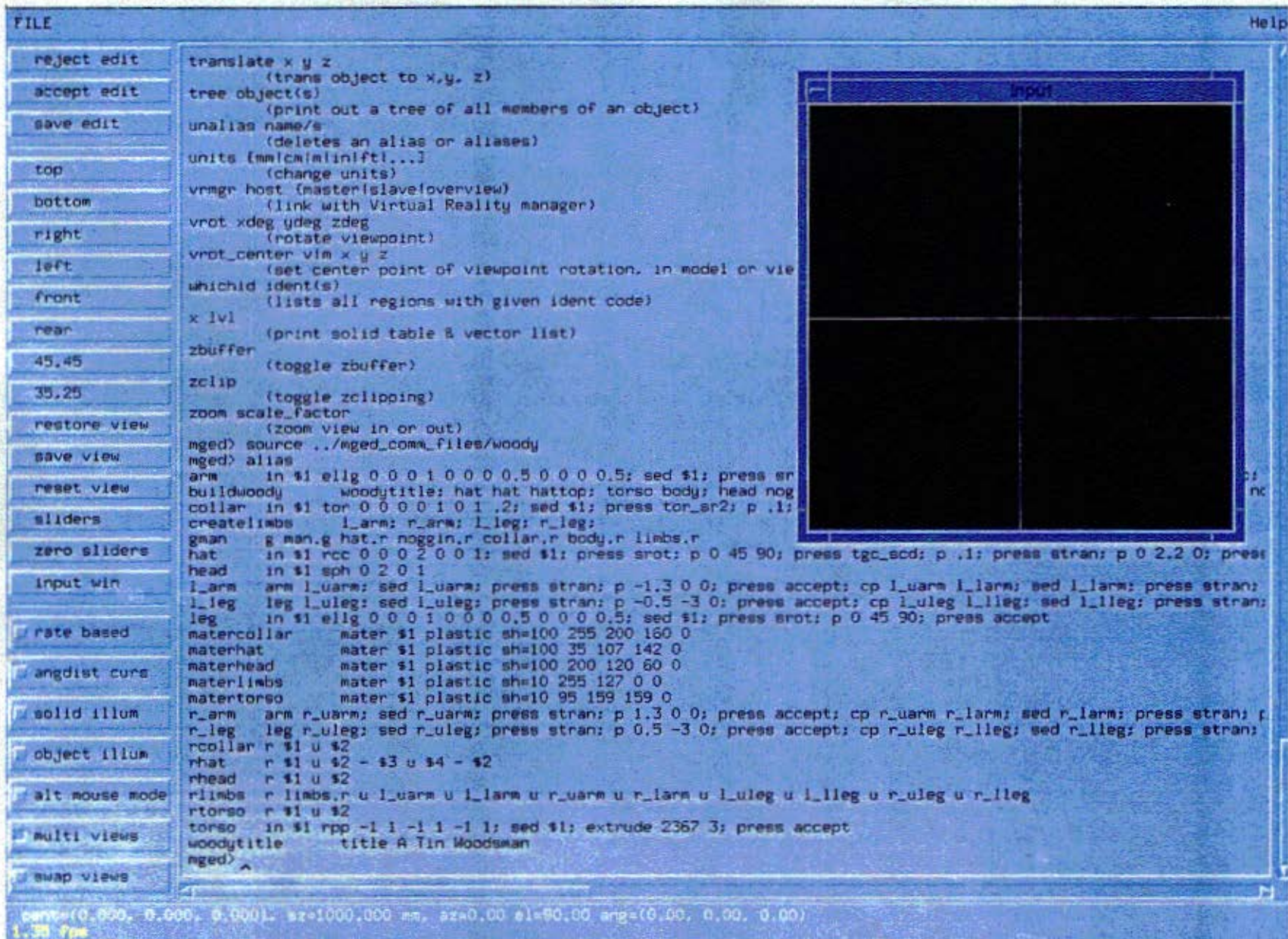


Figure 3: Separate User interface

```

% xmgcd file.g
BRL-CAD Release 4.3 Graphics Editor (MGED)
Fri Jan 14 17:13:46 EST 1994, Compilation 12
bparker@admi:/scivis/phil/cad4.3/.xmgcd.5d

attach (nu|tek|tek4109|ps|plot|sgi|X)[nu]? X
X Display(User Interface) [:0.0]? remotehost:0
X Display(Drawing Area 0) [:0.0]?
X Display(Drawing Area 1)
ATTACHING X (X Window System (X11))

```

When the user interface and drawing areas are to be displayed on different X servers, it is recommended that the drawing area be made local (i.e., on the machine running XMGED) while the user interface is made remote. This is because bandwidth to the drawing area is much more important than to the user interface.

In all cases of separate or multiple drawing areas, mouse input is active in all of them, but mouse input normally required in the drawing area, e.g., to center or resize the view, can be directed to an "input window" on the user interface machine instead. This input window is selected from the main menu.

4 Startup File

4.1 General Information

A startup file exists that is associated with the X display manager. The XMGEDRC environment variable defines this file which is read in during the initialization process. If this is not defined, or the startup file is not found, the current directory is searched for the file *.xmgedrc*. If this file is not found, it is searched for in the user's home directory. See a list of the startup defaults in section A.

The startup file provides for the specification of menus, key bindings, mouse button bindings, dial bindings, and button box bindings. The contents of each specification is enclosed within curly braces. All of the specifications begin with a keyword. In the case of a menu specification, the keyword is followed by a string. A string is defined as a series of one or more characters surrounded by double quotes. The body of the string starts out with an optional exclamation point followed by a letter. The rest of the string can be any character except a double quote or a newline. The following regular expression more succinctly defines the set of allowable strings.

```
!?[a-zA-Z] [^\n]*
```

4.2 User Menus

The menu specification is similar to that found in a *.twmrc* file. A menu can be defined anywhere and in any order within the startup file. Figure 4 shows a generalized menu specification. There are five keywords associated with a menu specification. They are *menu*, *f.title*, *f.separator*, *f.menu*, and *f.exec*. The *menu* keyword is used to begin the definition and is followed by a string that names the menu. The other keywords are functions/actions. *F.title* creates a title in the menu. *F.separator* places a line in the menu. *F.exec* and the '!' character are both followed by a shell command that is executed when selected. And finally, *f.menu* references a submenu and is followed by the name of the submenu.

Submenus can be used in more than one menu. It should be noted that submenus can also have submenus and that there is no software limit within XMGED to how deep they can go. However, if an infinite loop is detected, the corresponding menu button will be made insensitive. Top level menus are menus that are not referenced within another menu. They will be given buttons in the menu bar and will be labeled with the menu name.

```

menu string
{
    string1 action1

    string, action,
}

```

Figure 4: General Menu Specification

The contents of the menu consist of two columns of information. The first column contains the button labels, while the second contains a specification of some action to be taken. The action is either a function keyword possibly followed by a string, or an optional '!' followed by a string. If a string is specified by itself in the right column, it is interpreted by XMGED. Figure 5 shows two sample menus.

```

menu "main"
{
    "The Main Menu"      f.title
    ""                  f.separator
    "another menu"      f.menu "menu1"
    "shell ls"          f.exec "ls"
    "shell ls"          !"ls"
    "mged ls"           "ls"
}

menu "menu1"
{
    "user defined command" "buildtank"
    "exit"                 "q"
}

```

Figure 5: Sample Menu

In Figure 5, main is a top level menu, while *menu1* is a submenu. Main illustrates the different kinds of menu items that can be created and will be given a button labeled main in the menu bar.

4.3 Key Binding

A key can be bound to an arbitrary command. These key bindings take effect only within a drawing window or the input window. That is, if key *z* is bound to the command *zoom 0.5*, then pressing *z* while in the drawing window will execute *zoom 0.5*.

