WinDEU

FOR BEGINNERS

A Pictorial Guide

Version 1.1

By John W. Anderson

WinDEU for Beginners: A Pictorial Guide

Table of Contents

Introduction: Covering the Basics

This manual covers WinDEU version 5.24. There were two later versions, including one that edited HEXEN, but they never got much beyond the beta stage. Version 5.24 was the last stable version, and the version that I recommend using. WinDEU was written by Renaud Paquay for Windows, based on DEU by Raphaël Quinet and Brendon J. Wyber. DEU was the first level editor written for DOOM in February of 1994, and, in its day, the most powerful editor in town. Ten years later, there are very few tools that surpass it for its simplicity and ease of use. Many current editors can claim little more sophistication beyond a few bells and whistles. DEU's one drawback was its need for a DOS environment. The advent of Windows 95 made it much more difficult to access DEU, since it couldn't be run in a DOS window.

WinDEU is the Windows version of DEU, and it is every bit as stable and powerful as its predecessor. WinDEU went through several incarnations, including 16- and 32-bit versions for Windows 3.11. Version 5.24 was the best of the batch. By 1995, however, several other Windows editors (most famously, DoomCAD) had gained great popularity, and WinDEU settled into a rather ignominious obscurity. Those who liked DEU remained so fanatical and true to the editor that they would rather go through the trouble of booting into DOS and running DEU than using another Windows editor. I was one of them (though I had begun using a descendent of DEU called DETH). Why WinDEU never caught on, I'll never know. Why I'm writing a manual for an editor nine years old is easy: DOS is dead, and WinDEU is the only remnant of DEU that will run in Windows 2000 or XP. There are many other fine editors for Windows 2000/XP (Doom Builder comes to mind), but WinDEU is easy to run and to learn. There's no quicker way to get a level up and running. It also has the most powerful error checking tools around, and its texture alignment function works on the X *and* Y axis, which not many editors today will do. I believe once you understand the basics of WinDEU, even if you use another editor, you'll find yourself returning time and again to WinDEU for its power and simplicity.

Making levels is easy, once you get by the learning curve of the editor you're using. There are a number of good editors, and while most of them are fairly easy to learn, very few include clear instructions on the two most important aspects of making a DOOM map: constructing a simple Sector (or "room"), and connecting another Sector to the first. Sounds simple, and it should be: this is the foundation on which all levels are constructed. But very few editors come with tutorials that explain this most basic aspect of connecting Sectors. This manual will cover much more than that, but after a tour of the WinDEU interface in Chapter 1: Running WinDEU, we're going to get right to Sector construction. Chapter 2: Editing Basics is geared toward the beginner, and with getting a level up and running as soon as possible. My aim is to not take anything for granted, nor assume that you (the level designer) possess a knowledge beyond that of rank beginner. After all, none of us is born knowing how to make DOOM levels. We have to learn. And I'm going to teach you step by step. (I *am* going to assume you have a little knowledge of Windows. I can't go through everything; but I will get you up and running. Those of you who are Windows adepts will have to indulge me a little.)

I should mention that I'm going to provide not only the basics for making levels, but in-depth information on WinDEU, the short-cut keys it uses to make certain functions quicker, the jargon associated with DOOM mapping, and other references you'll need to make your levels the best. (See the Table of Contents for all the specific sections.)

But for now, let's get to it. I'm going to tell you how to get WinDEU up and running, explain the interface and commands, and then we'll start making a level.

John W. Anderson
04.04.04
drsleep@newdoom.com

SECTION 1

Running WinDEU

- *Configuring WinDEU*
- *Preferences*
- *Starting WinDEU*
- *The WinDEU Interface*
- *DOOM's Metrics*

1.0 Running WinDEU

1.1 Configuring WinDEU

Before you start running WinDEU, you need to edit the **windeu.ini** file with a text editor, such as Windows Notepad. (Plan on keeping Notepad or your text editor of choice handy. You'll also need it to create your WAD Author file, which most level designers include with their level when they distribute it on the Internet or other online services. I've provided a copy of the *WAD Authoring Template* in Appendix D, which is the standard template that nearly all designers use.)

To open Notepad, go to START | PROGRAMS | ACCESSORIES. To open **windeu.ini**, go to the FILE MENU, select OPEN, then LOOK IN C:\WinDEU. Click the drop-down list box FILES OF TYPE and click on ALL FILES *.*. You'll see **windeu.ini**. Double click on it. The only area you'll really need to edit is the one telling WinDEU what directory your DOOM or DOOM2 IWAD file is in.

```
#Name of the main wad file (un-comment only one of these 3 lines).
#main = e:\games\doom\doom.wad
main = e:\games\doom2\doom2.wad
#main = e:\games\heretic\heretic.wad
```

Change the third line to show what drive DOOM II is on, usually C:\Doom2.

```
main = c:\doom2\doom2.wad
```

or

```
main = c:\Program Files\Doom II for Windows 95\doom2\doom2.wad
```

By the way, if you want to edit DOOM or The Ultimate DOOM maps, you'll need to uncomment the first area and be sure to comment the second (stick a "#" in front of "main"), so the whole thing looks like this:

```
# Name of the main wad file (un-comment only one of these 3 lines).
main = c:\doom\doom.wad
#main = c:\doom2\doom2.wad
#main = e:\games\heretic\heretic.wad
```

There are a couple other areas you should adjust. Even if you're not an expert, uncomment the third line (delete the "#") to avoid annoying pop-ups, since this tutorial will be your guide:

```
# Set this if you think you are a brain.
# (Can be changed in the preferences dialog box.)
expert = true
```

WinDEU sometimes has trouble with the 3D Controls, which you don't need anyway. Uncomment the fourth line so that it looks like this:

```
# Set this if you don't want the dialog boxes to have a 3D look
# You MUST move CTL3DV2.DLL or CTL3D32.DLL in windows\system
# (Can be changed in the preferences dialog box.)
3DControls=false
```

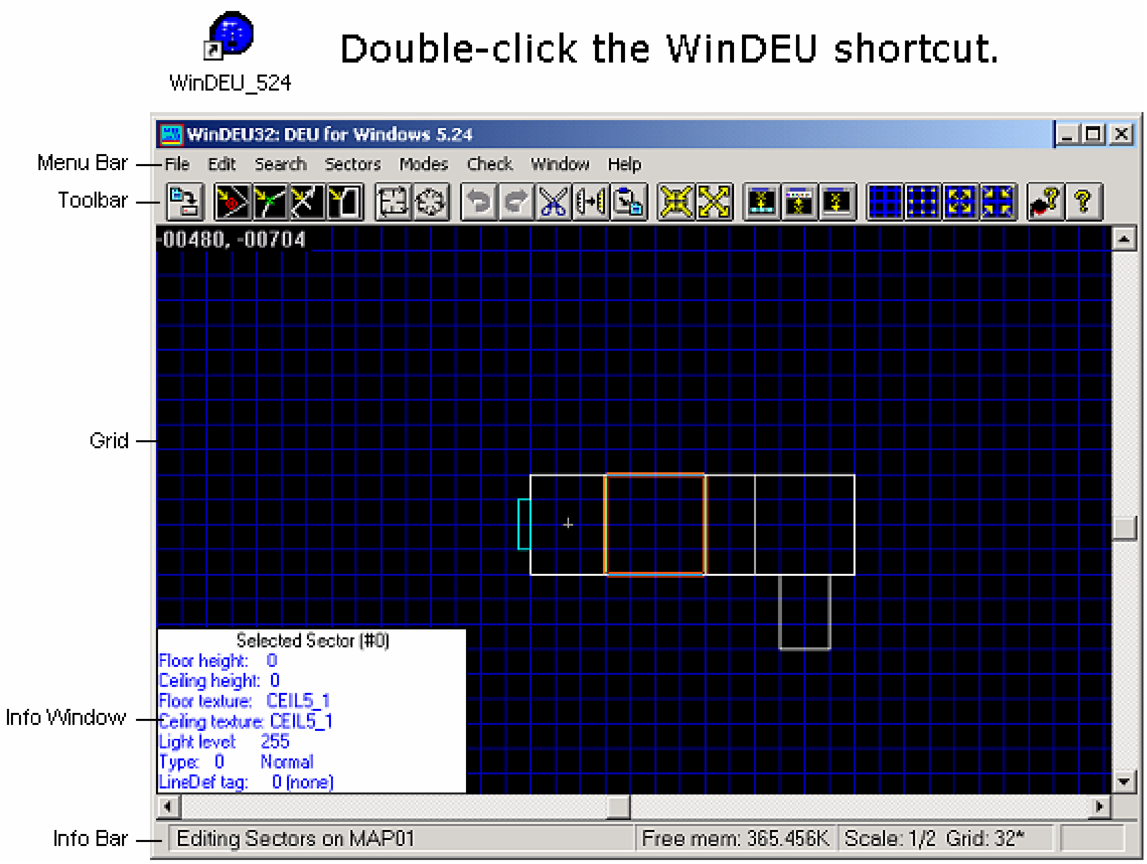
And finally, look at the line third from the end:

```
# Set this if you want the grid to be shown by default
#ShowGrid=true
```

Delete the "#" character so that your grid shows up every time you start the editor.

```
ShowGrid=true
```

Save your changes and fire up WinDEU!



Double-click the WinDEU shortcut.

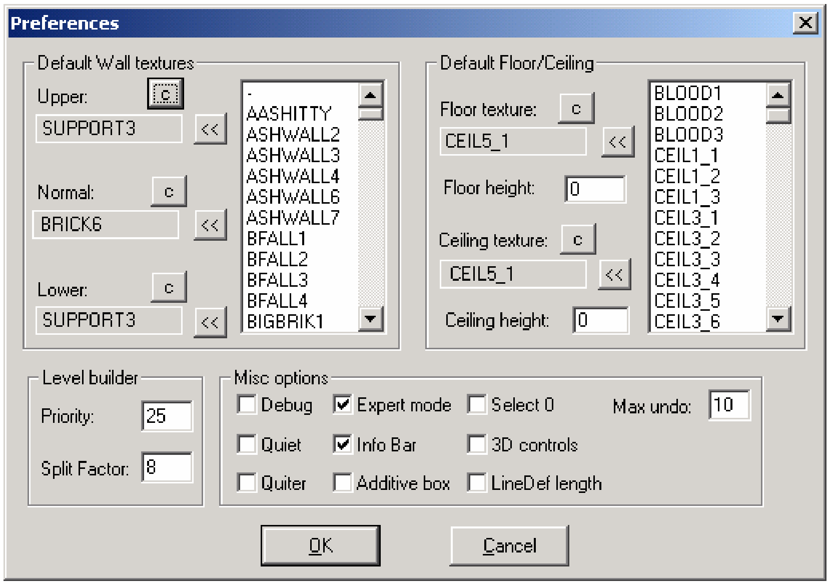
Selected Sector (#0)
Floor height: 0
Ceiling height: 0
Floor texture: CEIL5_1
Ceiling texture: CEIL5_1
Light level 255
Type: 0 Normal
LineDef tag: 0 (none)

Editing Sectors on MAP01 | Free mem: 365.456K | Scale: 1/2 Grid: 32*

1.2 Preferences

WinDEU's PREFERENCE settings can be accessed by pressing **F5** (or by clicking **EDIT** on the MENU BAR and choosing **PREFERENCES**). Many of the options you changed in the **windeu.ini** file can be changed here as well. Have a look at Figure 1.1.

.



Upper: SUPPORT3
Normal: BRICK6
Lower: SUPPORT3
Floor texture: CEIL5_1
Floor height: 0
Ceiling texture: CEIL5_1
Ceiling height: 0

Level builder
Priority: 25
Split Factor: 8

Misc options
☐ Debug ☑ Expert mode ☐ Select 0 Max undo: 10
☐ Quiet ☑ Info Bar ☐ 3D controls
☐ Quiter ☐ Additive box ☐ LineDef length

Figure 1.1.
Press F5 to access WinDEU's PREFERENCES dialog box.

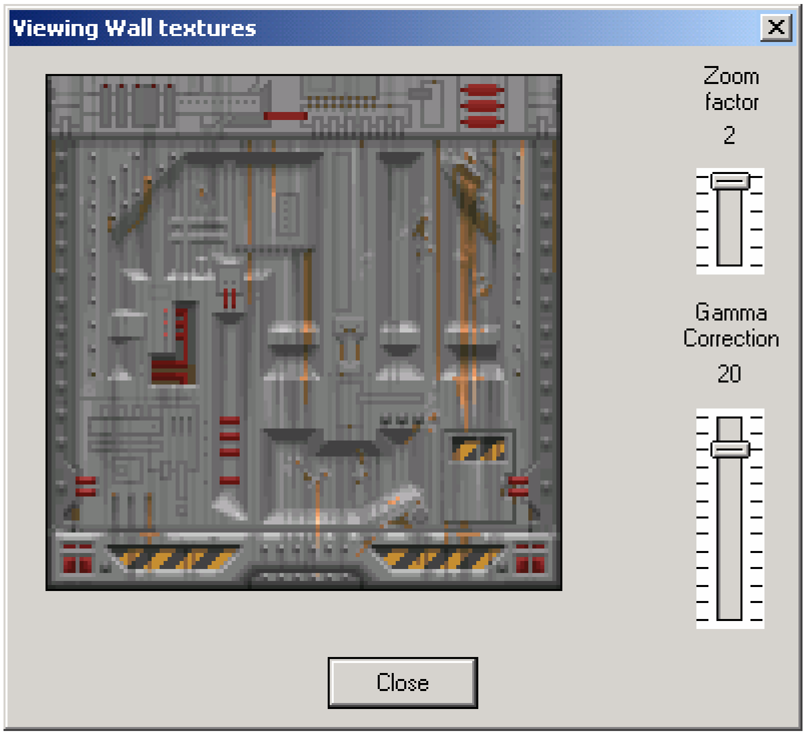
There are some convenient settings you can change here. First, let's make sure the defaults for our FLOOR and CEILING HEIGHTS are set at 0 and 128 (if they're different in your PREFERENCES, just enter the correct numbers in the text boxes next to each). Leave the settings in the **LEVEL BUILDER** area alone unless you're a DOOM Guru. The **MISC OPTIONS** area should have **EXPERT MODE** and **INFO BAR** checked. One you may want to change is **LINEDEF LENGTH**. When checked, tiny numbers will appear alongside your LineDefs when you move vertices from one grid to another. Some people like this feature. I keep mine off. (Your LineDef lengths are always available in the **INFO WINDOWS** when you're in LINEDEFS MODE.)

WinDEU uses certain textures and flats as defaults when it creates Sectors and SideDefs that require them. If you're partial to a certain set of textures or plan on using a particular texture a good bit throughout your level, you may want to enter them here so that you don't have to go back and replace every single texture in your level. In the **DEFAULT WALL TEXTURES** area, you can choose a texture from the scrollbar list and click the little double arrows box << to transfer the texture to the **UPPER**, **NORMAL**, and **LOWER** areas. **NORMAL** textures are those that occur on a typical wall (or SideDef, explained in Chapter 1.4.1.1). UPPER and LOWER TEXTURES are those that occur *above* and *below* windows, for instance. LOWER TEXTURES are also used on stair risers, the bottoms of lifts, and door tracks. The little **c** boxes c are used to *clear* the current texture name.

Double-click on one of the texture names in the scrollbar list to bring up WinDEU's **VIEWING WALL TEXTURES** dialog box. Isn't that great? You can keep the texture viewer open and scroll through the other textures with the arrow buttons by clicking on the PREFERENCE dialog box again. Now re-click one of the texture names in the list and press the ARROW-Down key.

You can bring up a separate flats viewer by double-clicking on the texture list in the **DEFAULT FLOOR/CEILING** area. This will overlap the texture viewer, so just click on the title bar and drag the window over a bit if you want to keep both of them open.

