Table of Contents

Legal Stuff

DMapEdit is my trademark
Doom is a trademark of id Software Inc.
UniVesa - Universal VESA TSR is copyright (C) 1993 Kendall Bennett.
Ms-Windows is a trademark of Microsoft

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. Since I am not a lawyer myself, I do not know exactly what is allowed and what isn't. Therefore, in addition to GNU General Public License (refered to as GPL from this point on in this document), the following restrictions also apply. In the event that I am not allowed to enforce further restrictions on my product under the GPL, this program is not under the terms of the GPL. Should any legal events occur requiring a determination be made on weather or not DMapEdit falls under the terms of the GPL, it can be determined at that time.

So, here are the further restrictions. You may not sell this program for more than $5, if you decide to do so at all. This is only if you sell it as a stand-alone product. If you sell it as part of a package (more than one program), this does not apply, and you may set whatever price you deem appropriate for the package. ('Program', as I am using the term here, refers to individual products, and not individual files.) In either case, this is still my program, and you may not modify it and redistribute it, claim ownership or authorship of it, etc. You may modify and/or create other programs based on DMapEdit, however if you plan to distribute any such derived works, they must follow the restriction listed within the source code. Each source file that is part of the DMapEdit program has a header that explains the restrictions on it.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. Even if DMapEdit doesn't in fact fall under the GPL, there is still no warranty on DMapEdit.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

id Software's "data utility license":

> "DOOM(tm) is a trademark of id Software, Inc. The Doom(tm) characters and the id Software logo are trademarks of id Software, Inc. Doom(tm) Copyright(c) 1993. id Software, Inc. All rights reserved. This utility product is not a product of id Software, Inc. and has not been approved by id Software, Inc. id Software will not provide technical support for this product. The purchaser of this utility product may not distribute for money or other consideration any software data created and/or modified by the use of this utility product."

This Editor is not a product of id Software and id Software cannot and will not provide support for it, nor for maps created with it, nor for DOOM itself if you modify the DOOM.WAD or DOOM2.WAD (IWAD) files. Same is true for the HERETIC.WAD and HEXEN.WAD files.

If any lawyer types happen to notice any "holes" in this section, please feel free to let me know about it. As I said, I am not a lawyer, and so really don't know much about lawyery things.

About DMapEdit

Welcome to DMapEdit. DMapEdit is a true, full-featured map editor. It allows full editing of original and add-on doom maps, and creation of new maps from scratch.

My philosophy in writting DMapEdit is to create a map editor that is BOTH very powerful and easy to use. Many people seem to believe that you can't have both, and that one cancels out the other. I, however, believe that it can be done, and DMapEdit is my proof of this. It is a little harder to accomplish such a programming feat, but far from impossible. It can do almost anything any other editor can, and usually easier. If it can't, it's probably something that I haven't gotten around to adding just yet (DMapEdit has really only one programmer working on it: myself), or it has little value in map making. Basically, you have full control over everything of value, while the computer does everything else for you, allowing maps to be built more quickly, easily, and reliably. If DMapEdit doesn't do something that you want it to, and have an idea on how it should be done, ideas are always welcome (email prefered).

DMapEdit is a map editor, and as such, it isn't designed to editing everything under the sun related to Doom. Things it isn't designed to do, and probably never will be unless the programming staff is expanded, include texture editing, sound effects editing, background music editing, sprite editing, or exe file editing. For doing these related tasks, there are many fine programs available out there, which can probably do a much better job of it, since it can specialize in only that task. DMapEdit is a map editor, and benefits from specializing only in map editing. If programmers out there wish to join this product in an effort add other areas of editing, or just help with the workload, I'd be more than happy to have you on, and work with you. In order to maintain the high quality of DMapEdit, I will act as quality control, however, and review your work before adding it to the source code. If I have problems with any code, you may either modify it yourself, or I can if you wish.

DMapEdit is a shareware product. Though it is no longer crippled in any way, you are required to register this product if you decide to use it. You may evaluate it for a period of 15 days, after which you must decide if you wish to register, or delete it from your system. Registeration is only $10, which should present no major financial problem for anyone. (If you can afford to use a computer, $10 shouldn't be outside your budget!) To register this program, simply send me a $10 check, money order or whatever.

My address is:

Jason Hoffoss
10321 110th st. N.
Stillwater, MN 55082

When I receive your order, I will include your name to my list of registered users. You only ever need to register once, as this will last for the life of this program. I would also request that you let me know what your email address (if you have one) is and current version of DMapEdit you are using when you register. If the version you are using is not the latest version, or if you explicitly request it, I will sent you the latest version at that time. There is no 'registered' version, however. If you ask me to send you a copy of DMapEdit on disk (3.5" only) when you already have the latest version, they will be identical.

When later versions become available, I will distribute them electronically through the Internet and Bulletin Board systems. If you can't get it through these methods, or don't want to wait for it to show up on your BBS's, you can send me more than $10, and I will send you upgrades by postal mail when they become available. Each upgrade I send will cost $2.50, so $15 gets your 2 upgrades, $20 gets you 4, and $25 gets you 6, etc.. If you want me to send it by some special method (Express Overnight or whatever), then make sure you send me enough money to cover such mailing methods, or else it goes by postal mail (about a dollar goes to mailing by this method). I have bubble-lined mailing envelopes I use, by the way, so you don't need to send me an envelope to use or anything.

And hey, send me a letter if you want. Tell me you love/hate this editor, or whatever. I don't have unlimited ideas, and many improvements and idea for DMapEdit come from the users. Tell me how to make it better for you! E-Mail can be send to me on this BBS:

Erebus BBS (612) 439-7808

Or on the internet, my account is **hoffo002@gold.tc.umn.edu** for anyone wishing to leave me email there. I read my internet email roughly daily, while I don't call BBS's really at all anymore. Why bother when you have internet? There is also an official DMapEdit WebPage:

http://www.umn.edu/nlhome/g253/hoffo002/dmapedit/dmapedit.html

The latest version (or beta version) of DMapEdit is guarenteed to be available there. When I can get through, I also upload versions (and beta versions) to ftp.cdrom.com. As the directory structure changes from time to time, and I don't have control over where files I upload are placed, you'll have to look around for them. It should be somewhere off of /pub/doom/ however, most likely.