The keyword *Keys* is used to begin the specification of key bindings. Figure 6 shows a generalized key binding specification. There are two columns of information, one for the keysym and one for the command. A keysym is a machine independent symbolic representation of a key. See section B for a list of the supported keysyms. Keys can also be bound using the *bindkey* command. See section 8.3 for a description of *bindkey*. Also see section A for a list of the default key bindings.


```

Keys
{
    keySYM1 string1

    keySYMn stringn
}

```

Figure 6: General Keg Specification

4.4 Mouse Button Binding

The keyword *Buttons* is used to begin the specification of the mouse button bindings. Figure 7 shows a generalized mouse button binding specification. There are three columns of information. The first of which specifies the mouse button. The following keywords are used for referencing a mouse button: *Button1*, *Button2*, and *Button3*. The last two columns specify the mode and function.

```

Buttons
{
    button, mode1 function1

    button, mode, function,
}

```

Figure 7: General Mouse Button Specification

For the purpose of binding mouse buttons, there are four different modes. The mouse buttons can be bound to one of three functions for each mode. Below is a list of the modes along with three possible functions for each.

view	inzoom, outzoom, slew
alt	translate, rotate, zoom
swap	getview, putview, swapview
edit	inzoom, outzoom, pick

Each of these is also a keyword. The *alt* mode can only be active while in the VIEW state or an EDIT state. The *view* and *swap* modes are only active when XMGED is in the VIEW state. Otherwise, the *edit* mode bindings take effect. See section A for a list of the default mouse button bindings.

4.5 Dial Binding

The dials can also be rebound. The *Dials* keyword begins the specification of the dial bindings. Figure 8 shows a generalized dial binding specification. There are two or three columns of information. The first column denotes the dial. The following keywords are used for referencing a dial: *Dial0*, *Dial1*, ..., *Dial7*. The *Shift* keyword, if used, is in the second column and indicates that the shift key is being held down. The last column is reserved for the function specifier. Table 1 shows a list of function specifiers along with a brief description of each. See section A for a list of the default dial bindings.

```

Dials
{
    dial1 [shift] function1
    ⋮      ⋮      ⋮
    dialn [shift] functionn
}

```

Figure 8: General Dial Specification

Table 1: Dial Keywords

Function Specifier	Function Description
NONE	no action
XJOY	x rotation
YJOY	y rotation
ZJOY	z rotation
XSLEW	x translation
YSLEW	y translation
ZSLEW	z translation
ZOOM	zoom view in or out
ADC1	rotate angle 1
ADC2	rotate angle 2
ADCX	x translation of angle distance cursor
ADCY	y translation of angle distance cursor
ADCDIST	move tick marks

4.6 Button Box Binding

A button box button can be bound to an arbitrary command. The *BBox* keyword begins the button box bindings specification. Figure 9 shows a generalized button box binding specification. There are two columns of information. The first refers to the button. The following keywords are used for referencing a button: *Button1*, *Button2*, , *Button3*. The second column indicates the command. See section A for a list of the default button box bindings.

```
BBox
{
    button1 string1

    button, string,
}

```

Figure 9: General Button Box Specification

5 Multiple Drawing Areas

Any number of drawing areas can be opened for drawing. This feature was created in order to display copies of the drawing area on different workstations for the purpose of training, or for collaboration with colleagues. Input events such as buttonpress, keypress, and pointer motion, can be captured as usual in the drawing window(s) or remotely, on the GUI display, using the input window. The input window is active only if the GUI is not combined with a drawing area. The GUI and drawing area 0 will be combined if, during the initial prompting for display names, the display name of drawing area 0 is the same as the display name of the GUI. Figure 2 shows the combined user interface, while Figure 3 shows the separated user interface. If running XMGED on an Sgi and displaying on an Sgi, XMGED will open a GL window for drawing, otherwise it will open an X window. If the drawing area is an X window and the X display supports color, then the solids will be drawn in a color that closely matches the assigned color.

6 Multi-Views

A new mode called *multi-views* allows four different views to be displayed at one time, as can be seen in Figure 10. The default views shown are 45,45, front, left and top. Multi-views mode is toggled on/off by clicking the button labeled *multi views* in the main menu or, assuming the default key bindings, by pressing the *m* key while in the drawing area.

All slider operations and view-related commands entered from the command window or by making a menu selection affect only the lower right window which shares its view with the main window. The views in each of the windows, other than the lower right window, can only be changed by using the mouse, keyboard, or dials and buttons. If use of the command window, menus, or sliders are desired to manipulate the view within either of these three windows, the view can be swapped over to the lower right window. This can be done while in swap mode or by using control-n or control-p to cycle through the views. When in the drawing window, control-n and control-p cause the view to be changed to the next or previous view respectively, and will work whether or not multi-views mode is active.



Figure 10: Multiple Views

7 Command Window

There is a text window associated with the GUI for entering commands. This window has command line editing capabilities which include a facility for accessing previous commands. The functionality is similar to that of `tclsh`. See Table 2 for a list of the features that have been implemented. For more information on command history, see the `history` command in section 8.7.

Table 2: Command Line Editing

Control sequence	Function
<code>^p</code>	previous command
<code>^n</code>	next command
<code>^b</code>	backward one character
<code>^f</code>	forward one character
<code>-a</code>	beginning of line
<code>^e</code>	end of line
<code>^d</code>	delete next character
<code>^h</code>	delete previous character
<code>^w</code>	delete previous word
<code>^k</code>	delete to end of line
<code>^u</code>	delete to beginning of line

Also note that if the GUI is combined with a drawing area, the drawing area and the text area are placed in a pane widget. This provides a way to increase or decrease the size of the text area by grabbing the pane control sash. The control sash is the small, square box located above and to the right of the text window, Figure 11 shows the combined user interface with the pane partially raised.

8 New Commands

8.1 A I P

```
aip [fb]
```

`Aip` advances the illumination pointer either forward or backward while in a pick state. If the current state is `ST_O_PATH`, then the path position is moved either forward or backward. If no parameters are given, the default, advancement direction is forward. This command was created for the purpose of binding to a key.

8.2 ALIAS

```
alias [alias_name [command(s) parameter(s)]]
```

This command, with no parameters, prints out a list of the currently defined aliases and their definitions. With one parameter, it prints out the definition of `alias_name`. Otherwise, it defines `aliasname` to be that specified by the remainder of the command line. Parameters may be introduced by using `$N` anywhere in the definition. For example, to define an alias to make an `arb8` using the `make` command, enter the following:

```
alias m8 make $1 arb8
```

To make an `arb8` with the name `new_arb8`, the following command can now be entered:

```
m8 new_arb8
```

Multiple commands within the alias definition are separated with semicolons. For example, to define another alias to make an `arb8`, solid edit the new `arb` by translating it to a new location and accept the edit, enter the following:

reject edit

accept edit

save edit

top

bottom

right

left

front

rear

45,45

35,25

restore view

save view

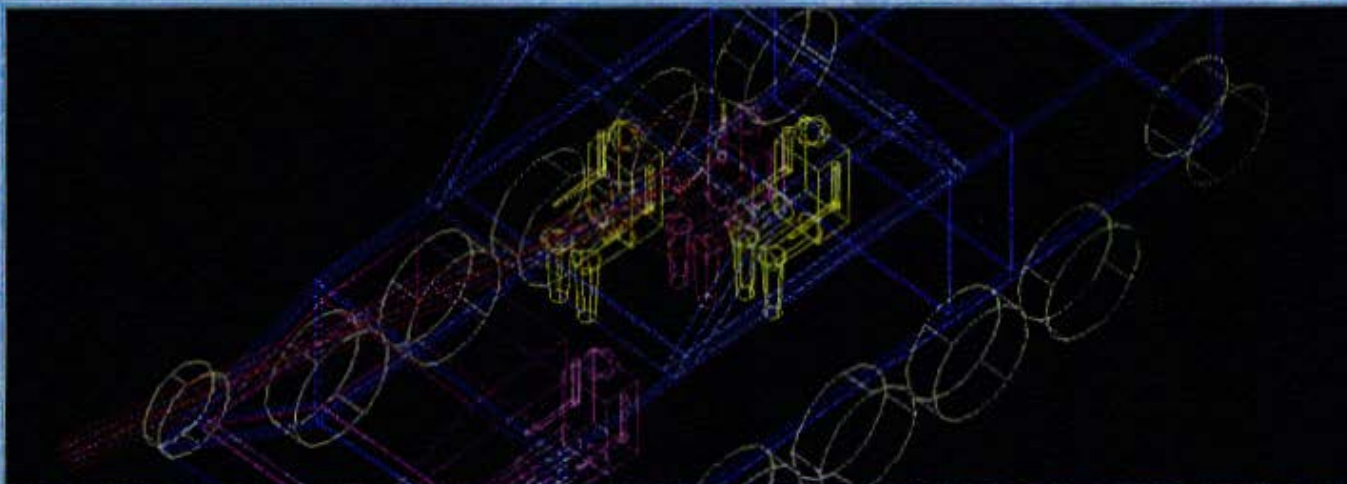
reset view

sliders

zero sliders

input win

rate based

 angled curs solid illum object illum alt mouse mode multi views swap views

r214/R	r32/R	w218	w45
r215/R	r33/R	w219	w5
r216/R	r34/R	w22	w6
r217/R	r35/R	w220	w7
r218/R	r36/R	w221	w8
r219/R	r37/R	w222	w9
r22/R	r38/R	w223	tank/
r220/R	r39/R	w224	tur/
r221/R	r4/R	w225	turret/
r222/R	r40/R	w226	wheels/
r223/R	r41/R	w227	
r224/R	r42/R	w228	
r225/R	r43/R	w229	

```

aged> l wheels
wheels: wheels (len 14) --
  u r32
  u r33
  u r34
  u r35
  u r36
  u r37
  u r38
  u r39
  u r40
  u r41
  u r42
  u r43
  u r44
  u r45
aged> ^

```

cent=(18.885, 7.475, -5.641), ez=305.054 in, az=45.00 el=45.00 angel(0.00, 0.00, 0.00)

18.63 fps

Figure 11: Partially Raised Window Pane

```
alias mse8 m8 $1; sed $1; press stran; p $2 $3 $4; press accept
```

To make an arb8 called moved_arb8 and reposition it to (10, 20, 30), you can now enter the following:

```
mse8 moved_arb8 10 20 30
```

As you can see, the use of aliases within another alias definition is allowed. However, recursive definitions are not allowed. For example, you can define I (alias ll*) to list object information for all objects in the database, but you cannot do the following:

```
alias m7 make $1 arb7
alias marbs m7 $1; m8 $2
alias m8 marbs $1 $2
```

The semiexpanded definition of m8 is:

```
alias m8 m7 $1; m8 $2
```

Now if you enter *m8 a7 a8*, an arb7 named a7 is created, but the command, *m8 a8*, is not recognized because the alias m8 has already been touched, so no substitution is made. That is, the command *m8 a8* is executed as is, which of course fails because it is not a valid MGED command.

8.3 BINDKEY

```
bindkey [key [command]]
```

Bindkey causes a key to be bound to a command. However, if only the *key* parameter is given, *bindkey* prints out the current key binding. If no parameters are given, then a list of all the current key bindings is printed. The *key* parameter is a keysym with the preceding *XX* stripped off. See section B for a list of the supported keysyms.

8.4 BUTTON

```
button number
```

This command simulates a button press and is intended to be used by XMGED when journaling. *Number* refers to a particular button. The following is a list of button numbers:

DESCRIPTION	NUMBER	DESCRIPTION	NUMBER
arb4 move point 1	0	ellipse scale A	97
arb4 move point 2	1	ellipse scale B	98
arb4 move point 3	2	ellipse scale C	99
arb4 move point 4	3	ellipse scale A,B,C	100
arb4 move face 123	4		
arb4 move face 124	5	tgc scaleH	101
arb4 move face 234	6	tgc scale A	102
arb4 move face 134	7	tgc scaleB	103
arb4 rotate face 123	8	tgc scale C	104
arb4 rotate face 124	9	tgc scale D	105
arb4 rotate face 234	10	tgc scale A,B	106
arb4 rotate face 134	11	tgc scaleC,D	107
		tgc scale A,B,C,D	108
arb5 move edge 12	12	tgc rotate Ii	109
arb5 move edge 23	13	tgc rotate AXB	110
arb5 move edge 34	14	tgc move end H(rt)	111

arb5 move edge 14	15	tgcmove end H	112
arb5 move edge 15	16	eto scale R	113
arb5 move edge 25	17	eto scale D	114
arb5 move edge 35	18	eto scale c	115
arb5 move edge 45	19	eto rotate c	116
arb5 move point 5	20	rpc scale B	117
arb5 move face 1234	21	rpc scale H	118
arb5 move face 125	22	rpc scale R	119
arb5 move face 235	23	rch scale B	120
arb5 move face 345	24	rch scale Ii	121
arb5 move face 145	25	rch scale R	122
arb5 rotate face 1234	26	rch scale C	123
arb5 rotate face 125	27	epa scale H	124
arb5 rotate face 235	28	epa scale A	125
arb5 rotate face 345	29	epa scale B	126
arb5 rotate face 145	30	ehy scale H	127
arb6 move edge 12	31	ehy scale A	126
arb6 move edge 23	32	ehy scale B	129
arb6 move edge 34	33	ehy scale C	130
arb6 move edge 14	34	spline pick vertex	131
arb6 move edge 15	35	spline move vertex	132
arb6 move edge 25	36	nmg pick edge	133
arb6 move edge 36	37	nmg move edge	134
arb6 move edge 46	38	nmg split edge	135
arb6 move point 5	39	nmg delete edge	136
arb6 move point 6	40	nmg next eu	137
arb6 move face 1234	41	nmg prev eu	138
arb6 move face 2365	42	nmg radial eu	139
arb6 move face 1564	43	nmg extrude loop	140
arb6 move face 125	44	nmg debug edge	141
arb6 move face 346	45	object scale	142
arb6 rotate face 1234	46	object X move	143
arb6 rotate face 2365	47	object Y move	144
arb6 rotate face 1564	48	object XY move	145
arb6 rotate face 125	49	object rotate	146
arb6 rotate face 346	50	object scale X	147
arb7 move edge 12	51	object scale Y	148
arb7 move edge 23	52	object scale Z	149
arb7 move edge 34	53	solid rotate	150
arb7 move edge 14	54	solid translate	151
arb7 move edge 15	55	solid scale	152
arb7 move edge 26	56	reject edit	153
arb7 move edge 56	57	accept edit	154
arb7 move edge 67	58	save edit	155
arb7 move edge 37	59		
arb7 move edge 57	60		
arb7 move edge 45	61		
arb7 move point 5	62		
arb7 move face 1234	63		
arb7 move face 2376	64		
arb7 rotate face 1234	65		
arb7 rotate face 567	66		

arb7 rotate face 145	67		
arb7 rotate face 2376	68	top	156
arb7 rotate face 1265	69	bottom	157
arb7 rotate face 4375	70	right	158
		left	159
arb8 move edge I2	71	front	160
arb8 move edge 23	72	rear	161
arb8 move edge 34	73	45,45	162
arb8 move edge I4	74	35,25	163
arb8 move edge I5	75	restore view	164
arb8 move edge 26	76	save view	165
arb8 move edge 56	77	reset view	166
arb8 move edge 67	78	sliders	167
arb8 move edge 78	79	zero sliders	168
arb8 move edge 58	80	input win	169
arb8 move edge 37	81		
arb8 move edge 48	82	angdist curs	170
arb8 move face 1234	83	solid illum	171
arb8 move face 5678	84	object illum	172
arb8 move face 1584	85	alt mouse mode	173
arb8 move face 2376	86	multi views	174
arb8 move face 1265	87	swap views	175
arb8 move face 4378	88	rate based	176
arb8 rotate face 1234	69		
arb8 rotate face 5678	90	help	177
arb8 rotate face I584	91	Mged database file	178
arb8 rotate face 2376	92	Postscript file	179
arb8 rotate face 1265	93	Pix file	180
arb8 rotate face 4378	94	Open	181
		Insert	182
torusscale radius I	95	Show Info	183
torus scale radius 2	96	Manual	164
		Exit	185

8.5 CLOSEW

closew [host]

Close the drawing area associated with *host*. *Host* must be in the same format as would the DISPLAY environment variable. For example, *host* could be *someMachine:0*. If the GUI is combined with a drawing area, then that area is not affected by this command. If *host* is not specified, a list of the currently opened drawing areas is printed.