Figure 1.2.
Access the VIEWING WALL TEXTURES dialog box by double-clicking on a texture name in the scroll-down list.



You can change these options at any time by press **F5**. Your changes will be written to the **windeu.ini** file, by the way. Click OK to exit.

If you plan on following along and building the level shown in Section 2.0, you should set some default textures, flats, ceiling, and floor heights here.

1. Choose **BRICK6** from the texture list and transfer it to the NORMAL area of the DEFAULT WALL TEXTURES. Choose **SUPPORT3** for the UPPER and LOWER textures.
2. Choose **CEIL5_1** for the FLOOR and CEILING flats.
3. Set the FLOOR HEIGHT to **0**, and the CEILING HEIGHT to **128**. Click OK to exit.

The next chapter introduces you to WinDEU's start up options.

1.3 Starting WinDEU

Now that you have your PREFERENCES set, you're ready to start editing an existing map, or create a new level from scratch. For now, though, you need to familiarize yourself with the start-up options. The following sections explain the start-up TOOLBAR and how to open or create a new map.

1.3.1 The Start-Up Toolbar

After you start WinDEU and the program opens, you're presented with the main program interface. We'll get to the main WinDEU controls a little later, but for now note the TOOLBAR buttons, since these particular buttons occur only when you first start WinDEU. If you let your mouse cursor hover over each button, a "tool tip" bubble will appear telling you what each button does (the function also appears in the top Window bar). The first four buttons are for opening PWADs, editing them, and creating new levels.

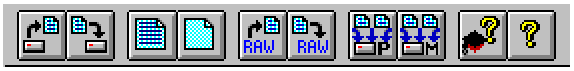
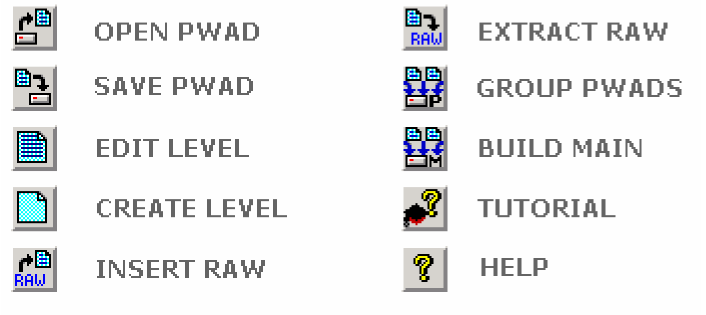


Figure 1.3.
The start-up TOOLBAR.

The next four buttons help you take advantage of WinDEU's powerful capabilities, since it can do much more than just edit and save a level for you. Here, you can also group your level with other levels into a single PWAD (just the thing if you're planning to create a 32-level replacement for DOOM2). Remember, a PWAD – or **Patch WAD** – can hold more than just one level. Don't confuse this with DOOM's main IWAD – **Internal WAD** – file, though. The DOOM2.WAD holds more than just levels: it contains all the textures, sounds, sprites, and other information that decorates and enhances the levels. PWADs can contain new textures and sounds too (something you may want to get into, depending on your mania).

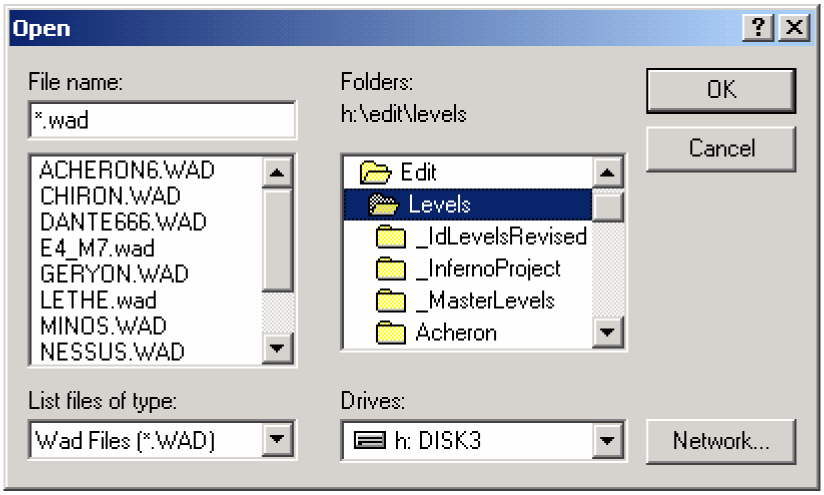
But for now, I just wanted to point out the place to go for more intricate editing details. We'll look at these advanced operations in Section 3.0. Some of the functions require specific knowledge of the objects contained in the DOOM2 IWAD file, so you shouldn't fool with these unless you know what you're doing. Required reading for understanding the DOOM IWAD is Matt Fell's *The Unofficial DOOM Specs v1.666*.

Figure 1.4.
WinDEU's start Toolbar.



To open an existing PWAD, click the first icon, **OPEN PWAD.** The OPEN dialog box comes up (Figure 1.3). This is a standard Windows OPEN dialog. WinDEU starts looking for PWADs in your C:\My Documents folder. Navigate to the directory where your PWADs are stored, select the PWAD you want to edit, and click OK.

Figure 1.5.
The open map dialog box.



File name: *.wad

ACHERON6.WAD
CHIRON.WAD
DANTE666.WAD
E4_M7.wad
GERYON.WAD
LETHE.wad
MINOS.WAD
NESSUS.WAD

Folders: h:\edit\levels

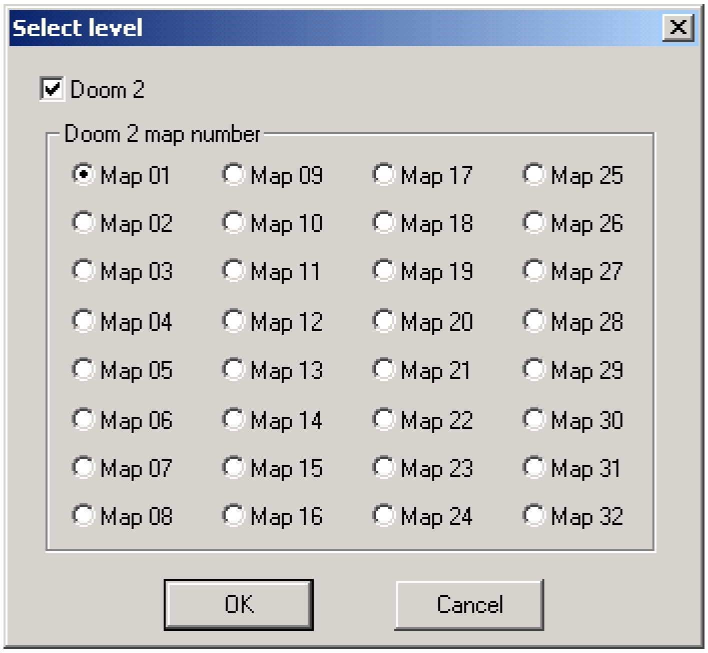
Edit
Levels
_IdLevelsRevised
_InfernoProject
_MasterLevels
Acheron

List files of type: Wad Files (*.WAD)

Drives: h: DISK3

To create a new, blank map, click the fourth icon, **CREATE LEVEL**. The SELECT LEVEL dialog box appears.

Figure 1.6.
Click the option button next to the map number you want your new level to be.



☑ Doom 2

Doom 2 map number

● Map 01 ○ Map 09 ○ Map 17 ○ Map 25
○ Map 02 ○ Map 10 ○ Map 18 ○ Map 26
○ Map 03 ○ Map 11 ○ Map 19 ○ Map 27
○ Map 04 ○ Map 12 ○ Map 20 ○ Map 28
○ Map 05 ○ Map 13 ○ Map 21 ○ Map 29
○ Map 06 ○ Map 14 ○ Map 22 ○ Map 30
○ Map 07 ○ Map 15 ○ Map 23 ○ Map 31
○ Map 08 ○ Map 16 ○ Map 24 ○ Map 32

Here you choose what map number to create. The map number is the directory entry in the PWAD that indicates what level the map will be. This is also called the *lumpname*. Aside from the music soundtrack and the sky changes, there is no difference between levels.

For the purposes of this tutorial, when you create a new map, choose **MAP01** by selecting the option button next to it. Click OK.

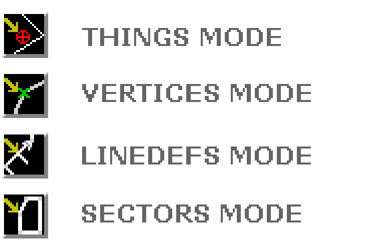
Later, you may want to assign different map numbers to your level, and you can change that here with WinDEU.

1.4 The WinDEU Interface

1.4.1 WinDEU's Editing Modes

Now you have your grid and you're just about ready to go. If you're in a terrible hurry, you can skip ahead to Chapter 2.1: Creating a Sector. I recommend you familiarize yourself with WinDEU and the objects that make a DOOM map possible, first. This chapter will explain the various components of WinDEU's interface, the TOOLBAR, MENU BAR, INFO WINDOWS, and a few hot keys to get you started. WinDEU has four modes of editing: **Things**, **Vertices**, **LineDefs**, and **Sectors**. These modes can be accessed by pressing their hot keys: **T** for Things, **V** for Vertices, **L** for LineDef, and **S** for Sector. You can also use the TAB key to switch between modes, or you can use the buttons on the toolbar.

Figure 1.7.
The edit mode TOOLBAR buttons.



Objects can only be moved or modified in their editing mode. Double-clicking an object, or highlighting it and pressing ENTER, will invoke the object's editing dialog box. The dialog boxes contain the settings for the properties of objects, such as texture images for walls, floors, and ceilings, ACTION TYPES that control how Sectors move, SECTOR SPECIALS that control Sector environments, and so on.

THINGS MODE allows insertion and modification of the properties of Things. Things are monsters, weapons, decorations, health pickups, player starts, etc. The THING EDIT dialog box contains the property settings for the THING TYPE, ANGLE and ATTRIBUTES, as well as an image preview of what the Thing looks like. Things are inserted into a level by using the INS key.

VERTICES MODE is for insertion and placement of Vertices. Vertices cannot be modified. Once created, they can be moved or deleted. The VERTICES EDIT dialog box allows changing the Vertex's X and Y coordinates. Vertices are created by pressing the INS key. Vertices define the start and end points of LineDefs.

LINEDEFS MODE is for creating and modifying LineDefs, including their FIRST and SECOND SIDEDEFS. LineDefs can be used to split Sectors, and can themselves be split by inserting a Vertex. Textures are placed on a LineDef's visible SideDefs through the LINEDEF EDIT dialog box. Properties for LineDefs include FLAGS, FIRST and SECOND SIDEDEF attributes, and a texture previewer that displays images of the textures.

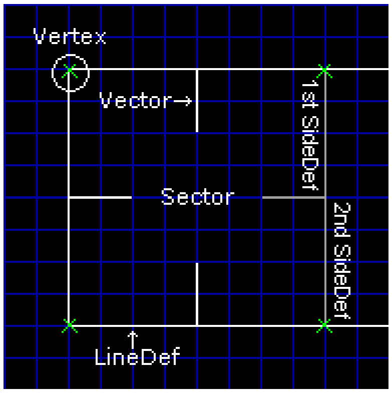
SECTORS MODE is for creating, moving, or modifying Sectors properties. In the SECTOR EDIT dialog box, Sectors can be assigned environment effects such as blinking lights and health damage. Floor and ceiling heights are modified here, as well as light levels and the floor and ceiling textures, more properly called *flats*.

1.4.1.1 DOOM Jargon for Level Designers

THINGS are enemies, weapons, keys, bonuses (such as armor and health packs), decorations (barrels, pillars, trees), hanging bodies, dead bodies, and light sources. Things can have an ANGLE which determines the direction they're facing, a FLAG that defines whether enemies are DEAF or not, plus a flag for assigning which SKILL LEVELS the object appears in. Things properties are edited in the THING EDIT dialog box.

VERTICES (plural of **VERTEX**) are the reference points defining the START and END of a LineDef, having an X, Y coordinate on the map. (Since DOOM is really 2-dimensional, there is no Z coordinate.) Vertices are not drawn in DOOM, but they serve a very real purpose. As you can see in Figure 1.7, they're the hinges between LineDefs and basically define the coordinates for Sectors on the X and Y planes. The DOOM grid unit is based on pixels for texture-mapping purposes.

Figure 1.7.
DOOM jargon diagram.



LINEDEFS – or *Line Definition* – represent lines from one Vertex to another. LineDefs may have an ACTION TYPE – sometimes called a *trigger* – that defines a function to be performed by a Sector (such as doors, lifts, switches, crushers, etc.) These LineDefs are activated by the player facing that LineDef and *using* the spacebar (or *use* key), or sometimes they may be walked over. LineDefs may also be assigned a TAG number that will cause all Sectors with the *same* TAG number to undergo the effects that the LINEDEF ACTION TYPE dictates. TAG numbers are not Sector numbers, nor LineDef numbers: they're an arbitrary identification number assigned by the level designer.

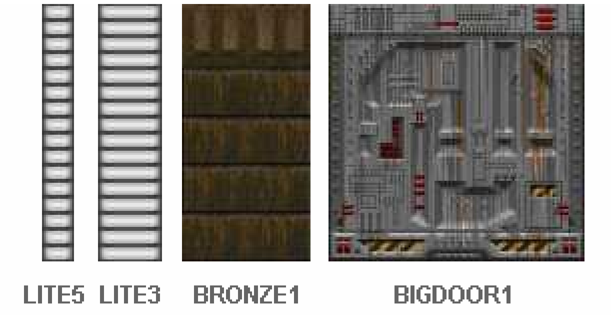
SIDEDEFS Each LineDef has at least one visible side called the FIRST SIDEDEF – which is always the RIGHT side – indicated by a **VECTOR**. A VECTOR is the little stick that juts out perpendicular to the LineDef. RIGHT and LEFT are based on the direction of the LineDef as indicated by the START and END Vertices. It can also have a second (or LEFT) side, called the SECOND SIDEDEF, if it adjoins two Sectors. SIDEDEFS actually define the boundaries in the map, and these boundaries define the borders of a Sector. An enclosed set of SideDefs is what comprises a Sector. In Figure 1.7 the LineDef on the east side of the main Sector has been labeled 1st SideDef and 2nd SideDef because it is shared between two Sectors. The VECTOR on the FIRST SIDEDEF is facing into its Sector of origin, and the SECOND SIDEDEF into an adjoined, shared Sector. Textures are assigned to the SideDefs – not the LineDef.

SECTORS are a horizontal (east-west, north-south) area of the map where ceiling and floor heights are defined. A Sector can be thought of as a room or an area inside a room. A "room" in DOOM may have many different Sectors. A set of stairs, for instance, is composed of many adjacent Sectors of varying floor and ceiling heights. Any difference in height, texture, or light intensity requires a new Sector. Your main Sector is called a PARENT SECTOR, and any Sector within it a CHILD SECTOR. Sectors are the only structures that move in DOOM. They can also have a SPECIAL SECTOR TYPE that describes an area-effect such as negative health (for a Sector containing toxic waste, for instance), secret, and certain lighting effects. Using separate

Sectors for special lighting effects makes a dramatic difference to a level. Sector properties can be modified in the SECTOR EDIT dialog box.