The naming I use for official releases of DMapEdit is either 'dmapedXX.zip' or 'dmeXXX.zip', depending on the length of the version number (v3.01 was DME301.ZIP for example). Beta versions are named 'dmeXbtY.zip', where X is the major version number, and Y the beta number, or 'dmeXbY.zip', if I can't use the previous method (because it exceeds 8 characters). This naming will be used everywhere, whether on a BBS or on the internet. While Unix machines (such as ftp.cdrom.com) can use longer names, I'll still use this naming, as DMapEdit is a Dos program, and Dos is limited to 8 characters (and anyone tempted to mention Win95 long filenames at this point, don't bother please).

Introduction

So, what all can you do with this editor? Well, you can edit things, Vertexes, Lines and Sectors, etc, in order to create a new and unique map that can then be played in Doom. If you are lost already, you should go and read the "concepts.txt" file before reading any further in this file, and then come back here. You can also generate the internal structures that doom needs to operate (Nodes, Sub Sectors, Segments and a Blockmap), which will all need to be made (or remade) in order to play any new or modified map with Doom. You can edit (or just examine) existing maps (from doom, or from PWADs created by other people). Be sure to follow the pwad's author's copyrights for such maps. (id software doesn't want any modified maps made from their original doom levels to be distributed. In fact, they don't want their unmodified maps to be distributed either, except for with the shareware doom package.)

Map making is rather a complex task. There is a lot that goes into making a map, and being able to just "slap a few lines down, make the nodes and run" really isn't possible. All any editor really does is make the task of creating maps easier, by doing a lot of the work for you. This can cut down on what the user is required to know, but there will still be much that he/she will still need to know. Furthermore, the more you know about making maps, the better your maps will tend to be. Therefore, I strongly recommend reading the file "concepts.txt", included with this package. It will give you an understanding of all the basic concepts of map building. Even if you think you already know it all, I still recommend you read it. You might just find something that you didn't know yet. My goal in writting DMapEdit is to make map generation as simple as possible, while still having the ability to do everything you could possible want to do. Having the ability to do exactly what you want is the main goal. If you think you can't do something, you probably can, but just don't know exactly how to go about it. In such a case, you can read through 'dmapedit.faq' to see if it's covered there. If not, you can read through this file and see if you can find any info about it. If that still doesn't help, you can contact me and I will help you. Please check this file first, though. If it's not in here, and you need to contact me, I will add it to this file (correcting the oversite). At present, I feel that this editor is easier to use than any other out there, while still being nearly equally powerful, and becoming more powerful with every release. If you disagree, and have ideas on how I can make it better (or can just point out what some other editor does better), please let me know.

If you ever run into problems or have questions that this document can't help you with, you should check out the DMapEdit FAQ (frequently asked questions). It's filename is 'dmapedit.faq' (surprise). If that doesn't help either, you'll have to contact me directly.

Features of DMapEdit

When it comes to selecting the best map editor for Doom from the many editors available out there, it's useful to know what features an editor supports without having to dig through the program to find out. That's what this section is all about.

- Doom, Doom II and Heretic support (Hexen is comming soon), all in one editor. You can use any of these formats for maps, and you don't need to quit and restart to switch formats or anything of that nature.

- Full editing ability for things, vertexes, lines, line sides and sectors in six editing modes: thing edit mode, vertex edit mode, line edit mode, sector edit mode, side edit mode, and rectangle mode.

- Very fast and ultra reliable node generator. With all the node generators out there claiming to be the "fastest on the planet" these days, it's hard to say which truely is. DMapEdit's node generator has beaten all such generators that I have compared it to. DMapEdit's node generator was also written from scratch by myself, with some help later from Robert Forsman. As such, it really is a completely different generator from most of them out there, which are all based originally on DEU's node generator. Thus, errors you may have encountered with these various other generators probably will not cause any problems for DMapEdit's.

- Extensive error detection and correction, both while you are working, and selectable whenever you wish to fully check out your map.

- Menus and toolbar available for easy access to DMapEdit's many functions.

- Easy to use and intuitive controls. Manipulating objects with DMapEdit is extremely simple and powerful, allowing you to quickly and easily get the job done. DMapEdit tries to do as much of the busy work for you, instead of making you do it all yourself as many other editors force you to do.

- Marking of objects, so that objects can be manipulated as a group, simplifying and speeding up your work.

- Copying and pasting of marked objects, to and from the same map, or between different maps. You can even edit the clipboard as a normal map as well as load and save the clipboard to a wad file.

- Multi-map, meaning several maps can be loaded at once into memory. Just as you can have multiple documents open in Word, for example, you can have multiple maps loaded in DMapEdit, being able to switch between them at will. Maps can be linked together as well, meaning several windows looking into the same map. Changing the map in one window will be reflected in all linked windows of the map.

- Polygon generator allows you to create regular polygon objects of any size or number of sides. DMapEdit allows you to see what it will look like and adjust it before you place it as well, allowing you to get it just right.

- Automatic staircase generator allows you to quickly and easily build staircases in your maps.

- Doom spawning, which means you can launch Doom to play the map you are working on and return to the editor when you quit Doom.

- Fully configurable through an initialization file called “dmapedit.ini”. This is just a plain ASCII text file, which you can read and change with any text editor, such as the MS-Dos ‘edit’ command.

- Things displayed as un-obtrusive icons, allowing you to see how things are layed out at a glance.

- Grid support, allowing for easy placement and alignment of objects, or as a measuring tool.

- Map is scrollable using the mouse. It is not auto-scrolling, however! I, personally, hate it when you accidently scroll the map off the screen when all you wanted to do was get the mouse out of the way, or access a menu or something. The map does not scroll unless you actually press a mouse button at the edge of the screen, giving you control over scrolling, instead of not giving you such control. Using the left mouse button only scrolls it one position over, allowing you finer control over scrolling, while the right mouse button scrolls continuously until you release it.

- Thing types and linetypes are selectable from a ‘picklist’. It displays a list of all known types, and you select the one you want from that list. You can also manually select the type by number if you wish, or are using an unlisted type.

- Texture picklists as well, with previews of the textures. Simply selecting a texture by name isn’t all that useful most of the time, but being able to see the actual textures, you can pick the one that is exactly right for your needs. Texture previewing is also extremely fast, and draws as a background operation, so you don’t have to wait on it to finish.

  - 3D preview mode, allowing you to view either the entire map, or simply what you have marked, in 3 dimentional space. Various options allow you to view walls as transparent or solid, texture average colored or gray, invisible backsides or not, and some others. Fast as well, assuming you have a math co-processor. If not, it is simply too slow to be useful.

A full history of changes made to DMapEdit in the various versions can be found in the file “history.txt”.