8.6 C U E

cue

Toggle depth cuing on/off. This command was created for the purpose of binding to a key.

8.7 HISTORY

history [N]

This command lists all of the previous commands or the last *N* commands if *N* is specified. MGED currently implements a subset of the csh history list recall facility with one caveat. The “@” character

replaces “!” because MGED already uses “!” to execute a shell command. Table 3 lists the implemented features.

Table 3: Command History

Command	Function
@N	execute the Nth command
@@	execute the last command
@-N	execute the Nth command from last one
@str	execute the last command with “str” anywhere in the command line

8.8 IKNOB

`iknob id [val]`

Iknob is like the *knob* command except that *val* is used to incrementally change the value associated with *id*. This command was created for the purpose of binding to a key.

8.9 IROT

`irotd x y z`

Irot is used to incrementally rotate solids/objects while in an edit state, or to rotate the view while in a nonedit state. The values of *x*, *y*, and *z* represent degrees of rotation. This command was created for the purpose of binding to a key.

8.10 ITRAN

`itrans x y z`

Itran is used to incrementally translate solids/objects while in an edit state, or to slew the view in a nonedit state. The values of *x*, *y*, and *z* represent increments in view coordinates. This command was created for the purpose of binding to a key.

8.11 JOURNAL

`journal [file]`

This command toggles journalling. When journalling is on, commands are written to file which can later be executed via the *source* command. If *file* is not entered on the command line, a file name is chosen for you. For more information, see a description of the MGED_JOURNAL environment variable in section 12.1.

8.12 LIGHT

`light`

Toggle lighting on/off. This command was created for the purpose of binding to a key.

8.13 OPENDB

`opendb database`

Read a new *database* into XMGED. If the given *database name* does not exist, the user will be prompted to create a new database. If the user decides not to create a new database and no database is currently open, the program is exited. Otherwise, the previous database remains loaded.

8.14 OPENW

`openw [host]`

Open a drawing area on *host*. *Host* must be in the same format as the `DISPLAY` environment variable. For example, *host* could be `someMachine:0`. If *host* is not specified, a list of the currently opened drawing areas is printed.

8.15 PS

`ps Cfl file`

Creates a postscript file of the current view. If the *f* option is given, the original faceplate is also drawn.

8.16 SAVEDIT

`savedit`

The *savedit* command allows the user, while in an edit state, to save all edits and remain in the edit state. To continue editing, another edit option must be selected.

8.17 SLIDER

`slider number value`

This command simulates a slider adjustment and is intended to be used by XMGED when journaling. The *number* refers to a particular slider and *value* refers to the slider's new position. The range of *value* is +/-100. Table 4 shows a list of slider numbers.

Table 4: Slider Numbers

Slider	Number
XSLEW/XTRAN	0
YSLEW/YTRAN	1
ZSLEW/ZTRAN	2
ZOOM	3
XROT	4
YROT	5
ZROT	6
XADC	7
YADC	8
ANG1	9
ANG2	10
TICK	11

8.18 SOURCE

`source [beh]command_file`

With an option of *h*, the commands in *command-file* are added to the history list. If there is no option or an option of *e*, the commands are executed. If the option is *b*, the commands are both executed and added to the history list. See a description of the `MGEDSRC-DIR` environment variable in section 12.2. Also see a sample source file in section C.

8.19 S V

`sv x y`

This command moves the view center to $(x,y,0)$.

8.20 TRAN

`tran x y z`

Tran is used to translate solids/objects while in an edit state, or to slew the view in a nonedit state. The values of x,y , and z are in view coordinates.

8.21 UNALIAS

`unalias defined-alias`

This command undefines the *defined-alias*.

8.22 ZBUFFER

`zbuffer`

Toggle `zbuffer` on/off. This command was created for the purpose of binding to a key.

8.23 ZCLIP

`zclip`

Toggle `zclipping` on/off. This command was created for the purpose of binding to a key.

9 Modified Commands

9.1 MATER

The *mater* command has been changed to handle the case where some or all parameters are entered at one time on the command line. The parameters are positional, so if some are missing, the user will be prompted to enter them in as before. If entering `rgb` values via the commandline, they must all be entered or they will not be used. The *del* parameter may also be used on the command line.

9.2 PRESS

The *press* command has been changed to include the new options as seen in Table 5. These options appear somewhat cryptic, so a little explanation is in order.

All options start with the solid type and are followed by an edit description. The edit description consists of the type of edit followed by the part to be edited. For non-arbs and non-nmgs, the first character after the underscore specifies the type of edit and is either an *m*, *s*, or *r* that stands for move, scale, or rotate respectively. For arbs, the first two characters after the underscore designate the type of edit. Table 6 shows a list of the arb edit specifiers and their respective meanings. For nmgs, the first two or three characters after the underscore designate the type of edit. Table 7 lists the nmg edit specifiers and their respective meanings. The rest of the option is used to describe the part of the solid to be edited. As an example, *epa_sb* represents a button in the *epa* solid edit menu that says scale *B*. And *tgc_rh* represents a button in the *tgc* solid edit menu that says rotate *H*. While *a8_me12* represents a button in the *arb8* solid edit menu that, says move *edge 12*. And *a4_rf124* represents a button in the *arb4* solid edit menu that says rotate face *1.24*. For completeness, Table 8 lists the original press options.

Table 5: New Press Command Options

a8_me12	a8_me23	a8_me34	a8_me14	
a8_me15	a8_me26	a8_me56	a8_me67	a8_me78
a8_me58	a8_me37	a8_me48	a7_me12	a7_me23
a7_me34	a7_me14	a7_me15	a7_me26	a7_me56
a7_me67	a7_me37	a7_me57	a7_me45	a7_mp5
a6_me12	a6_me23	a6_me34	a6_me14	a6_me15
a6_me25	a6_me36	a6_me46	a6_mp5	a6_mp6
a5_me12	a5_me23	a5_me34	a5_me14	a5_me15
a5_me25	a5_me35	a5_me45	a5_mp5	a4_mp1
a4_mp2	a4_mp3	a4_mp4	tgc_sh	tgc_sa
tgc_sb	tgc_sc	tgc_sd	tgc_sab	tgc_scd
tgc_sabcd	tgc_rh	tgc_raxb	tgc_mh	tgc_mhh
tor_sr1	tor_sr2	eto_sr	etosd	eto_sc
eto_rc	ell_sa	ell_sb	ell_sc	ell_sabc
ars	spl_pv	spl_mv	a8_mf1234	a8_mf5678
a8_mf1584	a8_mf2376	a8_mf1265	a8_mf4378	a7_mf1234
a7_mf2376	a6_mf1234	a6_mf2365	a6_mf1564	a6_mf125
a6_mf346	a5_mf1234	a5_mf125	a5_mf235	a5_mf345
a5_mf145	a4_mf123	a4_mf124	a4_mf234	a4_mf134
a8_rf1234	a8_rf5678	a8_rf1584	a8_rf2376	a8_rf1265
a8_rf4378	a7_rf1234	a7_rf567	a7_rf145	a7_rf2376
a7_rf1265	a7_rf4375	a6_rf1234	a6_rf2365	a6_rf1564
a6_rf125	a6_rf346	a5_rf1234	a5_rf125	a5_rf235
a5_rf345	a5_rf145	a4_rf123	a4_rf124	a4_rf234
a4_rf134	rpc_sb	rpcsh	rpc_sr	rhcsb
rhc_sh	rhc_sr	rhc_sc	epa_sh	epa_sa
epa_sb	ehysh	ehysa	ehy_sb	ehy_sc
nmg_pe	nmg_me	nmg_se	nmg_del	nmg_neu
nmg_peu	nmg_reu	nmg_el	nmg_de	

Table 6: Arb Edit Specifiers

Edit Prefix	Description
mp	move point
me	move edge
mf	move face
rf	rotate face

Table 7: ii IG Edit Specifiers

Edit Prefix	Description
pe	pick edge
me	move edge
se	split edge
del	delete edge
neu	next edge used
peu	previous edge used
reu	radial edge used
el	extrude loop
de	debug edge

Table 8: Original Press Options

35,25	45,45	accept	adc	bottom
front	left	oill	orot	oscale
ox	oxy	oxscale	oy	oyscale
ozscale	r e a r	reject	reset	restore
right	save	sedit	sill	slice
srot	sscale	sxy	stran	top
help				

9.3 SET

The set command has been changed to include three new MGED variables. They are *w-axis*, *v-axis*, and *e-axis*. These variables correspond to the world axis, view axis, and edit axis, respectively. All of these variables have corresponding X resources for specifying a" axis color. The names of the resources are the same as the variable names.