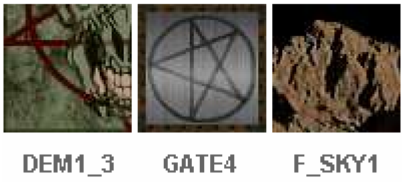
TEXTURES are the graphics that DOOM applies to all visible vertical surfaces. Normally, they're applied to the **FIRST SIDEDEF** of a one-sided LineDef – and thus are often called NORMAL TEXTURES. *But not always!* LineDefs can be two-sided, and if separated by varying Sector heights, will then have an UPPER TEXTURE and a LOWER TEXTURE, as in the case of some stair or window Sectors. This leads to the NORMAL TEXTURE also being called the MIDDLE TEXTURE, since it is placed between UPPER and LOWER TEXTURES. Textures are mapped from the *top down*, starting with the upper left corner and tiling horizontally to the right. SideDefs have X and Y OFFSETS to accommodate pasting, tiling, and alignment of the textures.

Figure 1.8.
Some common textures of different sizes: 16, 32, 64, and 128 wide by 128 high respectively.



FLATS are floor and ceiling textures. Unlike "wall" textures, FLATS cannot be manually aligned. The DOOM engine presupposes a fixed 64-pixel grid along which FLATS are automatically arranged. This means, for instance, that if you create a teleporter, you must build it on a 64-unit grid in WinDEU. If you don't, the teleport FLATS you apply to the floor will be misaligned and look goofy. So always set your grid in WinDEU to 64 when you plan on constructing a teleport Sector. (FLATS are assigned through the SECTOR EDIT dialog box.)

Figure 1.9.
Some common floor and ceiling flats.



IWAD – or *Internal WAD* – is the main DOOM or DOOM2 data base file. It contains all the information about graphics, sound, map/level data, etc. that is necessary to play the game.

PWAD – or *Patch WAD* – is an external file that has the same structure as the IWAD, but far fewer entries in the directory. The data in a PWAD is substituted for the original data in the IWAD. When a PWAD is loaded, only those resources listed in the PWAD's directory are changed: everything else is loaded from the IWAD. A typical PWAD usually contains new data for a single level, in which case it would contain the 10 lumps and 11 directory entries necessary to define the level. PWADs may contain many levels, however, and may even contain new textures, graphics, and sounds.

OBJECT is the generic, collective term used in this manual when referring to the specific map structures VERTEX, SECTOR, LINEDEF, SIDEDEF, and THING.

LUMPNAME refers to the level number of your map. This is the directory entry in DOOM2 PWADs that says **MAPNN,** and **ENMN** in DOOM1. Don't confuse LUMPNAME (MAP01) with the filename (TEST.WAD) for your map. A **LUMP** is just data in one of several different formats. Some contain graphics data; some contain level structure data. There are actually 10 different types of map **LUMP TYPES**, which typically appear in a PWAD. MAP01 is technically a marker under which a set of related lump formats appear: THINGS, LINEDEFS, SIDEDEFS, VERTEXES, SEGS, SSECTORS, NODES, SECTORS, REJECT, and BLOCKMAP. There are 13 other **LUMP TYPES** in the DOOM IWAD, some of which may be placed in a PWAD (such as the lumps for textures and sounds), but this requires different utilities and a deeper knowledge of DOOM than this manual covers.

MARK and **SELECT** are the terms used in this manual for highlighting objects in WinDEU. You **MARK** an object by *holding down the SHIFT key and left-clicking*. You **SELECT** an object by *holding the mouse cursor over an object*. **MARKED** objects are **green**. **SELECTED** objects are **yellow**.

The definitions above only begin to scratch the surface. Sectors, LineDefs, Vertices, and Things have many more properties that will be explained in detail when we get to Section 2.0.

1.4.2 The Toolbar

The TOOLBAR holds buttons for various modes and functions in WinDEU. You may or may not use the TOOLBAR frequently, depending on whether you're a mouse person or a keyboard person, since most of the icon functions are represented by hot keys. For some people it's quicker to press a key instead of moving the mouse cursor, but it doesn't hurt to know what the buttons do.

If you hold your cursor over each button, a tool tip bubble will appear showing what the button does. The first one you may want to remember is the **UNDO** icon, which you'll need if you're a human being. Many of the functions represented on the TOOLBAR are also available in the MENU BAR, and in the FUNCTION MENU that appears when you right-click on an object.

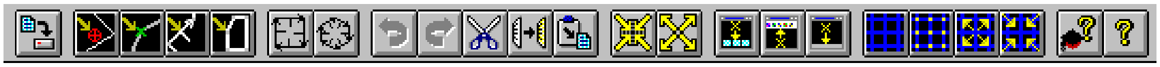
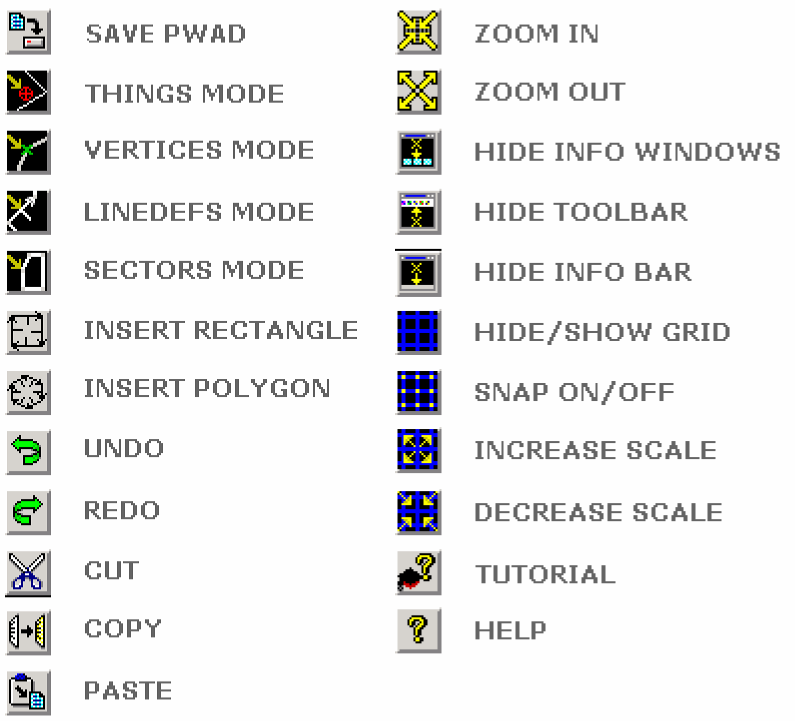


Figure 1.10.
WinDEU's main TOOLBAR.

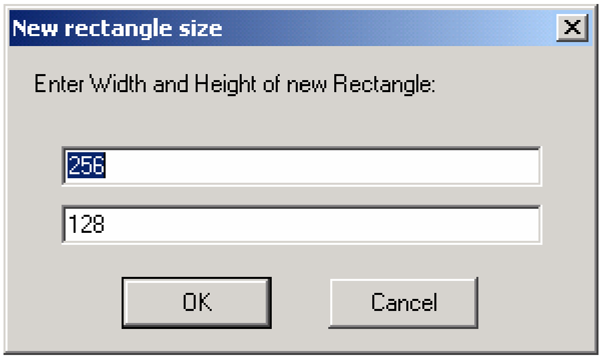
Figure 1.11 shows what each button does. These are explained in further detail below.

Figure 1.11.
TOOLBAR button functions.



Windows adepts will be familiar with SAVE, UNDO, REDO, CUT, and COPY buttons. The edit mode buttons have already been discussed in the previous chapter. Many of the buttons' functions are self evident; but some of the other buttons bear further explanation.

INSERT RECTANGLE is a quick tool for creating a Sector. When you click the button, the NEW RECTANGLE size dialog box comes up (Figure 1.12). Enter a WIDTH and a HEIGHT for the new Sector and click OK. Your mouse cursor turns into the INSERT CURSOR. Place the crosshairs over a position on the grid and left-click to place the Sector.



New rectangle size

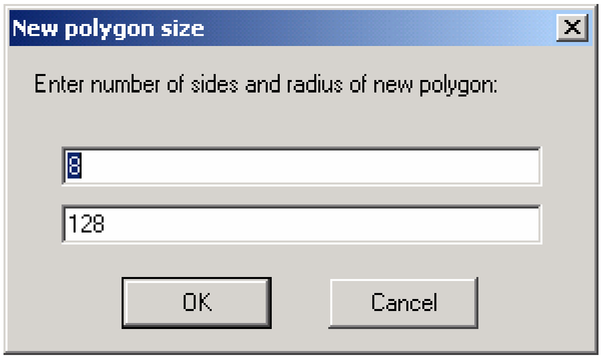
Enter Width and Height of new Rectangle:

256

128

Figure 1.12.
Click the insert rectangle button to create a new Sector.

INSERT POLYGON creates a circular Sector. The NEW POLYGON SIZE dialog box asks for the number of SIDES you wish the polygon to have. Enter a RADIUS for how wide you want the Sector to be. Position the INSERT CURSOR and left-click to create the polygon.



New polygon size

Enter number of sides and radius of new polygon:

8

128

Figure 1.13.
The more sides you enter, the smoother and more circular the polygon will appear.

ZOOM IN increases the magnification of your grid and makes the map appear larger. This function is also available by pressing the **[** key.

ZOOM OUT decreases the magnification of your grid and makes the map appear smaller. This function is also available by pressing the **]** key.

HIDE/SHOW GRID does just that. Click it once and the grid disappears. Click it again and the grid comes back on.

SNAP TO GRID causes the Vertex to jump to the nearest visible grid junction when you insert or move a Vertex. This is good to keep on in order to avoid sloppy structures.

INCREASE SCALE causes the grid to go to the next highest grid scale. Valid numbers are 8, 16, 32, 64, 129, and 256. The hot key is **SHIFT+G**.

DECREASE SCALE causes the grid to fall to the next lowest grid scale. The hot key is **G**.

1.4.3 The Menu Bar

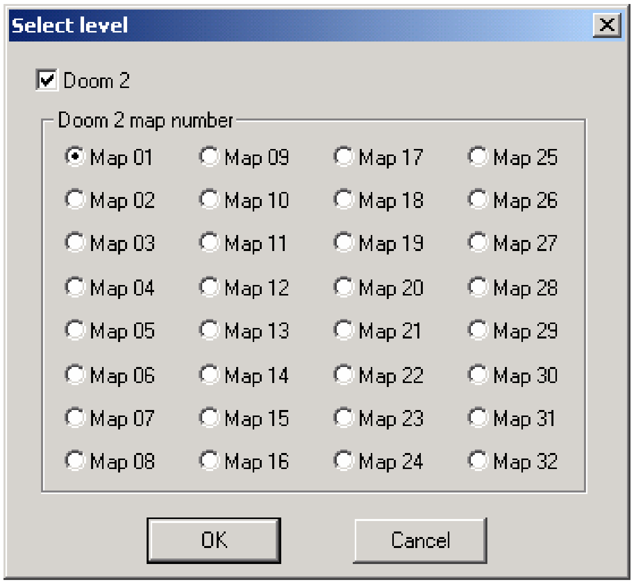
WinDEU has two separate MENU BARS. As you learned in the previous chapter on the TOOLBAR, WinDEU has two modes: START-UP MODE is displayed immediately after you start WinDEU and before you load a map to edit. The MENU BAR in START-UP MODE contains WinDEU's advanced functions. These are explored in detail in Section 3.0.

EDIT MODE is entered after you open a PWAD or choose to create a new level. The edit mode MENU BAR is the one examined in this chapter. Many of the functions in the MENU BAR are available in the TOOLBAR, the FUNCTION MENU, or through the hot keys. But since you won't have memorized all of the hot keys yet, we should go through the menus, just so you have an idea where to find certain tasks. There are eight menus, the fourth of which changes according to the editing mode you're in: VERTICES, SECTORS, SIDEDEFS, and THINGS. The other seven are FILE, EDIT, SEARCH, MODES, CHECK, WINDOW, and HELP. Some of these are fairly self-explanatory, so not all of the commands are covered below.

1.4.3.1 The File Menu

Aside from the standard SAVE and QUIT commands, the FILE MENU hosts the command for changing the level number of your map.

SAVE AS will save your map to a different file name if you wish, and will bring up the SELECT LEVEL dialog box. Click the option button next to the DOOM 2 MAP NUMBER you want your level to be and click OK.



Select level

☑ Doom 2

Doom 2 map number

● Map 01
○ Map 02
○ Map 03
○ Map 04
○ Map 05
○ Map 06
○ Map 07
○ Map 08
○ Map 09
○ Map 10
○ Map 11
○ Map 12
○ Map 13
○ Map 14
○ Map 15
○ Map 16
○ Map 17
○ Map 18
○ Map 19
○ Map 20
○ Map 21
○ Map 22
○ Map 23
○ Map 24
○ Map 25
○ Map 26
○ Map 27
○ Map 28
○ Map 29
○ Map 30
○ Map 31
○ Map 32

Figure 1.14.
The SAVE AS command lets you save your level to a different map number.

The standard SAVE AS dialog box comes up. Just enter the same or a new filename and click OK.

QUIT EDITOR closes the current map you're editing. You are prompted to save any unsaved changes, and then taken back to the start up interface. You can choose to open another PWAD, or you can quit WinDEU altogether by choosing EXIT from the FILE MENU.

1.4.3.2 The Edit Menu

The **EDIT MENU** contains commands for various editing functions and access to the PREFERENCES settings.

EDIT OBJECT will invoke the EDIT [OBJECT] dialog box for the current selected object. The same as ENTER or double-click.

ADD OBJECT inserts an object relative to the editing mode. The same as the INS key.

INSERT STANDARD OBJECT divides into two sub-menus: **RECTANGLE**, which brings up the INSERT RECTANGLE dialog box, and **INSERT POLYGON (N)**, which brings up the INSERT POLYGON dialog box.

PREFERENCES invokes the PREFERENCES dialog box, which allows setting program defaults.

1.4.3.3 The Search Menu

NEXT OBJECT highlights the next Thing, Vertex, LineDef, or Sector by number.

PREVIOUS OBJECT highlights the previous selected Thing, Vertex, LineDef, or Sector.

JUMP TO OBJECT brings up the JUMP TO A [OBJECT] NUMBER dialog box. Enter the number of the object you wish to look for. WinDEU will highlight the object.

1.4.3.4 The Modes Menu

The **MODES MENU** contains commands for entering the four editing modes: Vertices, LineDefs, Sectors, and Things.

NEXT MODE jumps to the next editing mode.

PREVIOUS MODE jumps to the previous editing mode.

1.4.3.5 The Check Menu

The **CHECK MENU** contains various error checking commands for your level. Few editors have error checking features as powerful as WinDEU's.

CHECK ALL SECTORS ARE CLOSED is the most valuable error checking tool. An unclosed Sector can cause DOOM to crash. Unclosed Sectors are those whose LineDefs do not all reference the Sectors they belong to. This is explained in detail in Chapter 2.7.

CHECK CROSS-REFERENCES as makes sure that all of your Vertices reference their proper LineDefs, and that your SideDefs reference their proper Sector assignments. It also clears any unused Vertices and LineDefs that may be transposed with or buried under others.

CHECKING TEXTURES looks for missing UPPER, LOWER, and NORMAL TEXTURES.

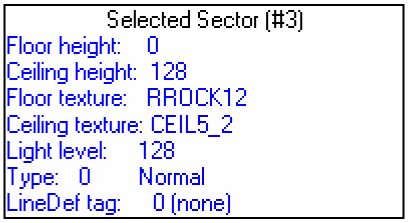
CHECKING TEXTURE NAMES looks for invalid texture names. This may only be a problem with your level if you've edited it with another editor that allows manual insertion of texture names.

DISPLAY STATISTICS brings up a list of the number and sizes of all the entities in your map: Things, Vertices, LineDefs, SideDefs, and Sectors.