Interested in beta testing the next version of DMapEdit? There’s only one requirement if you are not on the internet: You must be a registered user. So, there’s a good reason to register. If you are already registered and are interested in a beta, just let me know. Of course, you’ll have to handle shipping charges if you want me to mail it to you. Modem transfer is a much better method, if you have a modem. Internet users can always get the latest version or beta version from my WWW homepage:

http://www.umn.edu/nlhome/g253/hoffo002/dmapedit/dmapedit.html

Want the source code? Maybe you want to tweak some things for yourself, or help out with the DMapEdit project, or maybe just learn from what I did. The only requirement is that you must be a registered user. If you are, just request a copy from me, or if you are on the net, it’s available from the DMapEdit homepage.

Getting started

Setup

Ok, running this program is pretty easy. All the files that DMapEdit comes with should all be in the same place. You can run it without it being in the current directory, if you wish. If Doom.wad, Doom2.wad or Heretic.wad isn’t in the current directory, you will need to tell DMapEdit where it can find these files. This is done by giving the path to it with the *-i* (for Doom), *-I2* (for Doom 2) and *-I3 (for Heretic)* switches, like so:

```
dmapedit -i/doom/doom.wad
```

```
      or

dmapedit -i/doom/doom.wad -i2/doom2/doom2.wad

      or

c:\wherever/it\is\dmapedit -ic:\and/doom\is/here/
```

or by adding (or more likely just changing) a line to the INI file like so:

```
doom path = c:/doom
doom 2 path = c:/doom2
heretic path = c:/heretic
```

Note that either '/' or '\' can be used. You should beware of using a '\' as the last character of the path (before a space) in the command line. The compiler sees this as an override, making the space and the next switch part of the current switch. Also note that the path that you give should either point to the directory the IWAD can be found (an IWAD is what the file 'doom.wad' is) or point to the IWAD file itself.

There are many other options you can change within the 'dmapedit.ini' file, which the file itself explains in detail. I will assume for the remainder of this section that you have not changed anything beyond the path lines mentioned above. If you have made further changes, your results may vary from those mentioned here. Just so you know.

Dmapedit defaults to VESA SVGA 640x480x256 colors. If you don't have a VESA card but you do have SVGA, try running univesa to create a VESA interface for your card. If that doesn't work for you, I would recommend trying to get a VESA driver specific to your card from your card's manufacturer, or whatever dealer you bought the card from. They should be able to provide you with this. If you don't even have SVGA, why not? I mean really, a 386+ without a SVGA card? If you want to use a different graphics interface, or resolution, refer to the *Alternate Video Modes* section near the end of this file.

Once the program is up and running, you should see the main editing screen, which includes a white menu bar across the top of the screen, a tan toolbar just below the menu bar, and a gray infobar along the bottom of the screen (further information on each of these can be found in the appropriate sections). An introduction pop-up window should also be displayed at this point. One you have finished reading it, you can press a key or click a mouse button.

Now, you should be face to face with a blank map. You may now either load an original IWAD map, load a map from a PWAD, or simply begin on a new map from scratch.

Demo feature

If you are not familiar with DMapEdit yet, you may wish to watch a demo before you start playing around with DMapEdit. Demos are readl-time demonstrations within DMapEdit that show you around the editor, or how how to go about doing various things.

To play a demo in DMapEdit, start DMapEdit with the *-play* (or *-demo*) switch:

```
dmapedit -play              Play the demo demo.rec
dmapedit -play controls     Play the demo controls.rec
```

All demos use the *'.rec'* filename extension, so this is how you can identify which files are demos (i.e. `dir *.rec`). This extension isn't required for the *-play* switch, however using it won't cause any problems either.

Playback can be halted at any time by pressing the **ESC** key. During the course of a playback, informational windows (info windows) will appear to tell you things. The demo is suspended at this point, and will remain so until you instruct it to continue, by either pressing a key or clicking a mouse button.

Editing basics

There are six editing modes in DMapEdit: thing edit mode, vertex edit mode, line edit mode, side edit mode, sector edit mode, and rectangle mode. Each is described in individual sections. Most of these modes operate in a similar fashion, and have things in common.

First are the 4 mouse functions: add/move, delete, edit, and mark. These functions work closely with the mouse, and thus these functions can be accessed with the mouse button. Each function also has a key equivilent: insert key = add/move, delete key = delete, spacebar = edit, and **M** key = mark.

Since there are only 3 mouse buttons (2 on many mice), and 4 functions, having each function correspond to a mouse button isn't practical. Thus, the mouse buttons are configurable. Each mouse button can correspond to any one of the 4 mouse functions, and these assignments can be changed at any time. The only restriction is that 2 mouse buttons cannot correspond to the same mouse function.

The current mouse button to function assignments are displayed on the toolbar. There are 4 buttons corresponding to the 4 mouse functions. Below each is the mouse button that performs that function. Note that on some mice, the right mouse button is seen by DMapEdit as the middle mouse button. This is a problem with your mouse driver, and not in DMapEdit. You will need a new driver, or just need to remember that the right mouse button is actually the middle mouse button.

To change the assignment of a mouse button, all you need to do is click on the function button using the mouse button you wish to assign to that function. For example, lets say you wanted to make the middle mouse button perform the add/move operation. You could move the mouse pointer to the add button on the toolbar, and click on it using the middle mouse button. This tells DMapEdit you wish to use the middle mouse button for adding/moving.

Mouse functions have both instant actions, if you click and release the mouse button right away, and drag actions, where you click and hold the mouse button down, move (or drag) the mouse to a new location, and release the button. The key equivilents of the mouse functions normally perform the instant actions. In order to perform the drag actions from the keyboard, you need to hold down the Alt key when pressing the key equivilant. At that point, you will enter the drag mode for that function and remain in it until you press a key or click a mouse button.

Add/Move

The instant action for this mouse function is usually to add (or create) a new object, while the drag action is to move a selected object. Which of these 2 actions are available depends on the context you use it in, however. Refer to the appropriate edit mode section for further information on how this mouse function behaves within that mode.

When you have marked objects, the move operation is always used, and operates on the group of marked objects.

Delete

This performs the instant action of removing the selected object, or group of marked objects. There is no drag action defined for this mouse function. This mouse function is also often used to cancel other mouse functions currently in the drag action, should you change your mind in the middle of it.

Edit

This performs the instant action of bringing up an edit window. The layout of the actual window will depend on the context you use the edit function in.