The world axis and the view axis can be viewed anytime. However, the edit axis can only be viewed while in a" edit state as see" in Figure 12. If the value of any of the variables is 0, the corresponding axis is off. Otherwise it is on. The variable *v-axis* is different in that it has seven possible settings. Table 9 shows a list of these settings. Figure 13 shows the view axis in the center with *v-axis* = 1, and Figure 14 shows the view axis in the lower left with *v-axis* = 2.

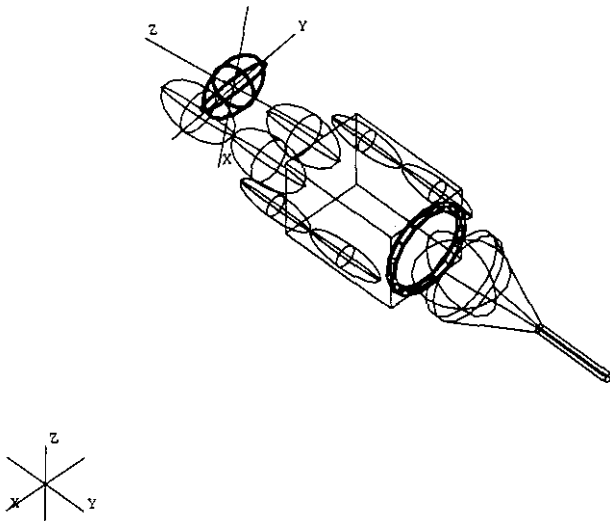


Figure 12: Edit Axis

Table 9: View Axis Values

State	Location
0	off
1	center(full size)
2	lower left(half size)
3	upper left(half size)
4	upper right(half size)
5	lower right(half size)
6	center(half size)

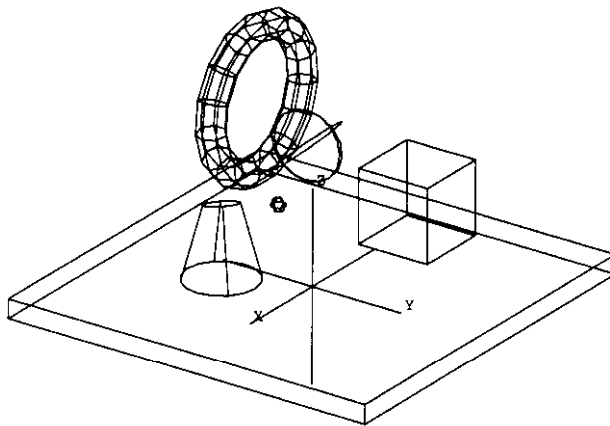


Figure 13: View Axis, center

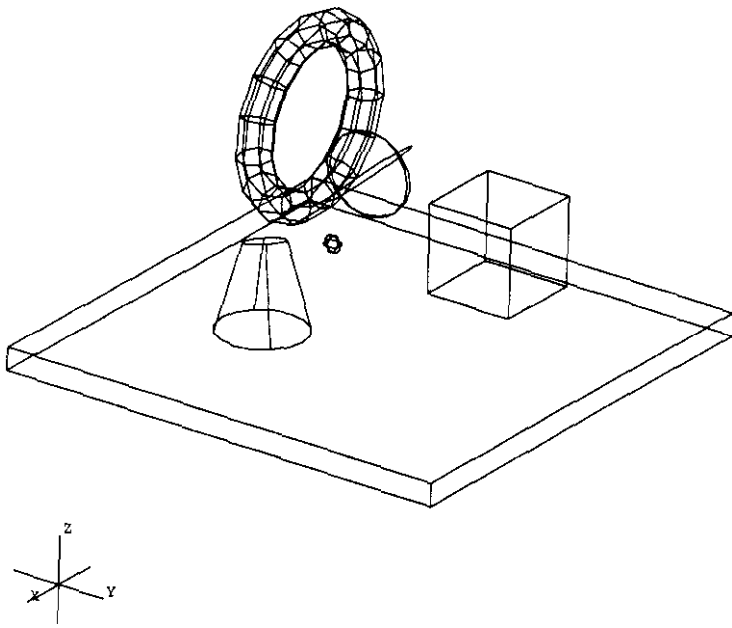


Figure 14: View Axis, lower left

10 New Buttons

10.1 SAVE EDIT

The *save edit* button allows the user, while in either the solid or object edit state, to save all edits and remain in the respective edit state. There is also a new command called *savedit*. Actually, the *saveedit* button is configured to call the *savedit* command when activated.

10.2 ALT MOUSE MODE

The alternative mouse mode can be toggled on/off by pressing the *alt mouse mode* button. This mode offers a virtual track ball style rotation and a grab style translation. Assuming the default mouse button bindings, while in the VIEW state, the left button is used for a translation, the middle button is used for rotation, and the right button is used for zoom/scale. Zoom/scale is effected by moving the cursor vertically while pressing the right mouse button. See section A for a complete list of the default mouse button bindings.

10.3 MULTI VIEWS

Pressing the *multiviews* button toggles between displaying the main drawing window and the multi-views window. The multi-views window shows four different, views at the same time. The default views shown are 45.45, front, left and top. All slider operations and view-related commands affect only the lower right window that shares its view with the main window.

10.4 SWAP VIEWS

The swap views mode can be toggled on/off by pressing the *swap views* button. This mode is useful only when in *multi* views mode. When in this mode, views can be swapped between the lower right window and any of the other three. For instance, assuming the default mouse button bindings, a left button press in any window, except the lower right, puts that window's view into the lower right window. A middle button press swaps that window's view with the view in the lower right window. And a right button press puts the lower right window's view into the window where the button press occurred. However, mouse button presses in the lower right window continue to function unaffected by swap *views* mode.

10.5 INPUT WIN

Display of the input window can be toggled by clicking the *input win* button. However, this button becomes insensitive when the GUI is combined with a drawing area. The input window is used to capture events (buttonpress, motion, keypress, etc.) so that the view and objects can be manipulated remotely.

10.6 RATE BASED

The sliders, as well as the knobs, can be used in either *rate-based* or *positional* mode. In *rate-based* mode, the slider/knob value represents a rate of motion. And in *positional* mode, the slider/knob value represents a nonrate-based translation, rotation, or zoom. This affords the user another mode of control over view change and object movement. Also, when in *positional* mode, the slider labels for *xslew*, *yslew*, and *zslew* change to *xtran*, *ytran*, and *ztran* respectively. In either mode, the mouse is still able to manipulate the view or object(s).

10.7 HELP

Display of the help menu can be toggled on/off by clicking the *help* button. A list of all MGED commands appears in the menu. To see a description of a command, click on its corresponding button.

11 The FILE Menu

11.1 MGED DATABASE FILE

Causes a copy of the current database to be made. The user is prompted for a file name. If no file name is given, no database file is created.

11.2 POSTSCRIPT FILE

Copies the current display into a postscript file. The user is prompted for a file name. If no file name is given, no postscript file is created.

11.3 PIX FILE

Raytraces the current display and puts the result into a file. The user is prompted for a file name. If no file name is given, it puts the result into a frame buffer. The user is also given the opportunity to change the default rt options.

11.4 OPEN

Open another database. The user is prompted for a file name. If no file name is given, the user will be prompted to create a new database. At this time, if the user refuses to create a new database and currently no database is open, the program is exited. See the *opendb* command.