1.4.4 The Info Windows

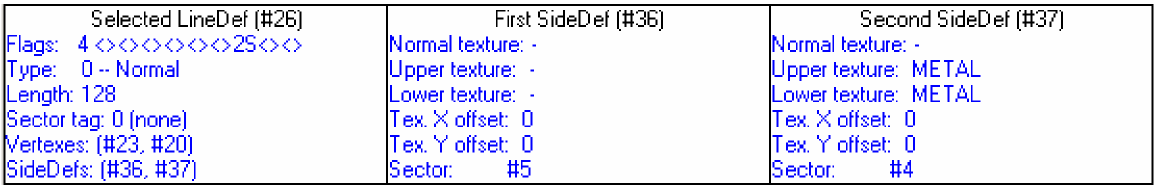
For each editing mode, an **INFO WINDOW** appears in the lower left corner of the main window. These windows display information about the object you've selected. For instance, when you select a Sector in **SECTORS MODE**, the window will display information about that Sector, as shown in Figure 1.15 below.

Figure 1.15.
Sector Information Window



```
Selected Sector (#3)
Floor height:    0
Ceiling height:  128
Floor texture:   RROCK12
Ceiling texture: CEIL5_2
Light level:     128
Type:   0        Normal
LineDef tag:     0 (none)
```

In LINEDEFS MODE, three INFO WINDOWS appear, showing statistics on first the LineDef itself, then the FIRST SIDEDEF and (if it exists) the SECOND SIDEDEF.



```
Selected LineDef (#26)
Flags:   4 <><><><><><>2S<><>
Type:    0 -- Normal
Length: 128
Sector tag: 0 (none)
Vertexes: (#23, #20)
SideDefs: (#36, #37)
```

```
First SideDef (#36)
Normal texture: -
Upper texture:  -
Lower texture:  -
Tex. X offset:  0
Tex. Y offset:  0
Sector:         #5
```

```
Second SideDef (#37)
Normal texture: -
Upper texture:  METAL
Lower texture:  METAL
Tex. X offset:  0
Tex. Y offset:  0
Sector:         #4
```

Figure 1.16.
LineDef Information Window.

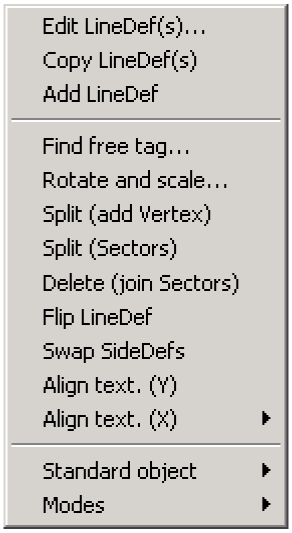
Likewise for Things and Vertices. These windows will be very important, since you will always want to know the ceiling and floor heights of your Sectors, what textures are being used, the length of your LineDefs (for when you come to aligning textures later in this document), and other valuable information. Just about everything you need to know is displayed here.

1.4.5 The Function Menus

To edit an object in any of the various modes, simply place the mouse cursor over the object until it is highlighted yellow and press **ENTER**. This will bring up the various editing dialog boxes for changing such things as textures, LINEDEF ACTION TYPES, lighting, floor and ceiling heights and flats, etc. You can also **double-click** on the object (Thing, Vertex, LineDef, or Sector) or **right-click** to invoke the FUNCTION MENU and choose **EDIT OBJECT**.

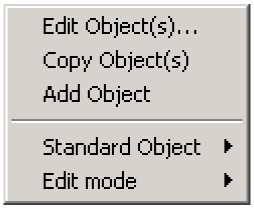
There are two FUNCTION MENUS: one is edit mode specific, and changes depending on the editing mode you're in. It is invoked by right-clicking on an object.

Figure 1.17.
Right-click on an object for the object FUNCTION MENU.
This is the LINEDEF FUNCTION MENU.



The second menu is a smaller, COMMON FUNCTION MENU (Figure 1.18) that appears if you right-click on the grid.

Figure 1.18.
Right-click on the empty grid to bring up the COMMON FUNCTION MENU.



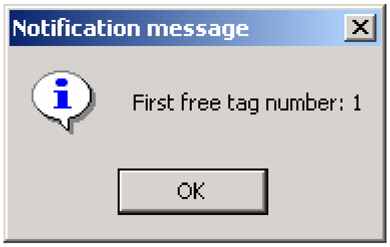
Notice that the FUNCTION MENU is context-sensitive. The commands on the menu vary depending on the editing mode you're in, and they're identical to the object mode menu in the MENU BAR. For instance, the functions in LINEDEFS MODE are identical to the commands in the LINEDEF MENU on the MENU BAR. There are also several commands that are identical in each mode, and are explained below.

EDIT OBJECT(S) will invoke the [OBJECT] EDIT dialog box for that object.

COPY OBJECT(S) will copy the object and its properties.

ADD OBJECT acts the same as the INS key. This will create an object at the position of your cursor.

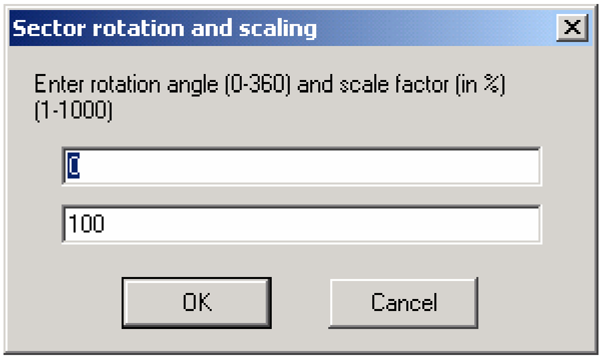
FIND FREE TAG looks for the next TAG number that has not been used. This feature also exists in the SECTOR EDIT and LINEDEF EDIT dialog boxes. TAG numbers are used to identify and link a LineDef with a LINEDEF ACTION TYPE to a Sector that will perform the function dictated by the ACTION TYPE. This is explained in Chapter 2.4.1.



First free tag number: 1

Figure 1.19.
The message displayed when you search for the next free TAG number.

ROTATE AND SCALE invokes the VERTEX ROTATION AND SCALING dialog box. This command will rotate an object by degrees and scale it by a percentage.



Enter rotation angle (0-360) and scale factor (in %) (1-1000)

0

100

Figure 1.20.
The SECTOR ROTATION AND SCALING dialog box.

STANDARD OBJECT divides into another menu with **RECTANGLE**, which is the INSERT RECTANGLE function and **POLYGON (N SIDES)**, which is the INSERT POLYGON function.

EDIT MODES divides into another menu with the four editing modes, **NEXT MODE**, and **PREVIOUS MODE**. These will take you directly to or switch you between editing modes.

Let's look at the other functions by their editing modes (there are no special function commands for THINGS MODE aside from the ones above).

•Vertices Mode

DELETE (JOIN LINEDEFS) deletes the selected Vertex and merges the two separated LineDefs into one.

MERGE will join two marked Vertices into a single Vertex.

SPLIT SECTOR will draw a LineDef between two marked Vertices and split the Sector into two.

•LineDefs Mode

SPLIT (ADD VERTEX) will insert a Vertex and split the LineDef.

SPLIT SECTORS will insert a Vertex in each LineDef (you must select at least two) and draw a LineDef between them, splitting the Sector.

DELETE (JOIN SECTORS) will delete a LineDef shared by two Sectors and join them into one.

FLIP LINEDEF will physically switch the FIRST SIDEDEF and the SECOND SIDEDEF. This changes the start and end Vertices, which determine a LineDef's RIGHT SIDE, which is the FIRST SIDEDEF. The Sector references are changed accordingly so the SideDefs reference the correct Sectors.

SWAP SIDEDEFS will switch the references for each SideDef without physically altering the directions of the FIRST and SECOND SIDEDEFS. For example, a LineDef shared between two Sectors will have a SideDef that references the Sector it faces into. If the FIRST SIDEDEF faces Sector 1, and the SECOND SIDEDEF faces Sector 2, **SWAP SIDEDEFS** will give the FIRST SIDEDEF a reference for Sector 2 (instead of Sector 1), and the SECOND SIDEDEF will reference Sector 1 instead of (Sector 2). This command does not change the direction of the SideDefs, as does **FLIP LINEDEF**.

ALIGN TEXT. (Y) aligns textures on the selected LineDefs along the Y axis (vertical plane). WinDEU has powerful texture aligning functions. Not all editors will align textures along the Y axis.

ALIGN TEXT. (X) aligns textures on the selected LineDefs along the X axis (horizontal plane). WinDEU will only align textures that are the same. It will not align textures if a LineDef selected has a different texture or if a texture is missing. The check function on the following menus will check to see if textures are missing or not the same. Texture alignment is calculated by looking at the lengths of LineDefs between Vertices and the width of the texture. This menu further divides into another with eight choices:

- **SD1, CHECK** looks for textures along the FIRST SIDEDEFS that are missing or not the same as the first selected LineDef. If it finds that there is a different texture, WinDEU will display a notification telling you which SideDef has a different (or no) texture. If the textures are all present and the same, this function will align all textures on the selected LineDefs, using the first selected LineDef as the reference point.

- **SD1, CHECK AND OFFSET** does the same as the previous, only it allows you to define a beginning offset to start from. A dialog box will come up where you can enter the offset.

- **SD1, NO CHECK** aligns textures without checking for missing or different textures.

- **SD1, NO CHECK AND OFFSET** does no checking and allows you to define a beginning offset.

The following choices perform the same functions as the previous four, but along the SECOND SIDEDEF:

- **SD2, CHECK**.

- **SD2, CHECK AND OFFSET**.

- **SD2, NO CHECK**.

- **SD2, NO CHECK AND OFFSET**.

MAKE DOOR will turn the selected Sector into a manual door, ACTION TYPE **1 DR OPEN DOOR**. The door will be assigned BIGDOOR2 textures on the front SideDefs.

MAKE LIFT will turn the selected Sector into a generic lift, with a *walkover trigger* ACTION TYPE **88 WR LOWER LIFT** assigned to the appropriate LineDefs. If the floor height of one of the adjoining Sectors is lower than the lift Sector, it will change the shared LineDef into a *switch*, ACTION TYPE **62 SR LOWER LIFT**, and flip the LineDef so that the FIRST SIDEDEF faces into the lower Sector (*switches* must be activated from the RIGHT SIDE (FIRST SIDEDEF) of a LineDef). The texture **SHAWN2** will be applied to the *lower textures* on the visible SideDefs.

DISTRIBUTE FLOOR H. will adjust the floor heights of Sectors with different ceiling heights. You must select at least *three* Sectors.

DISTRIBUTE CEILING H. will adjust the ceiling heights of Sectors with different floor heights. You must select at least *three* Sectors.

1.4.6 WinDEU's Hot Keys

A full list of **Hot Keys** are at the end of this manual in Appendix A. For now, you just need a few to maneuver between editing modes, and they're easy. **T** is for Thing Mode, **V** is for Vertices Mode, **L** is for LineDef Mode, and **S** is for Sectors mode. You can also use the **TAB** key to switch to the next mode, or you can click on the group of four black-background icons to go into a particular mode. You'll need to remember the **SHIFT** and **INSERT** (**INS**) keys for adding LineDefs and Sectors. Also the - and + keys for zooming in and out, and **G** and **SHIFT-G** for changing grid scales. (This is important because you'll want to change scales all the time for detail work. The grid represents DOOM units and the lowest grid number is 8. The highest is 256. WinDEU starts out with a default of 32, which is good for basic Sector creation and modification. You need to know this because of DOOM's metrics, which I'll cover briefly in the next section.)

These hot keys again are:

WinDEU Hot Keys	
T	Thing Mode
V	Vertices Mode
L	LineDef Mode
S	Sectors Mode
TAB	Move to next mode
INS	Create Sector
SHIFT+INS	Add LineDef or Sector
-	Zoom Out
+	Zoom In
C	Clear Selected
G	Increase Grid
SHIFT+G	Reduce Grid
SHIFT+MOUSE1	Mark Object
ARROWS	Scroll
ENTER	Edit Object
Q	Quit
ESC	Exit (no save)
F2	Save

I'll remind you of the various hot keys as we go along.

1.5 DOOM's Metrics

DOOM is binary, or scaled at powers of 8. This goes for your textures and flats (flats are floor and ceiling textures). You need to know this when scaling your floor and ceiling heights so that the textures fit correctly on your "walls" (SideDefs). Most textures are 128 high by 64 wide or 128 by 128. This doesn't mean that every room needs to be either 128 or 256 or 384 high and so on. You can get away with different heights depending on the way textures tile beside one another. But you should be consistent in your choices in order for your level to look professional. Door textures come in only two sizes (with one exception) - 128x64 and 128x128. So your door Sectors must match the textures, otherwise your doors will look just plain sloppy or goofy. The same goes for stairways and small steps. There are specific textures to use for the side of steps and stairs (but you're not limited to these), and they're usually 16x32. These textures work for steps that are only 8 units high, also. When you go to apply textures to your level you'll see that WinDEU lists the sizes of each texture for easy reference.

Your flats (floor and ceiling textures) come in one size: 64x64. And they're scaled to the grid in 64 unit sections by the DOOM engine. In other words, you can't align flats manually, and you need to keep this in mind when making your rooms and hallways – and especially your teleporters! (A teleporter has to be constructed on a 64x64 grid unit or the texture for it will be messed up. Keep that in mind when planning a teleporter. Teleporters are also traditionally 8 units high with a plain metal texture surrounding the base.)

The player height limit in DOOM is 56 units. That's the smallest opening a player can get through. So don't make an area with a height lower than this, unless you don't intend for the player to be able to get through it. (Small Sector sizes can be used for other reasons, though!) Also, the player's step distance is 24 units. So anything higher than 24 will be impassable. A good size for steps is either 16 or 8 units. The smallest player width is 33 units. Anything less wide will be impassable. (Keep in mind DOOM's monsters who have different height and width properties. A Cyberdemon cannot fit through most of the same doors as an Imp or Demon. A list of monster sizes appears in the appendix of this manual.)

Scott Amspoker has written an excellent guide outlining these different size considerations in Appendix B: DOOM Metrics, which is included at the end of this manual. (He's also written a nice guide on textures called *Managing Textures and the "Unpegged" Attribute*.)

Finally! We're going to start building a level.

SECTION 2

Editing Basics: Making a Level with WinDEU

- *Creating Sectors*
- *Adding Sectors*
- *Modifying Sector, LineDef, and Thing Properties*
- *Making a Door*
- *Making a Lift*
- *Making a Teleporter*
- *Checking Your Level for Errors*
- *Saving Your Level*

2.0 Editing Basics: Making a Level with WinDEU

Section 2.0 will guide you through the process of creating a DOOM level with WinDEU. You've learned all about the WinDEU interface in Section 1.0, but that was all preparatory to your actually sitting down and constructing a level Vertex by Vertex, LineDef by LineDef, and Sector by Sector. It's going to be fun. Start WinDEU and let's go.

First, you need to create a blank level. Click the CREATE LEVEL button in the start-up TOOLBAR. The select level dialog box comes up. MAP01 is chosen by default, which is what we'll use. DOOM starts with MAP01, so you won't have to "warp" to get to your level (if you were to make it MAP02 or anything else). Click OK. You're ready to create a Sector.

2.1 Creating Sectors

Since it's best to follow the diagrams in this document, make sure your grid is set at **32**. Look in the lower right corner of the INFO BAR, which is the very bottom bar of WinDEU. Press **G** and you'll see the setting increase to 64. Press **SHIFT-G** to decrease the grid setting. Or you can click the TOOLBAR grid buttons to do the same. Next to the *grid* setting in the INFO BAR is the *scale* setting. Zoom in using the + key (or the first of the Zoom icons in the TOOLBAR) until you're at scale 1/2. Now you'll be building at the same scale as the manual diagrams.