Mark

and a pop-up window. The program defaults to thing edit mode at startup. At this point, you can move things around by pressing the left mouse button on a thing and dragging it to it's new location. Pressing the left mouse button while not on a thing will add a new thing (using the defaults) and placing it when you releast the button. The right mouse button is used to delete a thing. The middle button (if you have one) is used to change the information about a thing, or the default thing if you are not on a thing at the time you press the button. This is the standard mouse interface I use for this program, and editing vertexes and lines works the same way.

Setting up a video driver

The graphics library used in DMapEdit, unfortunately, can only detect a very limited about of video cards. If yours happens to be one of the unlucky few, you can try explicitly telling the library what video driver to use. For this, you need to know what video chip your video card uses. If you don't know, you can look in your video card manual, open your machine up and look at the card, or if you really need to, try each one out to find one that works.

```
SET GO32=DRIVER C:\DOOM2\DETH\DRIVERS\VESADRV.VDR <or your driver>
```

How do I play a map I just made?

Once you have made a map, and saved it to a PWAD file, you need to create 3 structures required by Doom before it will run in Doom. These are the nodes, blockmap, and reject bitmap. Instead of generating each one individually, you can generate all three at once with the 'compile, save and quit' item from the file menu. It will make all these structures, save it, and then quit you back to Dos. Now, you just start Doom up like so:

```
doom -file pwad1.wad [pwad2.wad] [pwad3.wad] [...]
```

In this example, you would have saved your PWAD with the name *pwad1*. If you used the name *new*, you would type:

```
doom -file new.wad
```

Another thing to note about the first example, you can use more than one PWAD file at once. This would be done for files that have different information in them. If they both use the same information (like both having E1M1 in them), you probably shouldn't do it this way, and really don't have any reason to. You would use this for something like *pwad1* having E1M1, *pwad2* having E1M2, and *pwad3* having new sound effects. (by the way, files shown nonbolded and within [ ] are optional, in case you didn't know this. You wouldn't actually type the '[' or ']' either.)

Keyboard commands

Below are all the keyboard commands available in DMapEdit. Each table describes a different context, or location, where these keys are available.

Global commands (available in all edit modes)

?	Display key commands
Arrow keys, Home, End, Page up, Page down	Scrolls the map around on the screen
=, +	Zoom in
- (minus)	Zoom out
_ (underscore)	Center the map position of the mouse cursor on the screen
Alt-_ (underscore)	Re-center the map on the screen
Alt-**Q,** Alt-**X,** Escape	Quit and exit editor
Insert	Add (same as left mouse)[1]
Alt-Insert	Sticky add[2]
Delete	Delete (same as right mouse button)[1]
Spacebar	Change/edit (same as middle mouse button)[1]
Enter	Copy selected objectdef to default objectdef
M	Mark/unmark selected object
Alt-**M**	Sticky mark (stretch-box marking)[2]
A	Mark all objects on map in current editing mode
U	Unmark all objects currently marked in all edit modes
F1	Select thing edit mode
F2	Select vertex edit mode
F3	Select line edit mode
F4	Select sector edit mode
F5	Select rectangle creation mode
F6	Select side edit mode

F12	Compile the map, save it, and quit DMapEdit[3]
N	Start a new map from scratch (erasing the currently displayed map)
S	Save map to a PWAD, promting for name first
L	Load map from a PWAD, promting for name first[4]
Alt-**S**	Save map to a PWAD
Alt-**L**	Load map from a PWAD[4]
Alt-**R**	Reload map from original Doom WAD file (IWAD)
Ctrl-**P**	Play/test the current map in Doom (or whichever game it goes with)
C	Change map position (episode and mission numbers)
T	Change displayed things icon size: squares, big, small, off[5]
G	Toggle grid display on and off
Alt-**G**	Change grid display intensity level
>	Increase grid size
<	Decrease grid size
‘.’ (period)	Increase roundoff level[6]
‘,’ (comma)	Decrease roundoff level[6]
’ (apostrophe)	Toggle roundoff usage on and off
E	Run all error checks on the map to detect any problems
D	Enter “convert sector to door” mode
B	Enter “staircase generator” mode
#	Enter 3D preview mode (see 3D preview mode section)
1 - **5**	Change displayed vertex point size
0	Turn off vertex point display
F	Change thing display filter (see Thing display filter section)
Alt-**1** through Alt-**9**	Bring the selected map slot to the foreground to work on (see Multiple maps section)
X	Toggle the crosshair display on and off
*	Toggle bluebox display on and off

Ctrl-**A**	Display the About DMapEdit window
Alt-**N**	Generate a new Nodes structure for the map[3]
Alt-**B**	Generate a new Blockmap structure for the map[3]
Alt-**J**	Generate a new Reject bitmap structure for the map[3]
'}'	Increase right side indicator knob length
'{'	Decrease right side indicator knob length
Alt-F1	Select the normal line coloring scheme
Alt-F2	Select sector height-change line coloring scheme
Alt-F3	Select Doom automap line coloring scheme
Z	Show current statistics (object totals, and memory usage)
\	Change displayed lines size (thickness)

Notes

1. These are the default mouse buttons. The mouse buttons can be re-configured with the <tab> key.

2. These keys simulate the effect of holding down the equivelant mouse button while moving the mouse around (called dragging). It will remain "stuck" until another key is pressed, or a mouse button.

3. Compiling a map means to generate the nodes, blockmap and reject structures, which are required by Doom to use any map. DMapEdit, however, does not require these structures for normal operation. Because of this, and because compiling a map tends to take a while to do, it is usually a final step you do in an editing session, and then only if you plan to play or test the map. If the map isn't finished and you intend to come back later to work on it, there's no reason to compile it yet.

4. It will try to load a map from the current PWAD. If that map doesn't exist in that PWAD, however, it will load the map from the IWAD.

5. Thing icons can't be turned off while in thing edit mode. There is also a setting that shows circles. This represents the size of the object and is useful to see if monsters overlap, are stuck in walls, etc.

6. This is used to round off all map coordinates, so things can be placed with a more symetrical feel. This is also called "snap-to grid" in some other editors. A value of 1 will not produce any roundoff. The crosshair shows the rounded-off position. The grid display intersections also show the roundoff positions, unless the spacing drops below 8 between the lines, in which case every other line/4th line/8th line/etc will be displayed to meet this condition. This is to not clutter up the screen.

Line Edit Mode

F	Flip line
Alt-F	Flip line and sidedefs[8]

Notes

7. Flipping a line swaps the direction the line points (if you look at if from a vector viewpoint). This causes what was the left side to be the right side, and vice verse. If the sidedefs aren't swapped, the the right sidedef is still on the right side, but looking at it in the editor, it now appears to be on the other side of the line. Swapping the sidedefs along with the line makes the sidedefs appear to not change sides (even though they did).