11.5 INSERT

This causes another database to be inserted into the current database. The user is prompted for a file name.

11.6 SHOW INFO

Displays the solid/object information window

11.7 MANUAL

This executes a World Wide Web (WWW) browser (Mosaic by default) with references to a Uniform Resource Locator (URL) that, by default, points to the MGED manual as well as the XMGED documentation. The browser and the URL both have X resource names for specifying user preferences. The resource names are *browser* and *url*, respectively.

11.8 EXIT

Quit XMGED.

12 New Environment Variables

12.1 MGED_JOURNAL

This defines the journal file that is written to when journaling is active. If this is not defined, the file named *mged.journal* is created in the current directory.

12.2 MGED_SRC_DIR

This defines the directory that is searched for command files to be used with the source command. If this is not defined or the command file is not found in this directory, the current directory is searched. If the command file is still not found, the user's home directory is searched.

12.3 XMGEDRC

This defines the startup file that is read in during the initialization process. If this is not defined, or the startup file is not found, the current directory is searched for the file *.xmgedrc*. If this file is not found, it is searched for in the home directory.

13 Future Enhancements

These are a few of the items being considered for the future. It should be noted that all of the XMGED features are subject to change based on user feedback and future design decisions.

- Allow resize of main window
- Provide a choice between using the Motif or Athena widget sets
- Command line editing for MGED proper
- Consider “sing one or more of the following:
 - ◊ C++ and Fresco
 - ◊ Tcl/Tk
 - ◊ Opendgl
- Undo command
- Dump/read state command

INTENTIONALLY LEFT BLANK.

APPENDIX A:
STARTUP DEFAULTS

INTENTIONALLY LEFT BLANK.

A Startup Defaults

The following is a list of the startup defaults presented in the format of an XMGED startup file. See the **Startup File** section for a description of the startup file syntax.

```
Keys
{
    XK_o           "press reset"
    XK_i           "aip f"
    XK_I           "aip b"
    XK_p           "M 1 0 0"
    XK_O           "knob zero"
    XK_x           "iknob x .05"
    XK_y           "iknob y .05"
    XK_z           "iknob z .05"
    XK_u           "iknob X .05"
    XK_v           "iknob Y .05"
    XK_w           "iknob Z .05"
    XK_X           "iknob x -.05"
    XK_Y           "iknob y -.05"
    XK_Z           "iknob z -.05"
    XK_U           "iknob X -.05"
    XK_V           "iknob Y -.05"
    XK_W           "iknob Z -.05"
    XK_f           "press front"
    XK_t           "press top"
    XK_b           "press bottom"
    XK_l           "press left"
    XK_r           "press right"
    XK_R           "press rear"
    XK_3           "press 35,25"
    XK_4           "press 45,45"
    XK_Up          "irot -2 0 0"
    XK_Down        "irot 2 0 0"
    XK_Left        "irot 0 -2 0"
    XK_Right       "irot 0 2 0"
    XK_KP_Up       "irot -2 0 0"
    XK_KP_Down     "irot 2 0 0"
    XK_KP_Left     "irot 0 -2 0"
    XK_KP_Right    "irot 0 2 0"
    XK_KP_End      "irot 0 0 2"
    XK_KP_Next     "irot 0 0 -2"
    XK_KP_Prior    "zoom 1.005"
    XK_KP_Home     "zoom 0.995"
    XK_F1          "cue"
    XK_F2          "zclip"
    XK_F3          "perspective"
    XK_F4          "zbuffer"
    XK_F5          "light"
    XK_F6          "perspective 4"
    XK_5           "iknob distadc 5"
    XK_percent     "iknob distadc -5"
    XK_6           "iknob angl 10"
    XK_asciicircum "iknob angl -10"
    XK_7           "iknob ang2 10"
```

```

XK_ampersand      "iknob ang2 -10"
XK_8              "iknob xadc 10"
XK_asterisk       "iknob xadc -10"
XK_9              "iknob yadc 10"
XK_parenleft      "iknob yadc -10"
XK_KP_1           "itran -0.01 0.0 0.0"
XK_KP_2           "itran 0.01 0.0 0.0"
XK_KP_4           "itran 0.0 -0.01 0.0"
XK_KP_5           "itran 0.0 0.01 0.0"
XK_KP_7           "itran 0.0 0.0 -0.01"
XK_KP_8           "itran 0.0 0.0 0.01"
XK_a              "button 173"      #alternate mouse mode
XK_A              "button 170"      #angle distance cursor
XK_h              "button 177"      #help menu
XK_m              "button 174"      #multi views mode
XK_s              "button 183"      #show info window
XK_S              "button 167"      #sliders
XK_underscore     "set v_axis=0"    #view axis off
XK_F7             "set v_axis=1"    #view axis center
XK_F8             "set v_axis=2"    #view axis lower left
XK_F9             "set v_axis=3"    #view axis upper left
XK_F10            "set v_axis=4"    #view axis upper right
XK_F11            "set v_axis=5"    #view axis lower right
XK_F12            "set v_axis=6"    #view axis center - half size
XK_E              "set e_axis=0"    #edit axis off
XK_e              "set e_axis=1"    #edit axis on
XK_plus           "set w_axis=0"    #world axis off
XK_equal          "set w_axis=1"    #world axis on
}

```

Buttons

```

{
    Button1      View      Outzoom
    Button2      View      Slew
    Button3      View      Inzoom

    Button1      Alt       Translate
    Button2      Alt       Rotate
    Button3      Alt       Zoom

    Button1      Swap      Putview
    Button2      Swap      Swapview
    Button3      Swap      Getview

    Button1      Edit      Outzoom
    Button2      Edit      Pick
    Button3      Edit      Inzoom
}

```

BBox

```

{
    Button1      "button 177"
    Button2      "button 170"
    Button3      "button 166"
}

```



```

    Button4           "button 168"
    Button5           "button 142"
    Button6           "button 147"
    Button7           "button 148"
    Button8            "button 149"
    Button9            "button 184"
    Button10           "button 165"
    Button11          "button 143"
    Button12           "button 144"
    Button13           "button 145"
    Button14           "button 146"
    Button15           "button 174"
    Button16           "button 164"
    Button17           "button 151"
    Button16           "button 150"
    Button19           "button 152"
    Button20           "button 183"
    Button21           "button 172"
    Button22           "button 171"
    Button23           "button 153"
    Button24           "button 157"
    Button25           "button 156"
    Button26          "button 161"
    Button27           "button 162"
    Button28           "button 154"
    Button29           "button 158"
    Button30           "button 160"
    Button31           "button 159"
    Button32           "button 163"
}

Dials
{
    Dial0             xjoy
    Dial1             xslew
    Dial2             yjoy
    Dial3              yslew
    Dial4              zjoy
    Dial5              zslew
    Dial6              none
    Dial7              zoom
    Dial0              Shift      adcx
    Dial1              Shift      xslew
    Dial2              Shift      adcY
    Dial3              Shift      yslew
    Dial4              Shift      adc2
    Dial5              Shift      adcDist
    Dial6              Shift      adc1
    Dial7              Shift      zoom
}

```

INTENTIONALLY LEFT BLANK.

APPENDIX B:
KEYSYM LIST

INTENTIONALLY LEFT BLANK.