There are two ways to create a Sector in WinDEU. Go into VERTICES MODE by pressing **V** (or clicking the VERTICES MODE button on the TOOLBAR). Now zoom in on your grid so that you have a nice tight view, as in the picture below. (To zoom in or out use the **+** and **-** keys, or note the fifth set of two X-shaped icons on the toolbar.) We're going to create four Vertices in a square on the grid. Note that the map coordinates are shown against a black background in the upper left-hand corner of the grid. Try to place your cursor close to X +00000, Y +00000.

1. To place a Vertex, position your mouse cursor over a junction on the grid and press the **INS** key. You should see a green X, which represents a Vertex.

2. Do this again three times in a *clockwise* manner, from A to D, until you have four of them, as shown in Figure 2.1.

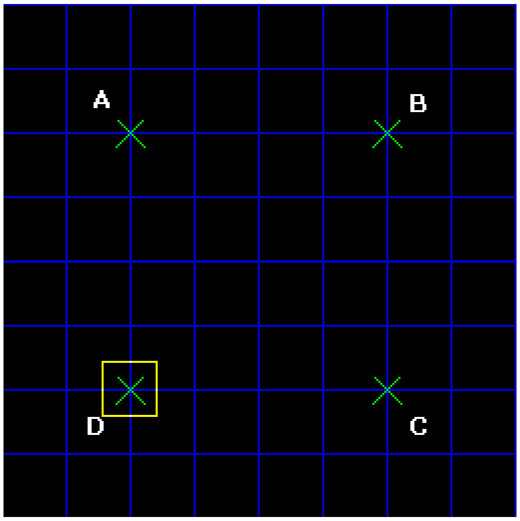


Figure 2.1.
Insert four vertices and then **mark** them in the order shown, A-D. Marked objects are highlighted green.

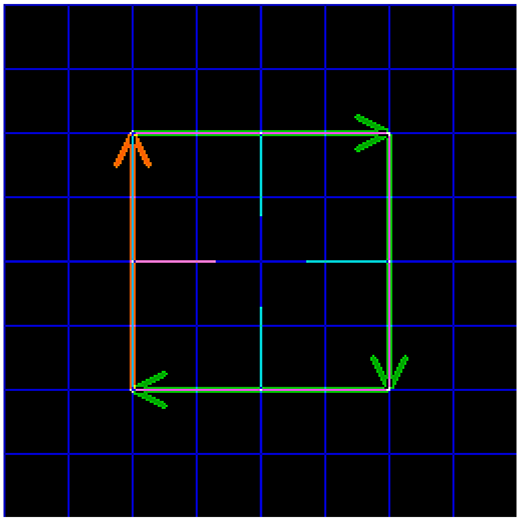
Now, let's **mark** each Vertex. When you hold (or even pass) your cursor over a Vertex, a yellow box will appear around it, showing it has been **selected**. *To **mark** a vertex, hold down the **SHIFT** key and **left-click**.* The box turns briefly red, then stays green.

It's very important to mark the Vertices in a *clockwise* order.

1. Mark each Vertex, in the order shown in Figure 2.1: **A**, **B**, **C**, then **D**.
2. Hold the **SHIFT** key down and press **INS**.

You've created a Sector!

Figure 2.2.
Hold down the SHIFT key and press INS. You've created a Sector! This is Sector 0.



Note that WinDEU has switched you into SECTORS MODE, and all four LineDefs are marked in green. Hit the **C** key to **clear** the marks. Pass your mouse over the Sector and it should light up yellow. If it does, you know that your Sector is valid. All new objects are assigned a number starting with zero, so this is Sector 0. Sector numbers are shown in light gray in the diagrams in this manual (see Figure 2.2).

The reason why you must insert and mark objects in a clockwise direction is this: a LineDef's RIGHT SIDE, or FIRST SIDEDEF, is determined by the position of the *start* and *end* Vertices. All LineDefs must have at least one side, and that side must be the RIGHT SIDE – the FIRST SIDEDEF. The FIRST SIDEDEF is the visible SideDef and must face into the Sector. The FIRST SIDEDEF is indicated by a **VECTOR** – the little stick that juts out perpendicular to the LineDef, as shown in Figure 2.2. If you look at the LineDef highlighted in red, its FIRST SIDEDEF is indeed the RIGHT SIDE. If you had marked the Vertices counter-clockwise, the FIRST SIDEDEF would have been facing into Void Space.

Sometimes you will need to draw counter-clockwise on purpose, but this is only when you're creating a Sector within another Sector. For now, make it a rule of thumb to insert and mark objects clockwise.

NOTE: Void Space is any area outside of a Sector – or inside of a Sector – from which no SideDefs are visible. A pillar inside a Sector is usually constructed as a square with the FIRST SIDEDEFS facing into the PARENT SECTOR and with no SECOND SIDEDEF. It is filled with Void Space.

You'll learn how to add Sectors and join them in the next chapter.

2.2 Adding Sectors

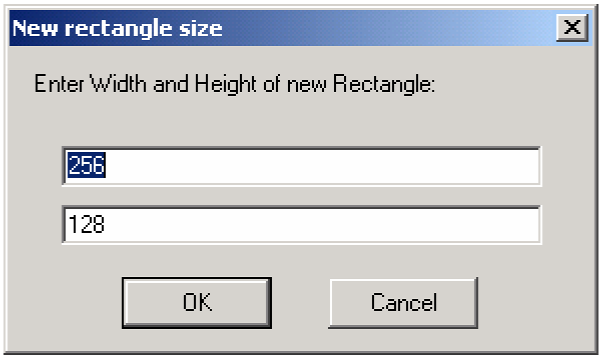
Adding Sectors in WinDEU is easy. You're going to learn a quick way to create another Sector, and you'll probably wonder why I didn't show it to you in the first place. It's important to understand how Sectors are constructed, and the best way to learn is to create one step-by-step by inserting Vertices and connecting them to form the shape you want. You'll be working mostly with squares in this manual, but please remember that you're not limited to them. Using square shapes for tutorial purposes is simply for expediency's sake and for the beginner. A Sector can be any shape at all, so long as it has at least three LineDefs and is self-contained.

2.2.1 The Insert Rectangle Function

There is an easier way to create a Sector, using the INSERT RECTANGLE function.

1 Click the INSERT RECTANGLE icon on the TOOLBAR. A dialog box appears asking you for the dimensions of your rectangle.

Figure 2.3.
INSERT RECTANGLE dialog box.



Enter Width and Height of new Rectangle:

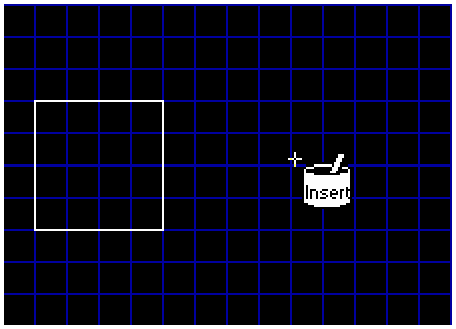
256

128

2 Enter **128** by **128** and click OK. Your mouse cursor changes into the INSERT CURSOR.

3 Position the little cross on the junction of the grid to the right of your first Sector, making sure there's room for a 128x128 Sector. Left-click to create the Sector.

Figure 2.4.
Position the INSERT CURSOR and left-click.



Notice that by using this option to create Sectors you don't have to go through all the fuss of inserting Vertices and marking them and all of that. It's important, though, that you know how to make a Sector by placing and connecting vertices, because sometimes you won't be able – or won't want – to use the INSERT RECTANGLE tool. You'll see why shortly.

Now we're going to join the Sectors.

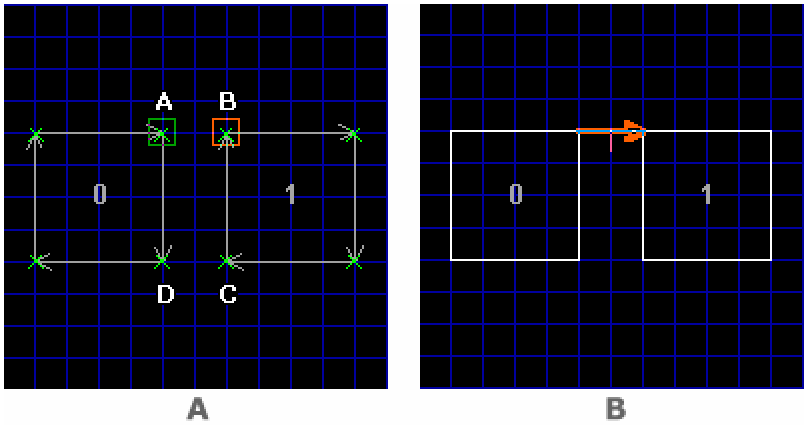
2.2.2 Joining Sectors

Joining Sectors is the foundation of making levels. You're going to be doing this again and again. There are a couple different ways to join two Sectors. The first way, you simply create a Sector between them. This is ideal for creating windows between rooms, or a door, or perhaps a lift. We'll be doing all of those, but first let's join the Sectors. To join Sectors, we need to switch back to VERTICES MODE (hit V). Follow the diagrams below.

I've labeled the Vertices on the boundaries between our two Sectors as **A, B, C,** and **D**, but we'll be doing this in two steps.

1. Mark Vertex **A** and then Vertex **B** (recall, you mark an object by pressing the SHIFT key then left-clicking on it. Remember to release the SHIFT key after each operation).
2. Press **INS**.

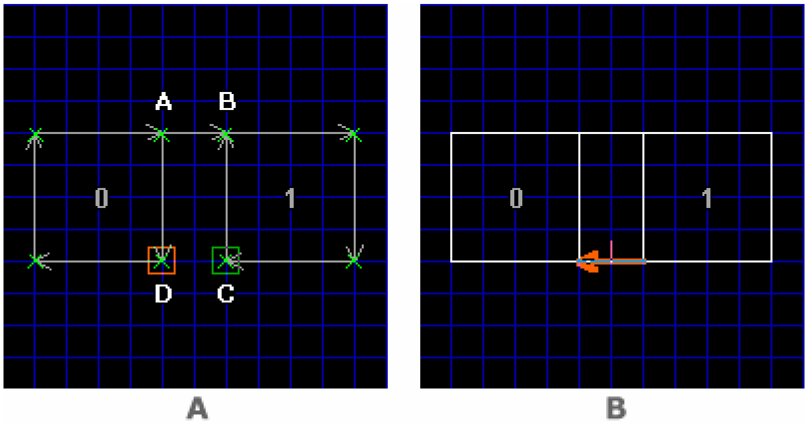
Figure 2.5.
Mark Vertex **A** and then Vertex **B**. Then press INS.



You've created a single LineDef, and WinDEU has switched you into LINEDEFS MODE.

3. Go back to VERTICES MODE (press V). Now mark Vertex **C** and then Vertex **D**. Press **INS**.

Figure 2.6.
Mark Vertices **C** and **D** and press INS.



Hit **C** to **clear** the marks. *For a moment, let's quickly go into* SECTORS MODE *(press S). Note that our new LineDefs are shown as red instead of white. That's because they are not bound to any Sector yet.)*

Go back into LINEDEFS MODE (press L). Now, starting with our first new LineDef, we're going to mark the LineDefs that define the boundary of our new Sector. See Figure 2.7.

1 Mark LineDef **A**, then **B**, then **C**, then **D** as above. Press **INS**.

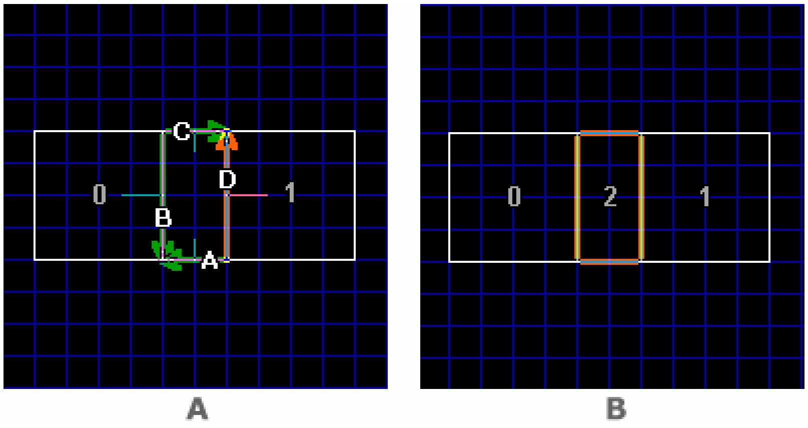


Figure 2.7.
Mark LineDefs **A**, **B**, **C**, and **D** and press INS.

We've now joined our two Sectors with a third in between them (see Figure 2.7b).

As I mentioned, there are many ways to create more Sectors as an offshoot of the first. Here's another. Let's branch our first Sector off to the left.

1 Go into VERTICES MODE. Insert two Vertices – **B** and **C** – as shown in Figure 2.8a.

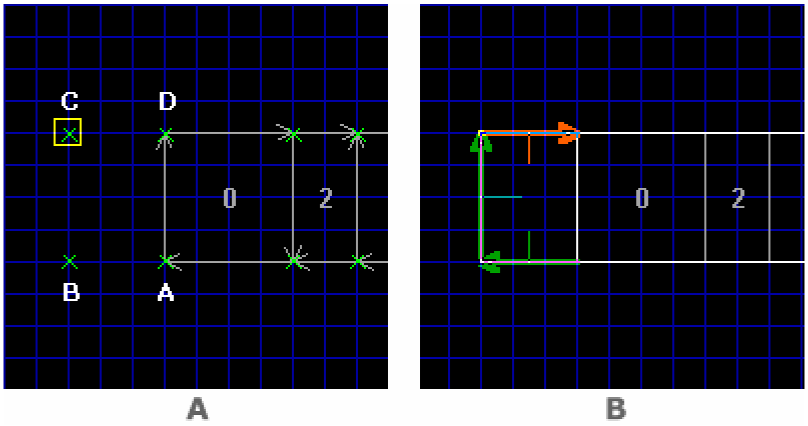


Figure 2.8.
Add two vertices B and C. Mark the vertices in order, A through D.

The Vertices are labeled in the order in which to mark them (clockwise).

2 Press **SHIFT** and **left-click** on Vertex **A**, then **B**, **C**, and **D**. Release the SHIFT key.

3 Press the **INS** key.

You've got three new LineDefs highlighted (Figure 2.8b). But they're not a Sector yet.

Clear them. And yes, we're going to mark them again, adding the LineDef from the existing Sector as our fourth.

4 Mark each of the new LineDefs in the order shown in Figure 2.9a: **A, B, C, D**.

5 Press the **INS** key to create the Sector.

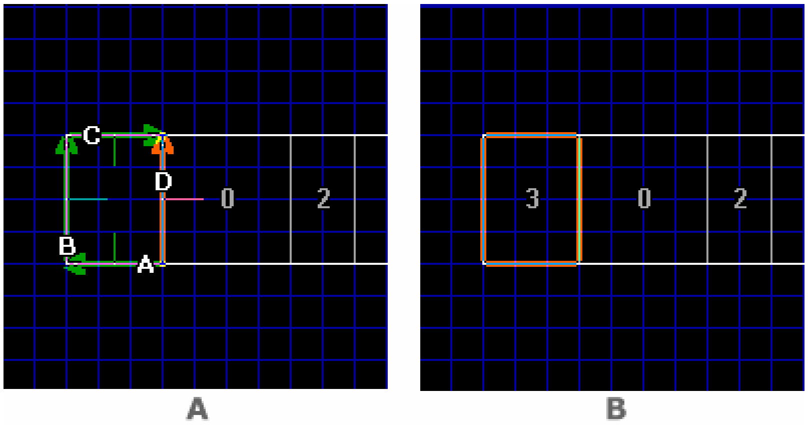


Figure 2.9.
Mark the LineDefs in the order shown, A through D. Press INS.