8. There are some limitations on what side a sidedef may be on for lines. If the line only has one side, the sidedef must be on the right side of the line. If there are 2 sides, but only one of the sidedefs has any wall textures, then that sidedef must be on the right side as well. If swapping the sidedefs isn't allowed because of one of these conditions, the sidedefs won't be swapped.

Inside of a Window

Enter	Accept and close window (Same as clicking the 'Ok' button, if present)
Alt-Q, Alt-X, Escape	Exit out of the window (same as clicking the 'Cancel' button)
Tab	Move mouse pointer to next button position
Page-up, Page-down	In windows with 2-way scroll buttons, scrolls the list

Pop-up windows

Pop-up window manipulation has become a little complex, so I figured I should put in a section to cover it. Pop-up windows are rectangles with a yellow boarder that appear with information inside it. There are two basic types. The first doesn't have an 'Ok' or 'Cancel' button. This is an informational window, and just gives you information. Pressing any key or clicking the mouse anywhere on the screen will make it go away. The second type is the interactive window (also called an edit window), and will have a 'Cancel' button, and usually an 'Ok' button as well. To close this type of window, you must click on either the 'Ok' button or the 'Cancel' button. If an 'Ok' button isn't there, making a selection from inside the window can act like an 'Ok' operation.

The 'Ok' button will exit the window, using the window's current information. Selecting 'Cancel', however, will ignore any changes made in the window. Another function these buttons can be used for is for confirmation of some operation you selected. In this case, you must select 'Ok' to proceed.

A similar sort of button you may see is the 'big button'. This button also has text inside it, but is white, and is always inside the window. Selecting such a button will perform the action the text describes.

Usually, edit windows will have small, circular buttons, which is the main form of doing any 'editing' in the window. Just click the button next to what you are interested in changing. These 'radio' buttons do one of two things: switch states ('on' or 'off' condition), or select item. Buttons that select an item will never be filled in ('on' state).

Inactive radio buttons will be displayed as dimmed (dark gray instead of white). This doesn't always mean you can't select them, however. Try clicking on it to see. When editing a group of marked items, only the non-dimmed items will be changed when you select 'Ok'. So, this being the case, you need to be able to change what's dimmed and what isn't. To do this, you can select the button normally, and it will activate automatically, or you can click on the button with the middle mouse button (or **Spacebar**) to toggle the active status of the button. Only buttons that can logically be toggled will be allowed to be toggled.

Thing display filter

Things are not always on every difficulty level. Generally, harder difficulty levels tend to have more monsters running around. The thing display filter is to allow you to display only what you want to display, such as only level 1 things. Using this filter is a little tricky, however.

There are 5 different flags used by all things in doom:

flag 1:	Skill level 1 & 2
flag 2:	Skill level 3
flag 3:	Skill level 4 & 5
flag 4:	Multiplayer mode (cooperative or deathmatch)
flag 5:	Deaf monster (see explanation below)

Skill level 1 being the easiest ("don't hurt me") and 5 being the hardest ("Nightmare"). If a skill level flag is set, then this thing will appear to a player playing on that skill level. If the multiplayer mode flag is set, then this thing will only appear when playing a multiplayer mode game. Note that this flag is in addition to the skill level flag. If you just set the multiplayer mode flag and leave all three skill level flags cleared, then the thing will never appear in any deathmatch mode game, since you must be in one of the difficulty settings if you are playing.

The deaf monster flag is only useful with monsters. If it is set, then monster will only "activate" (come alive) upon spoting a player or being damaged. If the monster isn't deaf, then it will also "activate" whenever it hears a players weapon being used (even punching air will do it).

There are 2 switches for each of these flags in the display filter. There are a total of 4 possible combinations for these 2 switches (per flag). The first switch is used as a matching state. The second switch is set if you want to force the first switch to match a thing's flag exactly. This may be confusing, so I will detail the 4 possible switch combinations..

Combination 1: both switches off (O O)

With this combination, the flag is ignored and doesn't contribute to a thing being displayed. If all 4 flags have this combination, no things will be displayed, since nothing is telling the program to display anything.

Combination 2: first switch on, second off (* O)

This will display a thing if that thing's flag is set. If the flag isn't set, the thing may still be displayed if another condition directs the thing to be displayed. This is a basic OR situation between flags. If this flag is set, OR another condition directs the thing to be displayed, it will be displayed.

Combination 3: first switch off, second on (O *)

For a thing to be displayed with this combination, the thing's flag must be off. If the flag is on, the thing will not be displayed, regardless of the other flag settings. This is used to hide what you don't want displayed. Basically, this is your NOT operation.

Combination 4: both switches on (* *)

With this combination, similar to combination 3, a thing's flag must be on to display that thing. Unlike combination 2, a thing will not be displayed if the flag is off. Thus, this acts as a "must be on state" for this flag. It also acts as a basic AND operation.

Common filter settings

O = off, * = on

Display level 1 & 2 things:

Skill level 1 & 2:	*	O
Skill level 3:	O	O
Skill level 4:	O	O
Multiplayer mode:	O	O

Display only level 4 things, non-deathmatch mode:

Skill level 1 & 2:	O	O
Skill level 3:	O	O
Skill level 4:	*	O
Multiplayer mode:	O	*

Display all things:

Skill level 1 & 2:	*	O
Skill level 3:	*	O
Skill level 4:	*	O
Multiplayer mode:	*	O

Display only multiplayer mode things:

Skill level 1 & 2:	O	O
Skill level 3:	O	O
Skill level 4:	O	O
Multiplayer mode:	*	O

Display only things unique to level 4:

Skill level 1 & 2:	O	*
Skill level 3:	O	*
Skill level 4:	*	O
Multiplayer mode:	O	O

Thing editing mode

Editing things is very easy. If you want to move a thing to somewhere else on the map, just click on it with the add/move mouse button (the default for this is the left mouse button), and drag it to it's new location while holding the button down and release the button at the destination. You can also do this by selecting the thing to move (place mouse cursor on top of the thing, so it is hilighted) and pressing Alt-Insert. Then, move the mouse to the destination and press the add/move, mark or edit mouse button or press the **Spacebar** or **Enter** or some other harmless key (don't use **Delete**, as this will cancel the move operation!) To delete something from the map, click on it with the delete mouse button (the default for this is the right mouse button), or press **Delete** while it's selected. To add a new thing to the map, click the left mouse button while the mouse isn't on a thing (no thing is selected). It will be placed when you release the button, so you can drag it around too. You can tell if you are on a thing by a "hilight box" around the thing (it pulsates green). If you see this box around it, then that is the thing currently selected. Information about this currently selected thing is displayed in the "bluebox" in the corner of the screen.