B Keysym List

This section contains a list of the supported keysyms which were taken from the `keysymdef.h` header file that is provided with the X11 distribution. The keysyms are used with the `bindkey` command and in the startup file for specifying key bindings. When using the keysym with the `bindkey` command, the preceding `XX_` is not used. The following sections of the header file are contained in the list, below:

- Miscellany
- Japanese keyboard support
- Cursor control & motion
- Misc Functions
- Keypad Functions
- Auxiliary Functions
- Modifiers
- Latin1

Supported Keysyms

<code>XX_BackSpace</code>	<code>XX_R1</code>	<code>XX_j</code>
<code>XX_Tab</code>	<code>XX_F22</code>	<code>XX_k</code>
<code>XX_Linefeed</code>	<code>XX_R2</code>	<code>XX_l</code>
<code>XX_Clear</code>	<code>XX_F23</code>	<code>XX_m</code>
<code>XX_Return</code>	<code>XX_R3</code>	<code>XX_n</code>
<code>XX_Pause</code>	<code>XX_F24</code>	<code>XX_o</code>
<code>XX_Scroll_Lock</code>	<code>XX_R4</code>	<code>XX_p</code>
<code>XX_Sys_Req</code>	<code>XX_F25</code>	<code>XX_q</code>
<code>XX_Escape</code>	<code>XX_R5</code>	<code>XX_r</code>
<code>XX_Delete</code>	<code>XX_F26</code>	<code>XX_s</code>
<code>XX_Multi_key</code>	<code>XX_R6</code>	<code>XX_t</code>
<code>XX_Kanji</code>	<code>XX_F27</code>	<code>XX_u</code>
<code>XX_Muhenkan</code>	<code>XX_R7</code>	<code>XX_v</code>
<code>XX_Henkan_Mode</code>	<code>XX_F28</code>	<code>XX_w</code>
<code>XX_Henkan</code>	<code>XX_R8</code>	<code>XX_x</code>
<code>XX_Romaji</code>	<code>XX_F29</code>	<code>XX_y</code>
<code>XX_Hiragana</code>	<code>XX_R9</code>	<code>XX_z</code>
<code>XX_Katakana</code>	<code>XX_F30</code>	<code>XX_braceleft</code>
<code>XX_Hiragana_Katakana</code>	<code>XX_R10</code>	<code>XX_bar</code>
<code>XX_Zenkaku</code>	<code>XX_F31</code>	<code>XX_braceright</code>
<code>XX_Hankaku</code>	<code>XX_R11</code>	<code>XX_asciitilde</code>
<code>XX_Zenkaku_Hankaku</code>	<code>XX_F32</code>	<code>XX_nobreakspace</code>
<code>XX_Touroku</code>	<code>XX_R12</code>	<code>XX_exclamdown</code>
<code>XX_Massyo</code>	<code>XX_R13</code>	<code>XX_cent</code>
<code>XX_Kana_Lock</code>	<code>XX_F33</code>	<code>XX_sterling</code>
<code>XX_Kana_Shift</code>	<code>XX_F34</code>	<code>XX_currency</code>
<code>XX_Eisu_Shift</code>	<code>XX_R14</code>	<code>XX_yen</code>
<code>XX_Eisu_toggle</code>	<code>XX_F35</code>	<code>XX_brokenbar</code>
<code>XX_Home</code>	<code>XX_R15</code>	<code>XX_section</code>
<code>XX_Left</code>	<code>XX_Shift_L</code>	<code>XX_diaeresis</code>
<code>XX_Up</code>	<code>XX_Shift_R</code>	<code>XX_copyright</code>
<code>XX_Right</code>	<code>XX_Control_L</code>	<code>XX_ordfeminine</code>

XK_Down	XK_Control_R	XK_guillemotleft
XK_Prior	XK_Caps_Lock	XK_notsign
XK_Next	XK_Shift_Lock	XK_hyphen
XK_End	XK_Meta_L	XK_registered
XK_Begin	XK_Meta_R	XK_macron
XK_Select	XK_Alt_L	XK_degree
XK_Print	XK_Alt_R	XK_plusminus
XK_Execute	XK_Super_L	XK_twosuperior
XK_Insert	XK_Super_R	XK_threesuperior
XK_Undo	XK_Hyper_L	XK_acute
XK_Redo	XK_Hyper_R	XK_mu
XK_Menu	XK_space	XK_paragraph
XK_Find	XK_exclam	XK_periodcentered
XK_Cancel	XK_quotedbl	XK_cedilla
XK_Help	XK_numbersign	XK_onesuperior
XK_Break	XK_dollar	XK_masculine
XK_Mode_switch	XK_percent	XK_guillemotright
XK_script_switch	XK_ampersand	XK_onequarter
XK_Num_Lock	XK_apostrophe	XK_onehalf
XK_KP_Space	XK_quoteright	XK_threequarters
XK_KP_Tab	XK_parenleft	XK_questiondown
XK_KP_Enter	XK_parenright	XK_Agrave
XK_KP_F1	XK_asterisk	XK_Aacute
XK_KP_F2	XK_plus	XK_Acircumflex
XK_KP_F3	XK_comma	XK_Atilde
XK_KP_F4	XK_minus	XK_Adiaeresis
XK_KP_Home	XK_period	XK_Aring
XK_KP_Left	XK_slash	XK_AE
XK_KP_Up	XK_0	XK_Ccedilla
XK_KP_Right	XK_1	XK_Egrave
XK_KP_Down	XK_2	XK_Eacute
XK_KP_Prior	XK_3	XK_Ecircumflex
XK_KP_Next	XK_4	XK_Ediaeresis
XK_KP_End	XK_5	XK_Igrave
XK_KP_Begin	XK_6	XK_Iacute
XK_KP_Insert	XK_7	XK_Icircumflex
XK_KP_Delete	XK_8	XK_Idiaeresis
XK_KP_Equal	XK_9	XK_ETH
XK_KP_Multiply	XK_colon	XK_Eth
XK_KP_Add	XK_semicolon	XK_Ntilde
XK_KP_Separator	XK_less	XK_Ograve
XK_KP_Subtract	XK_equal	XK_Oacute
XK_KP_Decimal	XK_greater	XK_Ocircumflex
XK_KP_Divide	XK_question	XK_Otilde
XK_KP_0	XK_at	XK_Odiaeresis
XK_KP_1	XK_A	XK_multiply
XK_KP_2	XK_B	XK_Ooblique
XK_KP_3	XK_C	XK_Ugrave
XK_KP_4	XK_D	XK_Uacute
XK_KP_5	XK_E	XK_Ucircumflex
XK_KP_6	XK_F	XK_Udiaeresis
XK_KP_7	XK_G	XK_Yacute
XK_KP_8	XK_H	XK_THORN
XK_KP_9	XK_I	XK_Thorn

XK_F1	XK_J	XK_ssharp
XK_F2	XK_K	XK_agrave
XK_F3	XK_L	XK_aacute
XK_F4	XK_M	XK_acircumflex
XK_F5	XK_N	XK_atilde
XK_F6	XK_O	XK_adiaeresis
XK_F7	XK_P	XK_aring
XK_F8	XK_Q	XK_ae
XK_F9	XK_R	XK_ccedilla
XK_F10	XK_S	XK_egrave
XK_F11	XK_T	XK_eacute
XK_L1	XK_U	XK_ecircumflex
XK_F12	XK_V	XK_ediaeresis
XK_L2	XK_W	XK_igrave
XK_F13	XK_X	XK_iacute
XK_L3	XK_Y	XK_icircumflex
XK_F14	XK_Z	XK_idiaeresis
XK_L4	XK_bracketleft	XK_eth
XK_F15	XK_backslash	XK_ntilde
XK_L5	XK_bracketright	XK_ograve
XK_F16	XK_asciicircum	XK_oacute
XK_L6	XK_underscore	XK_ocircumflex
XK_F17	XK_grave	XK_otilde
XK_L7	XK_quoteleft	XK_odiaeresis
XK_F18	XK_a	XK_division
XK_L8	XK_b	XK_oslash
XK_F19	XK_c	XK_ugrave
XK_L9	XK_d	XK_uacute
XK_F20	XK_e	XK_ucircumflex
XK_L10	XK_f	XK_udiaeresis
XK_F21	XK_g	XK_yacute
	XK_h	XK_thorn
	XK_i	XK_ydiaeresis

INTENTIONALLY LEFT BLANK.

APPENDIX C:
SAMPLE SOURCE FILE

INTENTIONALLY LEFT BLANK.