You've created Sector 3. And now here's the second way to add a Sector, which is actually the easiest of all.

1 Click on the INSERT RECTANGLE button. Enter the dimensions 64 by 64 in the dialog box.

2 Position the INSERT CURSOR below the existing Sectors (as shown in Figure 2.10). Left-click to create the new Sector.

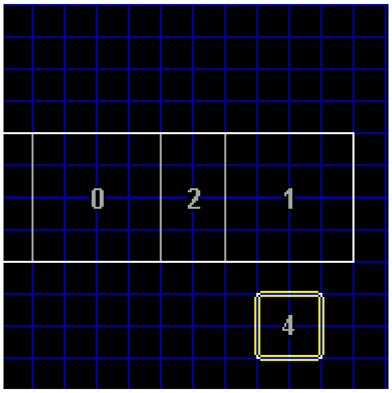
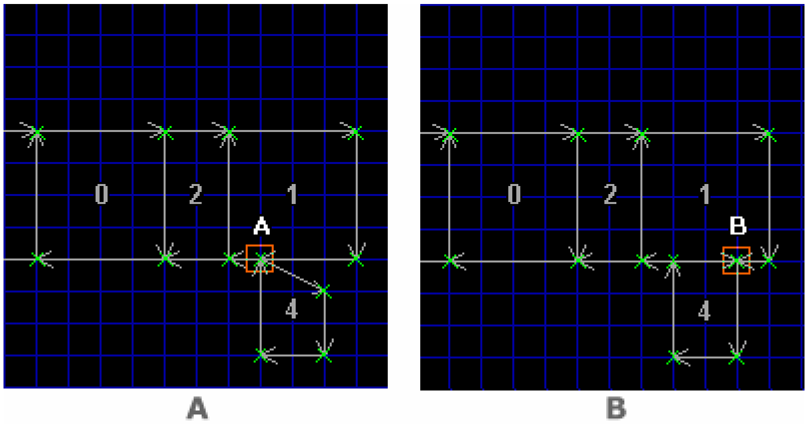


Figure 2.10.
Add a new Sector in the position shown with the INSERT RECTANGLE tool.

To join this new Sector to the one above it, we're simply going to **drag** the two top Vertices (A and B in Figure 2.11) to intersect the bottom LineDef of the Sector we're adding it to.

> **NOTE**: To **drag** an object, left-click on it and hold the left mouse button down. Move the mouse to *drag* the object. Release the button to drop the object to the new position.

1 Go into VERTICES MODE. As shown in Figure 2.11, first drag Vertex **A** to the position shown.



A

B

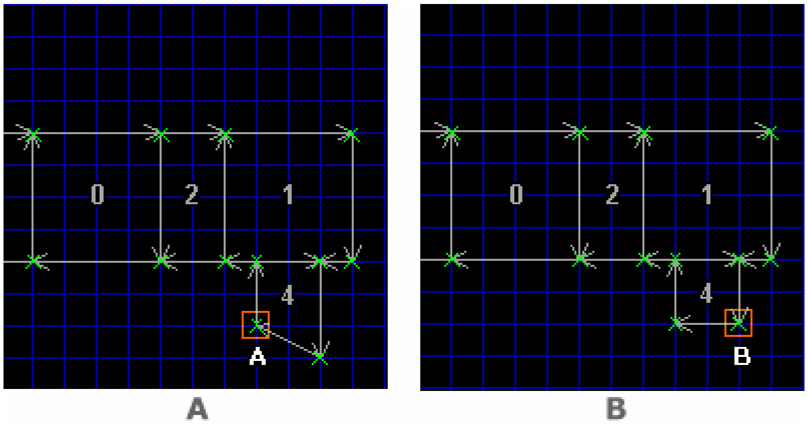
Figure 2.11.
Drag Vertex A to intersect the LineDef above it.

2 And then drag vertex **B**.

The Sector is now joined with Sector 1. Go into SECTORS MODE and pass your cursor over the Sectors to make sure they all light up yellow. This is a good quick test to see if your Sectors are all closed, but it's not foolproof. WinDEU has a set of ERROR CHECKING tools, which you'll learn about in Chapter 2.7.

Let's shorten the length of Sector 4 by moving the Vertices.

1 Enter VERTICES MODE. Drag Vertex **A** up one unit (32 pixels). Drag Vertex **B** up one unit.



A

B

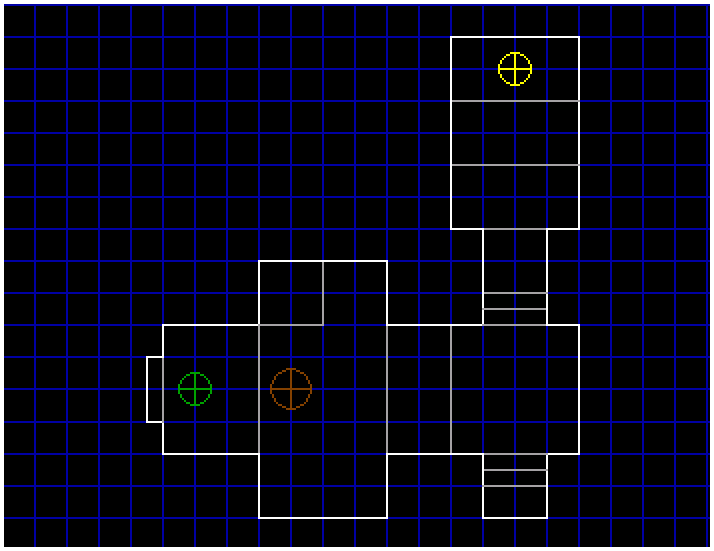
Figure 2.12.
Drag Vertex A and then Vertex B to their new positions.

You'll be using this Sector to make a door in Chapter 2.4. Next, you'll learn about splitting LineDefs and Sectors.

2.2.3 The Split LineDef Function

We're going to turn this rather plain and unimaginative map into a respectable DOOM level over the next few pages. With just a few clicks here and there, we'll fashion this map into a minor masterpiece, and you'll learn more about what WinDEU can do.

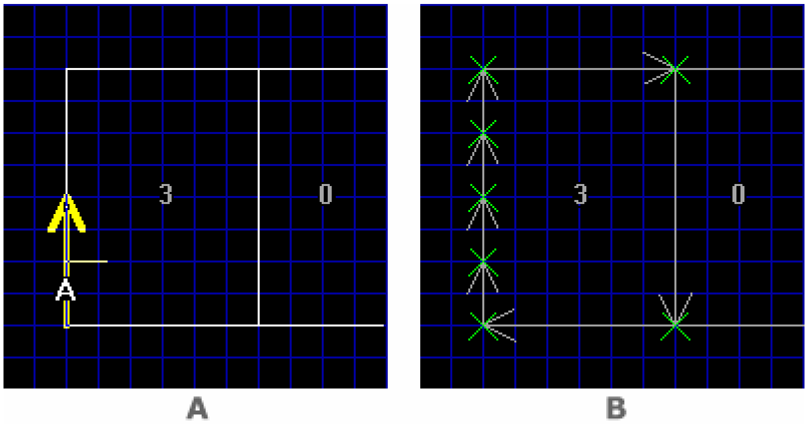
Figure 2.13.
TEST.WAD tutorial map in its finished state.



First, let's add a standard false start door, present in most DOOM levels. We'll do this by splitting a few existing LineDefs. We'll start with LineDef A as shown in Figure 2.14. Let's *zoom in* a little, though, since we'll be doing detail work involving Vertices. (Remember, WinDEU zooms around wherever your mouse cursor is, so place it near the LineDef or PLAYER 1 START.) Press the **+** key till you're at 1/1 or 2/1 scale. You want the grid smaller, so hit **SHIFT-G** to bring the grid scale to 16.

1. Go into LINEDEFS MODE (**L**). Place your cursor over LineDef **A** (it turns yellow) and **right-click**. A small FUNCTION MENU comes up with various actions one can perform on a LineDef. Choose **SPLIT (ADD VERTEX)**.

Figure 2.14.
Right-click on LineDef A and choose SPLIT (ADD VERTEX).



The LineDef has been split neatly with a Vertex in the middle. There are many reasons why one might want to split LineDefs. You may want to add detail around a door, such as metal trim. Or you may want to create an inset in a wall for a light box. Or you may want to create a false door inset, such as you see in many DOOM levels right behind the PLAYER 1 START (to create the illusion that that's the door you've just come through to start the level). We're going to explore all of those possibilities by splitting some more LineDefs and creating a new Sector.

We're going to stay with the LineDefs we just created and split them some more to create a false door inset, so stay in LINEDEFS MODE.

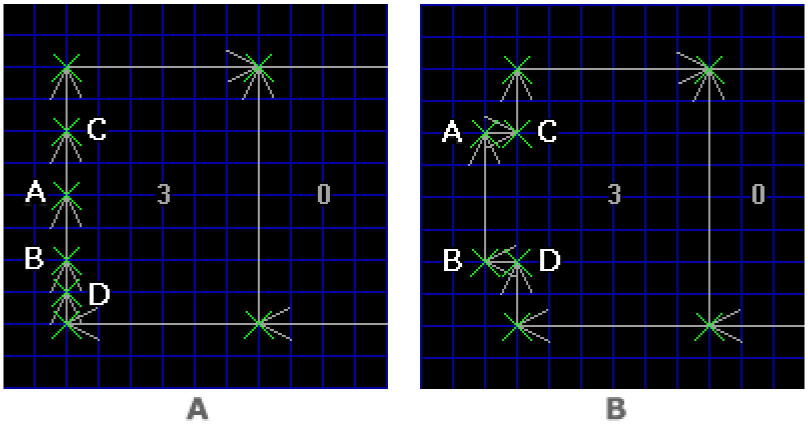
1. Right-click on the upper new LineDef. Choose **SPLIT (ADD VERTEX)**.
2. Split the lower LineDef using the same method (Figure 2.14a).

We have four new LineDefs and three new Vertices (Figure 2.14b). We need one more.

3. Choose one of the new LineDefs (it doesn't matter which, but let's say the lowest) and split it again until we have four Vertices. (You may want to *zoom in* to give you more room to work. Hit the + key to do so.)

Now go into VERTICES MODE to see the new Vertices. These are going to be the points that form an inset false door behind the PLAYER 1 START (see Figure 2.15a.)

Figure 2.15.
Split the LineDefs until you have four Vertices. Arrange the Vertices to form a door inset 16 pixels.



We want the door to be inset 16 pixels or units (or 1 grid unit) and 64 wide.

4. Move two of the center Vertices – Vertex **A** and Vertex **B** in Figure 2.15a – back a grid unit and move Vertices **C** and **D** so that the door is centered in the wall (see Figure 2.15b). (If your map looks like mine, you'd move A over one and up two, B over one, D up one, and leave C where it is.)

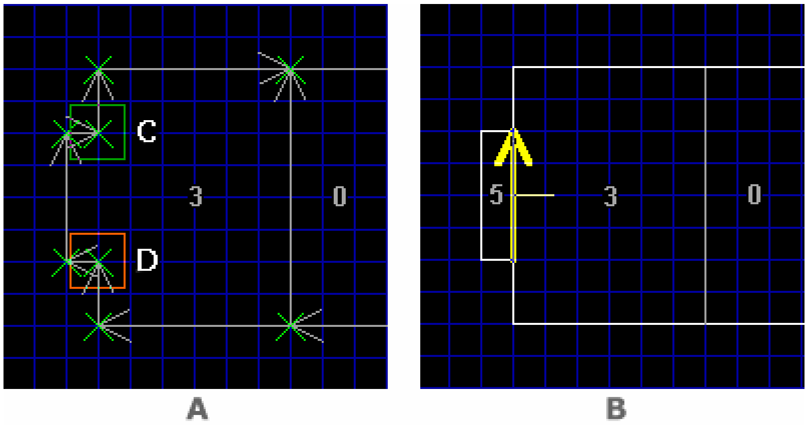
Now, we want the door inset to be a different height than the start Sector. All the Sectors we've created have a height of 128 (the floor is set to **0** and the ceiling to **128**). To do this we first have to make the door inset a separate Sector. We'll have to split the Sector, and we do this by adding a LineDef between Vertices **C** and **D**.

2.2.4 The Split Sector Function

The SPLIT SECTOR function comes in handy for many purposes. It's an easy way to chop a Sector up into units; perhaps for lighting purposes, or to create a set of stairs. Now, we want to split the door Sector because we need it to be a different height than Sector 3, and we'll want to add some lighting effects.

1. Stay in VERTICES MODE. *Mark* the Vertices. First, mark Vertex **C** by pressing the SHIFT key and left-clicking on the Vertex. Now do the same with Vertex **D** (SHIFT and left-click). (Remember, *marked* objects turn green, with the last item marked red.)
2. Right-click on Vertex **D** to invoke the FUNCTION MENU and choose **SPLIT SECTOR**.

Figure 2.16.
Mark Vertices C and D and right-click. Choose Split Sector.

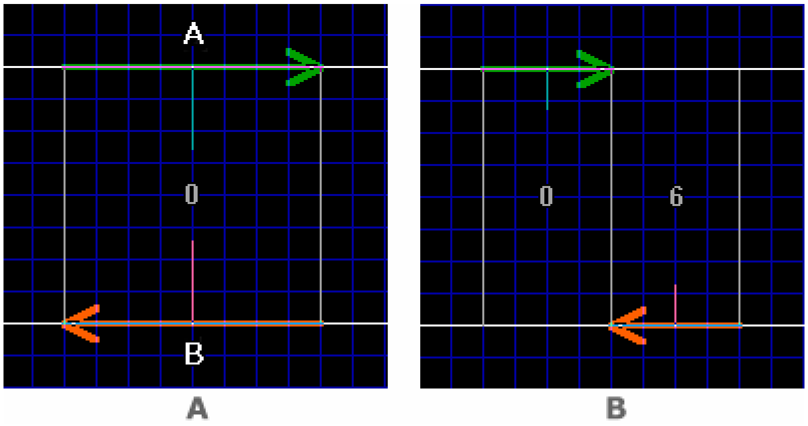


A LineDef has been created between the two Vertices, creating a new Sector! (Splitting a Sector always involves creating a LineDef between two Vertices. We can do this by splitting LineDefs, creating Vertices, and then joining them with a LineDef, as we did just now.

The main purpose of this lesson was to show you how to split LineDefs by adding Vertices. But there's another way to split a Sector with the SPLIT SECTOR menu function.

1. Go into LINEDEFS MODE. Mark LineDefs **A** and **B** (see Figure 2.17a), then right-click on one of them to bring up the menu. Choose **SPLIT (SECTORS)**. You've just split the Sector in two, creating a LineDef between them (Figure 2.17b). Easy as that.

Figure 2.17.
Right-click on a LineDef and choose SPLIT (SECTOR) from the FUNCTION MENU.



2.2.5 The Join Sector Function

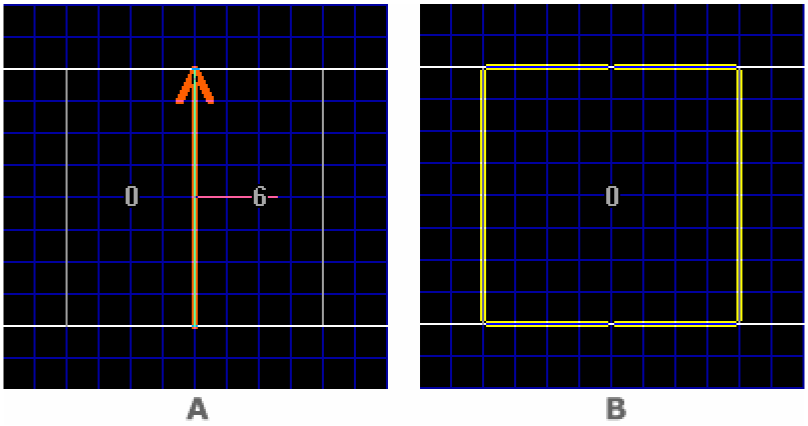
We're going to look at the FUNCTION MENU for joining Sectors. Sometimes you may be creating Sectors to create special light areas, and after a lot of chopping up, you need to get rid of a LineDef or two. You **should not** simply delete a LineDef by hitting the DELETE key. Instead, use the JOIN SECTOR function.