To change/edit the currently selected thing, press the edit mouse button (the default for this is the middle button) or the **Spacebar**. This will bring up all information about the thing. If you didn't select a thing, the default thing will be used instead, which is used as the model for all new things you add. To change any of this information, simply select the proper button for it.

Pressing the **Enter** key while in thing edit mode will scan the selected thing's information into the default thing. This is useful if you want to add duplicates of some existing object. Simply scan that thing into the default thing, and any more things you add will use have all this thing's characteristics.

Line editing

I have tried to make the line editing mode as useful and intuative as possible. I have also tried to put as many abilities in this one mode as I could. As a result, you shouldn't need to use vertex edit mode very much, if at all. To use this mode effectively, however, you need to know how to use it properly. So, here goes..

When you move the mouse cursor around the screen, it will lock onto the closest line, and the closest vertex, if it's not too far away. If you have a line selected, you can use the **Spacebar** (or edit mouse button) to edit that line's characteristics (linedef, or line definition), and it's sidedefs. Also, when a line is selected, information about it is displayed at the "bluebox" at the corner of the screen. Sidedef info is also displayed. Each line has 2 sidedefs (or possible sidedefs, anyway), a left sidedef and a right sidedef. Whichever side of the line the mouse cursor is on is also indicated, by a small dot in the top corner of one of the 2 sidedef areas in the bluebox. This dot will also mark the sidedef when in the line edit window as well.

Note: When a linedef that triggers an event is hilighted, the effected sector(s) are also hilighted, using dotted lines.

Adding a Line

A line is added by pressing the add/move mouse button at the starting point, and releasing the button at the ending point. Pressing the button when a vertex is selected (hilighted) will start the line at this vertex. If a vertex is not selected, a new vertex is added, and becomes the starting point for the line. Releasing the button when a second vertex is selected will make that vertex the ending point. Otherwise, a vertex is added and is used for the ending point.

One last possible situation is having a line selected, but not a vertex. When you press the left mouse button here, a vertex is added, spliting the line into two parts, and you can drag this vertex around until you release the mouse button.

Deleting a Line

Simply select the line you wish to delete, and press the delete mouse button. Doing this will leave a gap where the line was. If you use the vertex edit mode to delete a vertex, it will reroute all the line that went to that vertex to the closest vertex along one of that vertexes lines (deleting this line in the process, since with this vertex rerouting, the line would start and end on the same vertex, so it has no length anyway).

Please note that lines may switch sides from time to time. This is because Doom is very picky about having the 'important sidedef' on the right side. This means that if you only have one sidedef, it must be on the right side, so DMapEdit will flip the line around if needed to make it so. You don't have to worry about these details, just edit your map however you want to, and DMapEdit will make these changes as needed.

Sector editing

Once you have all your lines in place, all forming closed polygon shapes, you need to make sectors. Any location (polygon) a player or monster is allowed to be in needs to be a sector. A sector definition tells doom what heights the floor and ceiling is at for the polygon, as well as how much light there is there. If you are making a new map, and have put in all the lines, but have done nothing with the sectors yet, select "generate sectors" from the advanced options menu. This will examine your map and automatically assign a sector to every polygon that needs one. (Be sure your lines are set up correctly, however. Use "fix lines" or "error check lines" first) Then, you can simply edit the various sectors to your liking.

Sector edit mode is selected with the F4 key. In this mode, the mouse will hilight whatever sector the mouse is in (if the sectors aren't damaged). Information about the sector is displayed in the bluebox at the corner of the screen. The **Spacebar** or edit mouse button will let you edit this sector information.

Note: When you hilight a sector that is triggered with a linedef(s), the linedef(s) will be hilighted too, using a dotted line.

If you want to make a new sector where there currently isn't one (no lines hilighted), or want to fix a damaged sector (only some lines of closed polygon hilight), simply press the add/move mouse button (or insert key) to create a new sector there. It will remove whatever sector(s) were there, if any. The sector will have the characteristics of whatever sector most of the lines seemed to be part of. If it was totally empty, the new sector will take on the default sector definition.

If you want to remove a sector (making a pillar solid, for example, so a player can't be inside it) simply press the delete mouse button (or **delete** key) while the sector is hilighted. If a sector is damaged for some reason, you should remake it first (add/move mouse button) and then remove it.

In order for a player or monster to fit inside the sector, there must be a difference of 7 between the floor and the ceiling. This is the minimum. A difference of 6 will make it impossable to enter the sector. This is for players and typical monsters. Some monsters are a lot taller, and need even more room. If you find a monster doesn't move around when attacked, it's probably stuck in the ceiling or a wall. Also, a floor height difference of 3 can be climbed, but a difference of 4 is too high a step to climb. The player can fall down a step of any height, however.

Another option available in sector edit mode is the blend/copy function. You press **B** while in sector edit mode to get the blend/copy setup screen. Information from the hilighted sector is used to fill in information on this screen. You can change the initial sector information to be used, though, on this screen. You will also see adjustment factors to the right. This is how much it will change the initial values by each time. For example, having a floor height of 4, and an adjustment factor of +2 will make the first sector floor height 4, the next 6, the next 8, etc.. Using an adjustment factor of zero will basically perform a copy operation (the initial value never changes). This can be useful to set a bunch of sectors to one light value, or ceiling height, or whatever. If you select the button that's already on, you will turn it off, so that no adjustment factor is selected. When you do this, this Sector characteristic is ignored, and thus not changed when you blend/copy. This can let you change only the floor heights, for example, and not the light or ceiling levels. Or, you can simple change the ceiling to the sky texture for a group of sectors without changing anything else about the sectors. As you can see, this can be quite handy.

Once you have the setup set properly, select the Engage button. You will now be in blend mode. The top line will display what values you are currently at. To set a sector to these values, simply click on the sector, and the sector will be updated. This will also adjust each value by it's adjustment factor, and you are now ready to update the next sector. This is very useful for making stairs. Simply click on each stair, in order, until you are done. To get out of blend mode, simply press **Escape**, or the right mouse button (or **Delete** key). To get back to the blend mode setup screen, press **B** again, or the edit mouse button (or **Spacebar**).