C Sample Source File

The following commands set up the aliases necessary to build the tin woodsman simply by typing buildwoody. These kinds of commands can be stored in a file and read into XMGED using the source command.

```
alias woodytitle title A Tin Woodsman
alias hat in $1 rcc 0 0 0 2 0 0 1; sed $1; press srot; p 0 45 90;\
press tgc_scd; p .1; press stran; p 0 2.2 0; press accept;\
    cp $1 $2; sed $2; press tgc_sab; p .1; press stran;\
    p 0 4.2 0; press accept
alias torso in $1 rpp -1 1 -1 1 -1 1; sed $1; extrude 2367 3;\
    press accept
alias head in $1 sph 0 2 0 1
alias collar in $1 tor 0 0 0 0 1 0 1.2; sed $1; press tor_sr2;\
    p .1; press stran; p 0 1 0; press accept
alias arm in $1 ellg 0 0 0 1 0 0 0 0.5 0 0 0 0.5; sed $1;\
    press srot; p 0 45 90; press ell_sb; p .25; press ell_sc;\
    p .25; press accept
alias leg in $1 ellg 0 0 0 1 0 0 0 0.5 0 0 0 0.5; sed $1; press srot;\
    p 0 45 so; press accept
alias l_arm arm l_uarm; sed l_uarm; press stran; p -1.3 0 0;\
    press accept; cp l_uarm l_larm; sed l_larm; press stran;\
    p -1.3 -2 0; press accept
alias r_arm arm r_uarm; sed r_uarm; press stran; p 1.3 0 0;\
    press accept; cp r_uarm r_larm; sed r_larm; press stran;\
    p 1.3 -2 0; press accept
alias l_leg leg l_uleg; sed l_uleg; press stran; p -0.5 -3 0;\
    press accept; cp l_uleg l_lleg; sed l_lleg; press stran;\
    p -0.5 -5 0; press accept
alias r_leg leg r_uleg; sed r_uleg; press stran; p 0.5 -3 0;\
    press accept; cp r_uleg r_lleg; sed r_lleg; press stran;\
    p 0.5 -5 0; press accept
alias rtorso r $1 u $2
alias rlimbs r limbs.r u l_uarm u l_larm u r_uarm u r_larm u l_uleg\
    u l_lleg u r_uleg u r_lleg
alias rhat r$1u $2 - $3 u 54-52
alias rcollar r$1u $2
alias rhead r $1 u 52
alias createlimbs l_arm; r_arm; l_leg; r_leg;
alias materhat mater $1 plastic sh=100 35 107 142 0
alias matertorso mater $1 plastic sh=10 95 159 159 0
alias materlimbs mater $1 plastic sh=10 255 127 0 0
alias matercollar mater $1 plastic sh=100 255 200 160 0
alias materhead mater $1 plastic sh=100 200 120 60 0
alias gman g man.g hat.r noggin.r collar.r body.r limbs.r
alias buildwoody woodytitle; hat hat hattop; torso body;\
    head noggin; createlimbs; collar collar;\
    rhead noggin.r noggin: rtorso body.r body; rlimbs;\
    rcollar collar.r collar; rhat hat.r hat collar hattop;\
    materhat hat.r; matertorso body.r; matercollar collar.r;\
    materlimbs limbs.r; materhead noggin.r; gman
```

INTENTIONALLY LEFT BLANK.

APPENDIX D:
X11 RESOURCES

INTENTIONALLY LEFT BLANK.

D X11 Resources

D.1 Resource Names

You may wish to install the `XMged` resources file, or at least examine it to see about modifying things to taste. You can also set the environment variable `XENVIRONMENT` to the name of the file where your preferred resources are stored.

The resource names for the buttons are identical to the button descriptions as listed for the `button` command in section 8.4 except that the spaces are replaced by underscores. The resource names for the sliders and the help list buttons are the same as their respective labels. The remainder of the pertinent resource names are listed in Table D-1. Table D-2 lists the widget classes used in the implementation

Table D-1: X Resources

Description	Resource Name
command window	text
drawing window	draw
input window	input-win
view1 window	view1
view2 window	view2
view3 window	view3
view4 window	view4
screen info window	screen-info
info window	info
info scroll window	info_scroll
help scroll window	help_scroll
help list form	help_form
help list window	help_list
help text window	help-text
pane window	pane
menu bar	menu-bar
main menu	main_menu
top level form	top_form
open fsb (file selection box)	Open_Fsb
insert fsb	Insert_Fsb
save postscript fsb	Save_Ps_Fsb
save database fsb	Save_Db_Fsb
save pix fsb	Save_Pix_Fsb
rt option text	Rt_Option_Text
rt option label	Rt_Option_Label
WWW browser	browser
uniform resource locator	url
world axis color	w_axis
view axis color	v_axis
edit axis color	axis

Table D-2: X Classes

Widget Classes Used
CascadeButton
PushButton
ToggleButton
Text
PanedWindow
Form
RowColumn
Separator
ScrolledWindow
FileSelectionBox

D.2 Fallback Resources

This is a list of the fallback resources, i.e., the resources that are used if they are not explicitly set elsewhere.

```

XMged*fontList: fixed
XMged*Rt_Option_Text.value: -s512
XMged*draw.background: black
XMged*draw.foreground: white
XMged*pattern: *.g
XMged*Save_Ps_Fsb*pattern: *.ps
XMged*Save_Pix_Fsb*pattern: *.pix
XMged*shadowThickness: 2
XMged*button_container*marginHeight: 4
XMged*button_container*marginWidth: 4
XMged*input_win.background: black
XMged*input_win.width: 400
XMged*input_win.height: 400
XMged*help_list.orientation: HORIZONTAL
XMged*help_list.packing: PACK-COLUMN
XMged*help_list.numColumns: 22
XMged*text.rows: 2
XMged*help_text.rows: 10
XMged*help_text.columns: 80
XMged*keyboardFocusPolicy: POINTER
XMged*scales*frame.shadowThickness: 6
XMged*main_menu*shadowThickness: 2
XMged*separator1*shadowThickness: 4
XMged*separator2*shadowThickness: 4
XMged*separator3*shadowThickness: 4
XMged*separator1*margin: 4
XMged*separator2*margin: 4
XMged*separator3*margin: 4
XMged*separator1*height: 10
XMged*separator2*height: 10
XMged*separator3*height: 10
XMged*help_scroll.width: 800
XMged*help_scroll.height: 200
XMged*info_scroll.width: 300
XMged*info_scroll.height: 200

```



```
XMged*info*background: black
XMged*info*foreground: white
XMged*info.width: 800
XMged*info.height: 800
XMged*screen_info*height: 40
XMged*URL: http://admi1.ar1.army.mil/~bparker/mged/all.html
XMged*browser: Mosaic
XMged*w_axis: yellow
XMged*v_axis: cyan
XMged*e_axis: magenta
```

INTENTIONALLY LEFT BLANK.

**NO. OF
COPIES ORGANIZATION**

2 ADMINISTRATOR
DEFENSE TECHNICAL INFO CTR
ATTN DTIC DDA
CAMERON STATION
ALEXANDRIA VA 223046145

1 DIRECTOR
US ARMY RESEARCH LAB
ATTN AMSRL OP SD TA
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
ATTN AMSRL OP SD TL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

1 DIRECTOR
US ARMY RESEARCH LAB
ATTN AMSRL OP SD TP
2800 POWDER MILL RD
ADELPHI MD 20783.1145

ABERDEEN PROVING GROUND

5 DIR USARL
ATTN AMSRL OP AP L (305)

NO. OF
COPIES ORGANIZATION

ABERDEEN PROVING GROUND

9 DIR USARL
ATTN: AMSRL-SC,
WILLIAM H. MERMAGEN
AMSRL-SC-CC (4 CP)
AMSRL-SL-BV (4 CP)

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions **below** will aid **us** in our efforts.

1. ARL Report Number ARL-TR-868 Date of Report September 1995

2. Date Report Received _____

3. Does **this report** satisfy a need? (Comment on purpose, related project, or other area of interest for which the **report** will **be** used.) _____

4. Specifically, how is the report **being** used? (Information source, design data, procedure, **source** of ideas, etc

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve **future** reports? (Indicate changes to organization, technical content, format, etc.) _____

**CURRENT
ADDRESS**

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

**OLD
ADDRESS**

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

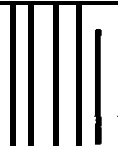
DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 0001, APG, MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
U.S. ARMY RESEARCH LABORATORY
ATTN: AMSRL-SC-CC
ABERDEEN PROVING GROUND, MD 210054067



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