We just split a Sector into two, creating a LineDef between them. We're going to get rid of the LineDef and join the Sectors again into one.

1 Enter LINEDEFS MODE. Mark the new LineDef and right-click on it. The FUNCTION MENU comes up.

2 Choose **DELETE (JOIN SECTORS)**.

Figure 2.18.
Right-click on the new LineDef and choose DELETE (JOIN SECTORS) from the FUNCTION MENU.



The LineDef is gone, and the Sector is joined again. Always use this instead of the DELETE key.

That covers all of the main FUNCTION MENU operations for Sectors. We still need to doctor up the door Sector, and you'll learn how to do that in the next chapter, which deals with the SECTOR, LINEDEF, and THING EDIT dialog boxes.

2.3 Modifying Sector, LineDef, and Thing Properties

The properties for objects are modified through the edit dialog boxes. Texture changes, Sector height and lighting modification, and Thing placement are all done here.

2.3.1 The Sector Edit Dialog Box

Let's change the properties for our door Sector. The properties that can be changed in the SECTOR EDIT dialog box are the SECTOR SPECIALS, CEILING and FLOOR HEIGHTS, the LIGHT LEVEL, and the flats, or the textures that go on the floor and ceiling.

1. Go into SECTORS MODE (S). Pass your cursor over the door Sector (a quick way of making sure your Sectors are valid is by seeing if they light up yellow) and hit ENTER (or right-click and choose **EDIT OBJECT**). The SECTOR EDIT dialog box comes up.
2. We want to make the Sector 64 units high, so change the CEILING HEIGHT from 128 to **64**. We also want to add some drama to the door and make it look like many of the false start doors in DOOM, so let's change the LIGHT LEVEL to **208**. And on top of that, we're going to make the lights blink randomly. So in the SECTOR SPECIAL list box, click on **(1) LIGHT GOES OFF RANDOMLY (NORMALLY ON)**. Click OK.

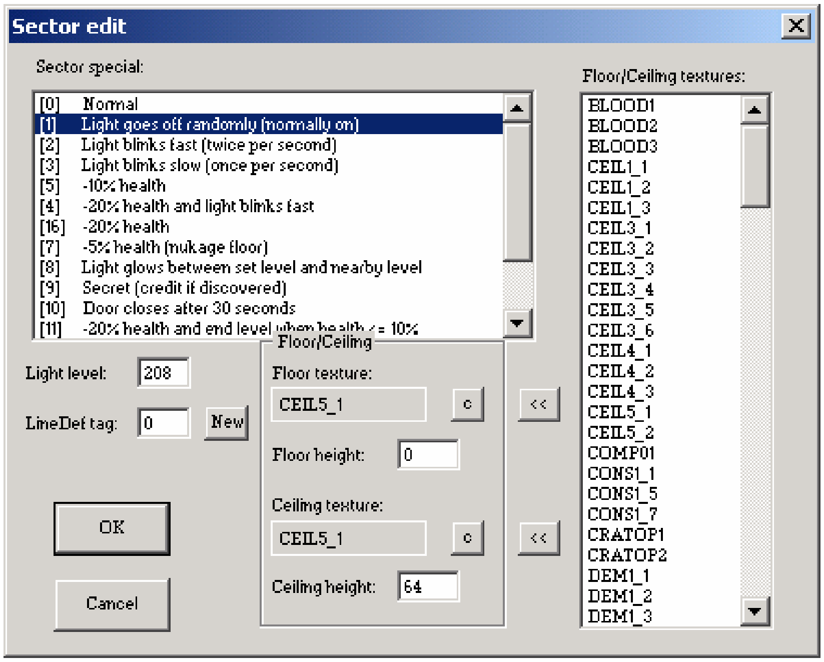


Figure 2.19. Change the CEILING HEIGHT to 64 and LIGHT LEVEL to 208 in the SECTOR EDIT dialog box.

Sector edit

Sector special:

[0] Normal
[1] Light goes off randomly (normally on)
[2] Light blinks fast (twice per second)
[3] Light blinks slow (once per second)
[5] -10% health
[4] -20% health and light blinks fast
[16] -20% health
[7] -5% health (nukage floor)
[8] Light glows between set level and nearby level
[9] Secret (credit if discovered)
[10] Door closes after 30 seconds
[11] -20% health and end level when health <= 10%

Light level: 208

LineDef tag: 0

Floor/Ceiling

Floor texture: CEIL5_1

Floor height: 0

Ceiling texture: CEIL5_1

Ceiling height: 64

Floor/Ceiling textures:

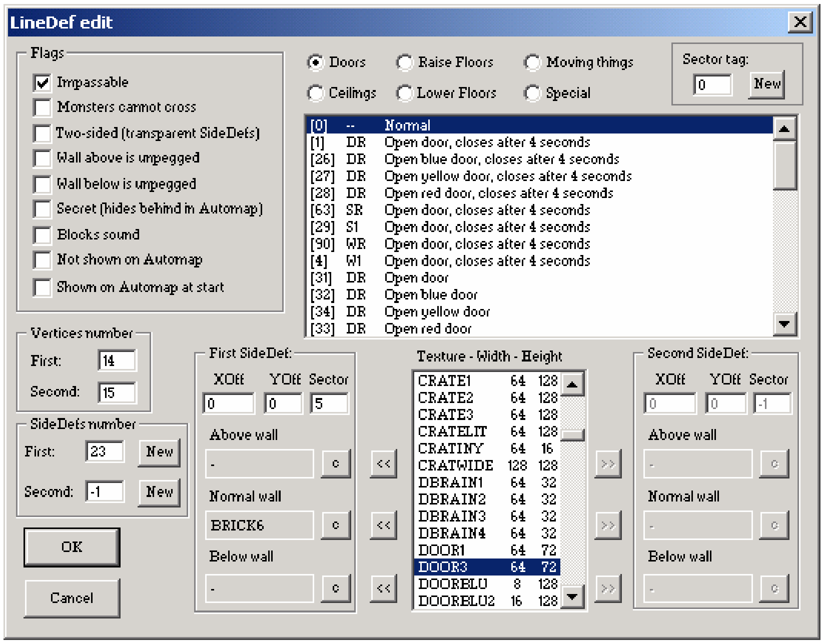
BLOOD1
BLOOD2
BLOOD3
CEIL1_1
CEIL1_2
CEIL1_3
CEIL3_1
CEIL3_2
CEIL3_3
CEIL3_4
CEIL3_5
CEIL3_6
CEIL4_1
CEIL4_2
CEIL4_3
CEIL5_1
CEIL5_2
COMP01
CONS1_1
CONS1_5
CONS1_7
CRATOP1
CRATOP2
DEM1_1
DEM1_2
DEM1_3

Now we want to change some textures to make our new Sector look like a door and add light panels to the inset. Let's do the door texture first.

2.3.2 The LineDef Edit Dialog Box

The LINEDEF EDIT dialog box has everything you need to modify LineDefs. All visible LineDefs have a FIRST SIDEDEF. The "walls" of your Sector are one-sided, meaning that the side facing inside the Sector is visible. This SideDef always needs a texture. The box in Figure 2.20 shows that the FIRST SIDEDEF has a NORMAL WALL texture BRICK6. The list of available textures (with their WIDTH and HEIGHT) is in a scroll list next to it. And next to that is the SECOND SIDEDEF area, which as you can see is blank. Notice that the little arrow boxes are grayed out, preventing transfer of textures. That's because there is no SECOND SIDEDEF. (Note the white text boxes above each SideDef. The Sector box shows what Sector the SideDef faces or belongs to. The FIRST SIDEDEF is Sector 2, the SECOND SIDEDEF shows Sector -1. Sectors start with the number 0, so **-1** indicates *No Sector*.)

Figure 2.20.
Double-click on an object or hit ENTER to invoke the LINEDEF EDIT dialog box.



Flags
☑ Impassable
☐ Monsters cannot cross
☐ Two-sided (transparent SideDefs)
☐ Wall above is unpegged
☐ Wall below is unpegged
☐ Secret (hides behind in Automap)
☐ Blocks sound
☐ Not shown on Automap
☐ Shown on Automap at start

● Doors ○ Raise Floors ○ Moving things
○ Ceilings ○ Lower Floors ○ Special

Sector tag: 0

[0] -- Normal
[1] DR Open door, closes after 4 seconds
[26] DR Open blue door, closes after 4 seconds
[27] DR Open yellow door, closes after 4 seconds
[28] DR Open red door, closes after 4 seconds
[63] SR Open door, closes after 4 seconds
[29] S1 Open door, closes after 4 seconds
[90] WR Open door, closes after 4 seconds
[4] W1 Open door, closes after 4 seconds
[31] DR Open door
[32] DR Open blue door
[34] DR Open yellow door
[33] DR Open red door

Vertices number
First: 14
Second: 15

SideDefs number
First: 23
Second: -1

First SideDef:
XOff: 0 YOff: 0 Sector: 5
Above wall: -
Normal wall: BRICK6
Below wall: -

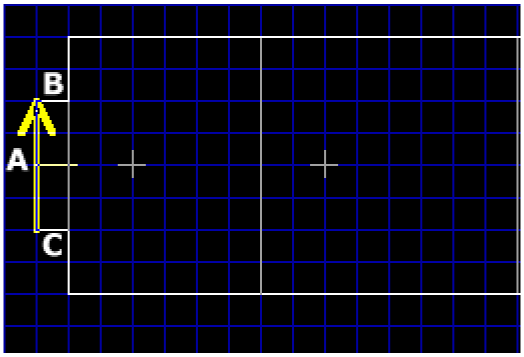
Texture - Width - Height
CRATE1 64 128
CRATE2 64 128
CRATE3 64 128
CRATELIT 64 128
CRATINY 64 16
CRATWIDE 128 128
DBRAIN1 64 32
DBRAIN2 64 32
DBRAIN3 64 32
DBRAIN4 64 32
DOOR1 64 72
DOOR3 64 72
DOORBLU 8 128
DOORBLU2 16 128

Second SideDef:
XOff: 0 YOff: 0 Sector: -1
Above wall: -
Normal wall: -
Below wall: -

You should finish the door Sector now.

1 Go into LINEDEFS MODE (L). Right-click on LineDef **A** (see Figure 2.21) to invoke the LINEDEF EDIT dialog box (see Figure 2.20).

Figure 2.21.
Select LineDef A and hit ENTER (or right-click and choose Edit Object) to bring up the LINEDEF EDIT dialog box.



2 In the texture list scroll down until you find **DOOR3.** Select it (see Figure 2.23).

3 Once you have **DOOR3** selected, click the double-arrow TRANSFER button << to the left to transfer the texture to the NORMAL TEXTURE area of the FIRST SIDEDEF. Click OK.

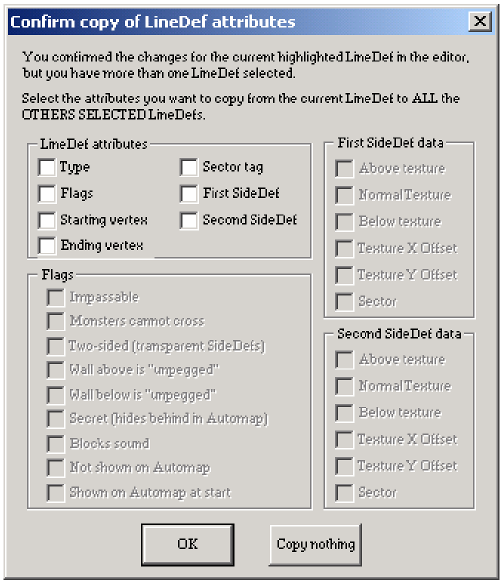
> **NOTE** To jump quickly to a texture or flat, click on the a texture in the list and type the first letter - in this instance D - and WinDEU will take you to the first texture that begins with that letter. If you type the second letter - O - you'll jump to the first texture name that begins with *DO*. You can also type the third letter, and so on. Usually you only need the first two to get within a short distance of the texture you want instead of scrolling down the list.

Note that DOOR3 is one of the three smallest door textures in DOOM (the others being DOOR1 and EXITDOOR), and that they have an irregular height of 72 (instead of the expected 64). You should always make your Sector heights match your door texture height. The other doors in DOOM are 128 high (with one exception), and putting these textures on a LineDef in a Sector with a height less than 128 always looks sloppy.

Now we want to add light panels to the door inset.

1 Mark both LineDef **B** and **C** (in Figure 2.21) then hit ENTER. The LINEDEF EDIT dialog box comes up. Choose texture **LITE5**, click the TRANSFER double-arrow button for the NORMAL TEXTURE on the FIRST SIDEDEF. Click OK.

2 Another dialog box – CONFIRM COPY OF LINEDEF ATTRIBUTES – comes up, asking you to confirm that you've chosen to change more than one object. It shows the attributes that are being changed – in this case, the textures on the FIRST SIDEDEFS. Click OK.



Confirm copy of LineDef attributes

You confirmed the changes for the current highlighted LineDef in the editor, but you have more than one LineDef selected.

Select the attributes you want to copy from the current LineDef to ALL the OTHERS SELECTED LineDefs.

LineDef attributes
☐ Type
☐ Sector tag
☐ Flags
☐ First SideDef
☐ Starting vertex
☐ Second SideDef
☐ Ending vertex

Flags
☐ Impassable
☐ Monsters cannot cross
☐ Two-sided (transparent SideDefs)
☐ Wall above is "unpegged"
☐ Wall below is "unpegged"
☐ Secret (hides behind in Automap)
☐ Blocks sound
☐ Not shown on Automap
☐ Shown on Automap at start

First SideDef data
☐ Above texture
☐ Normal Texture
☐ Below texture
☐ Texture X Offset
☐ Texture Y Offset
☐ Sector

Second SideDef data
☐ Above texture
☐ Normal Texture
☐ Below texture
☐ Texture X Offset
☐ Texture Y Offset
☐ Sector

Figure 2.22.
CONFIRM COPY OF LINEDEF ATTRIBUTES dialog box.

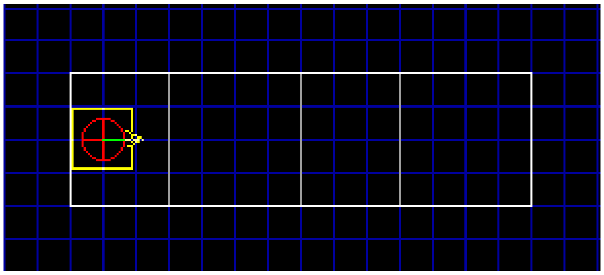
You're finished. You'll find out more about LineDefs when we create a door. For now, though, let's have a look at the EDITING THING dialog box. A level isn't any fun without any monsters or weapons in it.

2.3.3 The Editing Thing Dialog Box

Our map is getting a little unwieldy, even though it's still very small. Yet, you're probably anxious to see what it looks like, if this is your first time making a level. Before you can play it we need to do at least one more thing: add a **PLAYER 1 START**. So let's switch to THING MODE (press **T**).

1 Place your mouse cursor in the middle of Sector 3 and press **INS**.

Figure 2.23.
Add a Thing by pressing INS.