Rectangle Creation Mode

This mode allows easy creation of polygons (rectangles, of course), essentially combining the steps of creating the lines and making a sector within them all into one step. If a rectangle is not be what you want in a situation, you can use vertex edit mode to move the vertexes around, and line edit mode to split lines, adding more sides to the polygon. To create a rectangle, simply press and hold the add/move mouse button at one corner point of the rectangle you wish to create, drag to the opposite corner point, and release the mouse button. It will instantly create the new rectangle.

Creating a rectangle inside an existing sector, thereby forming a donut sector, will be correctly handled by DMapEdit. All 4 lines forming the rectangle will have 2 sidedefs.

Marking

Objects can be marked to perform an operation on them as a group. Marking is done using the 'm' key, or clicking on an object with the marking mouse button. By default, there isn't a mouse button for this, so you will have to reconfigure the mouse buttons with the **Tab** key. You can also mark objects with a stretch-box by dragging with the mouse button, or pressing 'Alt-**M**'.

Objects that you mark will also mark other objects that they affect. For example, if you mark a sector, it will also mark all the lines, sidedefs, and vertexes that sector uses. Thus, you can mark objects in one mode, and then switch to another mode to do an operation on it, such as marking a sector, and then switching to line edit mode to change the lines of that sector. The backspace key will unmark all objects currently marked.

With a group of objects selected, you can move the whole group with the add button. You can delete the group (effect depends on the edit mode you are in) with the delete button. After you do a delete, there will not be any objects marked. And finally, you can edit the group of objects (vertexes can't be edited, since they have no properties). You would edit the objects just as if you were editing a single object, except changes will be made to all of the marked objects. To tell what will and won't be changed, the buttons will be normal or dimmed. If it is dimmed, this property will not be changed (changing only occurs when you exit the edit window). If you change something, but then decide you don't want to change this after all, you can toggle the button dimming with the middle mouse button (**Spacebar**).

When you have a group of objects marked, you can copy them to the clipboard and paste them back into the same map elsewhere, or another map altogether. Map slot zero is reserved as the clipboard (see the 'multiple maps' section for a discussion of map slots). Thus, you can switch to it as if it were a normal map and edit it. You can also save it to a PWAD once you have switched to it, or load a PWAD in, and that will now be what's on the clipboard, so you can paste it into other maps. While you are editing the clipboard, you have a different temporary clipboard which you can copy and paste from and to, but can't switch to, since you are already switched to the clipboard. Once you switch back to a real map and out of the clipboard, the temporary clipboard is destroyed and the normal clipboard you were just editing becomes the real clipboard again.

Making maps from scratch

Map generation starts with putting vertexes and lines where you want them. This lays the foundation for everything else. Once all the lines are in place, you need to make sidedefs for these lines (unless you do so along the way while you are creating the lines). The easiest way to do this is to select the "Fix Lines" options from the advanced options menu (F9). If you already have sidedefs setup, then you should at least error check the lines, just to make sure it's all ok. Once this is done, you need to make all the Sectors. If you have all the sidedefs where you need them, you can use the "Generate Sectors" option from the advanced options menu. This will put make sectors for every polygon on the map. Another method you can use is to simply enter sector edit mode and add sectors manually to each polygon. One advantage to this method is that you don't need to have the sidedefs in place. It will create the sidedefs if it needs to for lines as it goes along.

At this point, you have the vertexes, lines, sidedefs, and sectors all created. However, they are all most likely set to standard values (walls are all brown1 texture, sectors are all the same heights and textures, etc.), so you will need to go in and set them to the values you want them to be at. By the way, things can be added at any time along the process of making a new map. It's usually better to do so after the lines are drawn, though, so you know where to put them, but you are the boss.

When you have everything set the way you want it, it's time to generate Nodes, and a Blockmap. Making a Blockmap is fairly quick, so you might want to do this first. I also recommend you save your map before you generate these two structure. In fact, I recommend you save your map often! Anyway, once you have made your Blockmap and Nodes, save your map, and enjoy playing your new map!

Notes:

Maps must have all areas enclosed by lines, forming a polygon. A map with only 3 lines, for example, in an open ended box shape, isn't complete. Keep in mind that a line doesn't have to be a wall. Lines are required anywhere you wish to:

Change floor or ceiling height
Change lines angle
Change wall, floor, or ceiling patterns
Change lighting
Change floor hurt-player damage

Also, every map should have a starting point for all four possible players. You should also have, minimum, 4 deathmatch mode starting points. And, of course, you should have an exit.

Multiple Maps

As with most larger, more sophisticated programs, you are not limited in DMapEdit to just working on one map at a time. You can have several different 'windows' open at once. DMapEdit doesn't really support 'windows' as such, since a map view occupies the full screen. Thus, several different map views are contained in separate 'map slots'. A different map slot can either be a completely different map, or simply a different view of the same map that exists in another slot. You can switch between these slots either from the 'view' menu, or with Alt-**0** through Alt-**9** from the keyboard. Switching to an inactive (empty) slot will automatically make it a link to whatever map you just switched from, unless it is the clipboard (see below).

When two or more slots share a common map, they are said to be linked. This means that they are simply different views of the same map. They can have different zoom levels and positions, be in different edit modes, etc. They all are working on the same map, however, and changing anything in the map in one of these slots is changing the map itself, so that it is seen in all the linked slots. Clearing the map ('new map' option in file menu), loading a new map position or loading a new map from a different PWAD will all unlink the current slot, and won't affect any linked slots.

The current map slot number you are viewing is displayed at the bottom-right of the screen. You will see a number in brackets, like '<1>‘, or '[3]'. This number is the map slot number you are viewing, and the brackets specify the linked state. If the number is surrounded with angle brackets ('< >‘), then the slot is not linked. If it's surrounded with square brackets ('[ ]'), then the slot is linked with one or more slots. The checkmarks in front of map slots in the 'view' menu represent what slots are active (not empty), and have nothing to do with which slot you are viewing.

Polygon Generator

The polygon generator is designed to be easy to use. Once you have selected 'create polygons...' from the miscellaneous menu, you are in 'polygon' mode. From here, you must select the center point to build the polygon on in the map. Click the add/move mouse button on this point and drag the mouse until you have the size polygon you wish and release. While you are dragging out the size of the polygon, you can use the delete mouse button to cancel and exit polygon mode, or press the edit or mark mouse buttons and drag (along with the add/move button mind you) to adjust the polygon's center position, if you decide it's a little off. Once you release the add/move mouse button, the polygon is created. You will still remain in polygon mode, however. You can leave polygon mode by pressing the delete mouse button, pressing **Escape** on the keyboard, or switching to a another editing mode.

Before you begin dragging out a polygon, you can press the edit mouse button or the **Spacebar** to change the characteristics your polygon will have, including the number of sides it will have, if the points or sides should be aligned with the zero degree ray, and an angle rotation offset if you wish.

Convert sector to door

DMapEdit can automatically convert a sector into a door for you. It doesn't hurt to "fine tune" in from there, though. To convert a sector into a door, select this option from the miscellaneous menu, or press **D**. At this point, you will be asked to select the sector you wish to convert into a door. Just click on whatever sector you want.

Before you select the sector, you may also press the edit mouse button or the **Spacebar** to change the characteristics of the door to be created.

Staircase Generator

The staircase generator can be used to quickly and easily create staircases for you maps. You can only create staircases aligned in the X and Y directions, however, with the 'rotate marked object' option from the edit menu, you will be able make your staircase in any direction you wish (when I finally implement the 'rotate marked object' function).

To make a staircase, first select 'create staircase...' from the miscellaneous menu. You are now asked to specify the starting corner of the rectangular staircase you wish to create. Press the add/move mouse button and drag out your staircase. It will normally rise in the X direction, but you can switch it to rise in the Y direction by pressing the edit mouse button or the **Spacebar**. Once you have the staircase the right size, release the button, and you will go to the specifications screen, where you can set up the staircase with all the characteristics you wish.

3D Preview Mode

While the 2D top-view display is ideal for creating maps, it's not always easy to get a good idea of what the map is really like, because it has no depth. You can run your map in Doom and find out, but you have to rebuild the nodes and blockmap, wait for Doom to load up, and then finally twist your way around the map to get to where you want to look at. An alternative is to use the 3D previewer in DMapEdit to get a quick peek at the map without all that fuss.

There are 2 ways to activate the 3D preview mode. The first is simply to select '3D preview' from the view menu. The other is by using the hotkey '#'. The difference in using the hotkey is that you can center the preview origin on whatever (x,y) position the mouse cursor is on. In order to select 3D preview mode from the menu, you have to use the mouse, and so an initial origin can't be set using the menu method.

When you enter 3D preview mode, the zoom factor you were in is aproximated for the 3D view, so if you are zoomed out real far before going to 3D preview, the 3D preview will also be zoomed out real far. Also, the previewer will only render the lines/sectors you have marked, allowing you to narrow down what you want to see, so things don't get too cluttered, or to speed up rendering. If you don't have anything marked, the entire map will be rendered.

The renderer will utilize a bsp tree (nodes), if available, to render the view. While this may not really speed up the rendering, it will ensure that it is rendered correctly. Not using a bsp tree may cause the renderer to detect one wall in front of another wall, while in fact it's not, making the view look like an optical illusion.

While in the 3D preview mode, you can control the view in many ways. First, there are 3 mouse button functions to alter the view: rotation, scale, and position. These functions, their current values, and the button used for each, are shown at the top of the screen. You can click on one with another mouse button to make that mouse button control that function, just like you do with mouse button functions in normal editing mode.

In addition, there's a bunch of keyboard keys to control the rendering options:

W	Select wireframe rendering mode
S	Toggle solid walls rendering mode
X	Toggle X-ray viewing mode of backs of walls[1]
H	Toggle hiding of backs of walls
B	Toggle between black/gray coloring of backs of walls
O	Toggle axis display
C	Toggle texture majority coloring of solid walls
F	Toggle flicker-free rendering mode

Notes

1. Single-sided lines make single-sided walls, which mean you can only see the front of the wall. The back sides of these walls are never normally seen when running around in Doom. X-ray viewing lets you see right through these back sides, rendering them as wireframe, while the front sides of walls are rendered solid still. If X-ray viewing is off, you will see these back sides as solid as well.

Miscellaneous

Changing a map's Episode and Mission number

A commonly question asked, so I'll explain it, since it probably isn't that intuitive. Suppose you want to change a map from E1M1 to E2M2. How does one do this? Just press 'C' or select 'change map position' from the 'file' menu. Now, select the episode/mission you want it to be. Once you do this, just click the 'Ok' button as opposed to the 'load map now' button, and now your map has a new episode/mission value. At this point you can save it to a PWAD or whatever. This can be useful for taking several one-map PWADs, all at E2M1 perhaps, and putting them all into one big PWAD (since saving is non-destructive, and won't wipe out everything else in the PWAD).

You can also change the game type at this window. For example, if you have a Doom map, you can change it to a Doom 2 map by selecting the change game type button, and then selecting the game you wish to use from those available. A dimmed button means that DMapEdit detects you as not having that game type.

Alternate video modes

DMapEdit defaults to 640x480x256 colors, svga. You can select other video modes if you wish, however. 640x400 is the minimum, however. DMapEdit will not run in resolutions lower than this. 256 colors are also required. DMapEdit does not support more than 256 colors. To make DMapEdit run in other video modes, use the '**-v**' switch.

```
dmapedit -vmode
```

Mode: This is a number, from 0 to 4. These are what each represents:

0: 320x200 (VGA/MCGA) (Mode not usable with DMapEdit)
1: 640x400
2: 640x480 (default)

3: 800x600
4: 1024x768

Thanks

I’d just like to say thanks to the following people:

- **Bruce Benko:** For his help in doing a little coding, editing data files, providing ideas, beta testing, etc. He is also responsible for all those lovely Heretic shapes!

    - **John Bayley:** For the XPD program, and general beta testing/idea suggesting.

- **Paul Hagstrom:** For being my internet contact originally back before I had an account.

- **James Sterrett:** For the very first registration I received.

- **Alistair Brown:** For finding that insanity-causing bug in the node generator. I’m recovering quite nicely now..

- **Rod McCabe:** For his great letter and ideas. (quite a while ago, though)

- **Pierre Fournier:** For tracking down that intermittent bug in the picklist so well for me. I’d see it from time to time, but couldn’t figure out how to re-create it, until Pierre came along.

- **DEU authors:** For making their editor’s source available. I learned a few tricks from it.

- **PWAD creators everythere:** For making fun PWADs for me to play when I get bored!

- **Registered users:** For registering!

- **Cameron Booth:** Beta tester and idea suggestor.

- **joost schuur:** Beta tester and idea suggestor.

- **Matt Fell:** For creating the Unofficial Doom Specs, making this program even possible.

- **id Software:** For creating Doom and Doom 2 in the first place. Great game guys.

    - **Raven Software:** For creating Hexen, and doing such an excellent job on it. Nice to see some true innovation, instead of just rehashing of old ideas. Thanks.