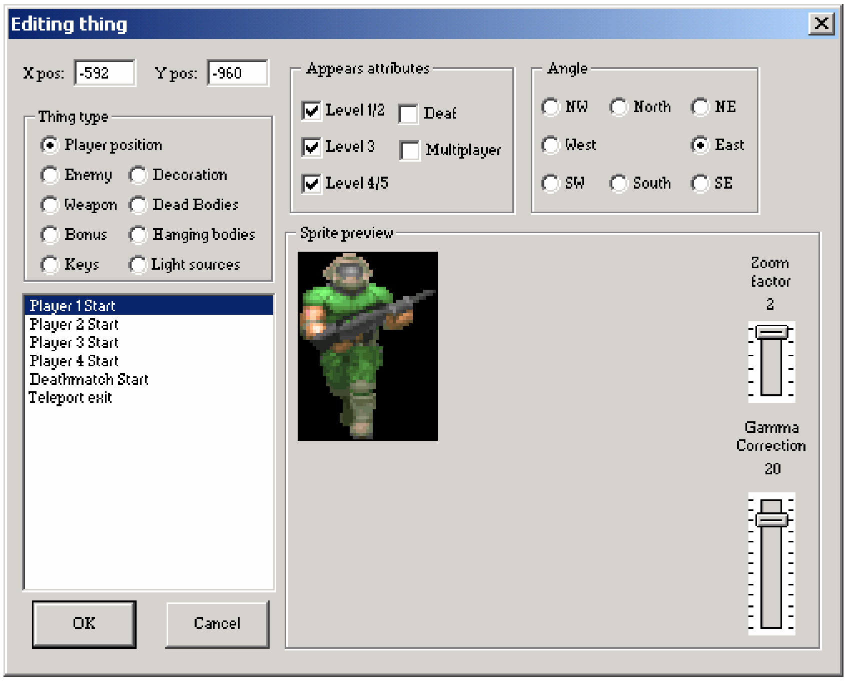


You should see a red circle surrounded by a yellow box with an arrow facing east (or right). This is the default Thing that WinDEU inserts the first time: a TROOPER. We need to change it to the **PLAYER 1 START**. This marks the point of entry for the player when you start your level. The arrow indicates the direction the player will be facing when he enters the level.

2 Press **ENTER** so that we can edit the Thing's properties.

The EDITING THING dialog box comes up with all the different Things you can choose, and their properties.

Figure 2.24.
The EDITING THING dialog box.



Editing thing

Xpos: -592 Ypos: -960

Thing type
● Player position
○ Enemy ○ Decoration
○ Weapon ○ Dead Bodies
○ Bonus ○ Hanging bodies
○ Keys ○ Light sources

Player 1 Start
Player 2 Start
Player 3 Start
Player 4 Start
Deathmatch Start
Teleport exit

Appears attributes
☑ Level 1/2 ☐ Deaf
☑ Level 3 ☐ Multiplayer
☑ Level 4/5

Angle
○ NW ○ North ○ NE
○ West ● East
○ SW ○ South ○ SE

Sprite preview

Zoom factor 2

Gamma Correction 20

3 Under THING TYPE, select the option button next to PLAYER POSITION and then click on **PLAYER 1 START** in the list below. Click **OK**.

(You notice that WinDEU shows you images of the different Things, which is very handy.)

No level is any fun without a few monsters, so let's add at least one.

1 Still in THINGS MODE, place your cursor in the furthest Sector to the East or right and press **INS**. Since the last Thing you edited or placed was a PLAYER 1 START, that's the default Thing that will be inserted.

2 Click the option box next to ENEMY under THING TYPE. Let's choose an **IMP** from the list by clicking on it.

Note the **APPEARS ATTRIBUTES** area. The first three LEVEL check boxes are always chosen by default, which means that Things will always appear in all five of DOOM's Skill Levels. Here's what the choices mean:

LEVEL 1/2	The Thing will appear in **SKILL 1: I'M TOO YOUNG TO DIE** and **SKILL 2: HEY, NOT TOO ROUGH.**
LEVEL 3	The Thing will appear in **SKILL 3: HURT ME PLENTY**.
LEVEL 4/5	The Thing will appear in **SKILL 4: ULTRA-VIOLENCE** and **SKILL 5: NIGHTMARE**.
DEAF	The Thing will not hear gunshots and will only attack when it sees the player.
MULTIPLAYER	The Thing will appear only in **COOPERATIVE** or **DEATHMATCH** modes.

If you decide to distribute your level on the Internet for others to download and play, you'll want to make the level as diverse as possible, implementing **Difficulty Levels** (for instance, a seasoned DOOM player doesn't expect a Cyberdemon to show up in a level if he's chosen **Skill 1: I'm Too Young to Die**. So you may want to tag the Cyberdemon as LEVEL 4/5 only, and place a Baron of Hell in the same room tagged LEVEL 1/2 and LEVEL 3. Likewise, you may want a super-weapon such as the BFG to appear only under the most difficult skill levels and substitute a Super-Shotgun for easier skill levels. You can imagine now just how involved complex and thoughtful level editing can get.)

The default **ANGLE** for a thing is *east*. You will definitely want to change this for your monsters so that they're facing the player when he enters a room (or maybe not. Maybe the Imp is guarding a door on the other side of the room, and you want him to have his back to the player as a lure).

At any rate, let's change the Imp's ANGLE to west.

3 Select the option box next to **WEST** in the ANGLE area. (ANGLE has no effect on any THING TYPES except PLAYER POSITIONS and ENEMIES. DECORATIONS, WEAPONS, BONUSES and the rest have their sprites rotated to the player's viewpoint by the DOOM engine.) Click OK.

We need a better weapon than the default pistol to kill our Imp with, so let's add a Shotgun.

1 Place your mouse cursor in front of the PLAYER 1 START a unit or two and press INS. Now press ENTER to bring up the EDITING THING dialog box.

2 Click on **WEAPONS** and then **SHOTGUN**. Click **OK**.

Next, let's learn how to make a door.

2.4 Making a Door

Now we'll start doing some fun stuff. Making a door is easy, but it's easy to get it wrong. WinDEU has a function to make a Sector into a door automatically, but we'll save that for later. You're going to do it the hard way first, so that you understand what how a door works.

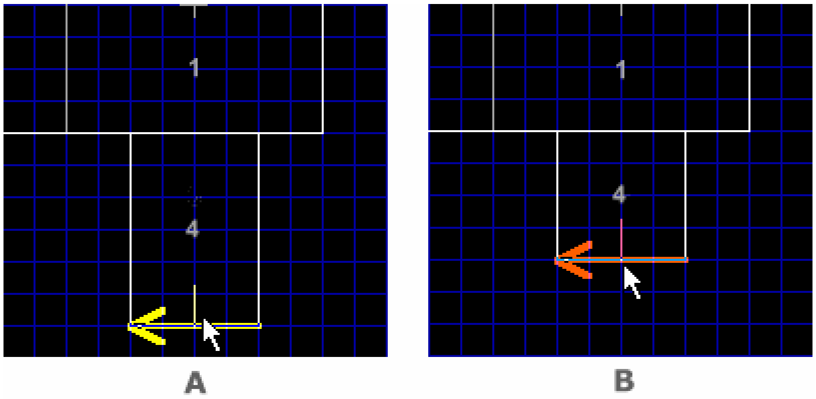
The type of door you're going to make is called a MANUAL DOOR. This door is activated by the player using the spacebar (or *use* key). It's a special category of door that requires a **LINEDEF ACTION TYPE**, but not a TAG number. ACTION TYPES are functions assigned to LineDefs that tell a Sector what to do. Sectors are the only moving structures in DOOM, and they can only move up or down. LINEDEF ACTION TYPES tell the Sector what action to perform. In order to open the door, the LineDef itself must be activated by the player. Normally, the LineDef and the Sector are assigned an identical, but arbitrary TAG number. If the LineDef has TAG number 4, then it looks for a Sector with the same TAG number. When the LineDef is activated, all Sectors with a TAG number of 4 will perform the assigned function.

The type of MANUAL DOOR you'll be making does not require a TAG number. The LineDef assigned this ACTION TYPE will act upon the Sector that its LEFT or SECOND SIDEDEF belongs too. There are ten of these kinds of doors, and they all have the *trigger* designation **DR** – which stands for ***D****oor* ***R****epeatable*. This type of door and its ACTION TYPE are known as a LOCAL ACTION.

Let's work with Sector 4 (the hallway that splits off from the bottom of the map). Let's shorten that so that it is 4 grid units deep (64 pixels) as in Figure 2.25. (Make sure your grid is set to 16.)

1 Enter LINEDEFS MODE. Position the cursor over the bottom LineDef and drag it up to the position shown in Figure 2.25b. (Remember, to drag an object, left-click on it, hold the button down, and move the cursor to drag the object to its new position.)

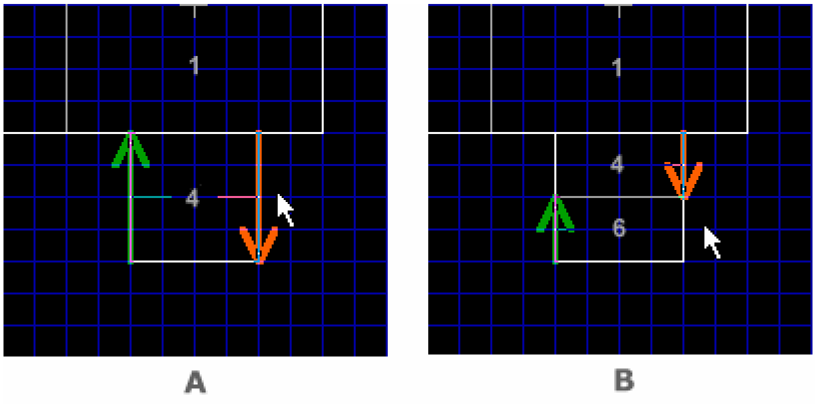
Figure 2.25.
Drag LineDef A to the position shown.



You're going to split this Sector into three smaller Sectors, and one of them will be the door. (The door doesn't go anywhere yet, I know.) You may want to zoom in a little, since the space you're working in is tight.

1 Mark LineDefs **B** and **C** as shown in Figure 2.26a, right-click on one to bring up the FUNCTION MENU, and choose **SPLIT (SECTORS)**. You should have two Sectors now (Figure 2.26b).

Figure 2.26.
Split Sector 3 into two.

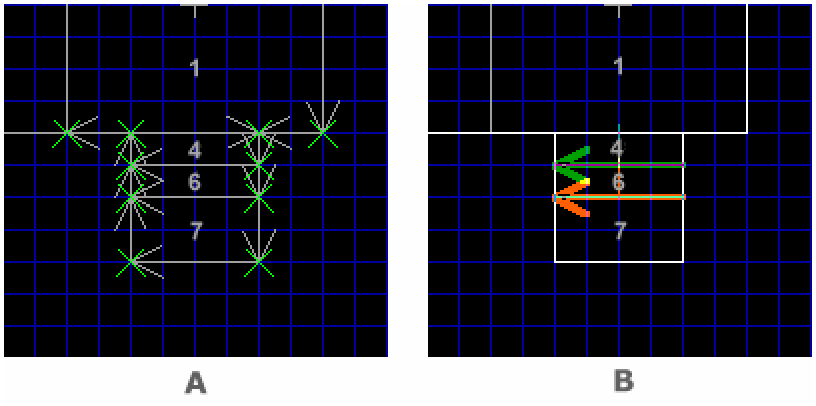


Now let's do it again.

1 Mark the two new upper LineDefs, right-click to bring up the FUNCTION MENU. Choose SPLIT (SECTOR). There should now be three Sectors.

The Vertices should be arranged as shown in Figure 2.27a, so that the door is 16 pixels (one grid unit) wide (Figure 2.27b).

Figure 2.27.
The door Sector should be 16 pixels wide.



Depending on which order you split your LineDefs, your two new LineDefs should have their vectors (the little stick in the middle) pointing up and down. When making a door, it is important that the LineDefs' vectors face *out* of the door Sector. That is, the vector indicates the FIRST SIDEDEF, and the FIRST SIDEDEF must face out of the door Sector.

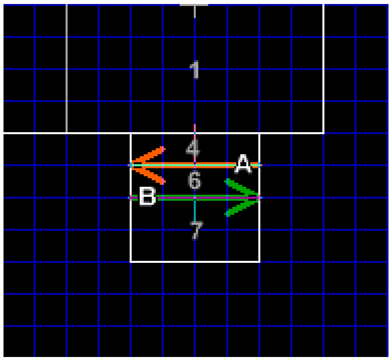
MANUAL DOORS were mentioned earlier as having an ACTION TYPE known as a LOCAL ACTION. LOCAL ACTIONS must be activated from the FIRST SIDEDEF. So you want the FIRST SIDEDEF pointing out of its Sector. Figure 2.27b shows that the lower LineDef has its FIRST SIDEDEF facing *into* its Sector. You'll need to *flip* the LineDef.

1 Right-click on the LineDef and choose **FLIP LINEDEF** from the FUNCTION MENU. The vector should be facing downwards now.

Now we can make this Sector a door.

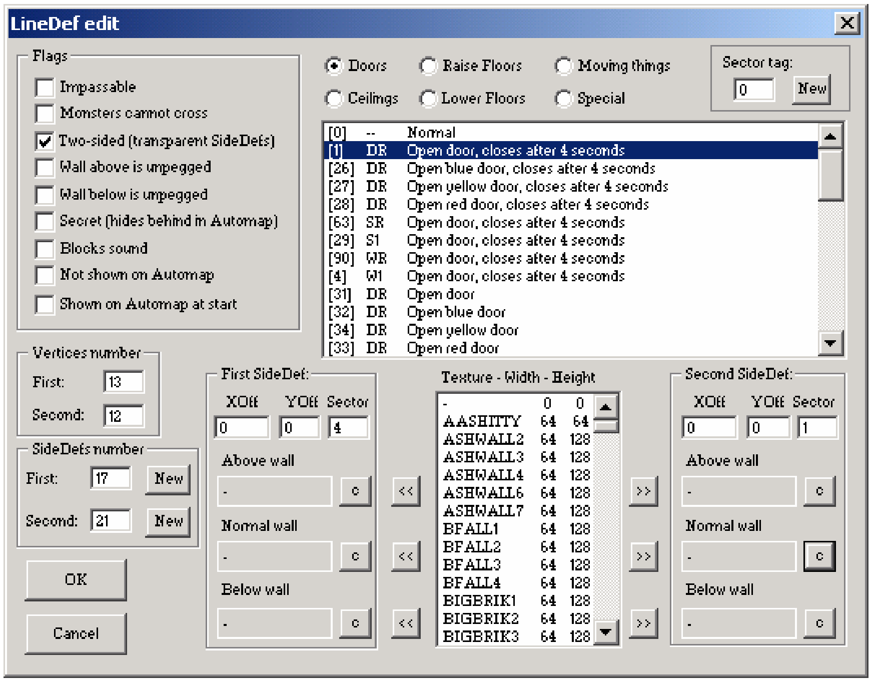
1 Mark LineDefs **A** and **B** and press ENTER. The LINEDEF EDIT dialog box comes up.

Figure 2.28.
Mark LineDefs A and B and hit ENTER.



2 Click on **[1] DR OPEN DOOR, CLOSES AFTER 4 SECONDS.** Click OK. The CONFIRM COPY dialog box comes up. Click OK.

Figure 2.29.
In the LINEDEF EDIT box, click on [1] DR OPEN DOOR, CLOSES AFTER 4 SECONDS.



Flags:
☐ Impassable
☐ Monsters cannot cross
☑ Two-sided (transparent SideDefs)
☐ Wall above is unpegged
☐ Wall below is unpegged
☐ Secret (hides behind in Automap)
☐ Blocks sound
☐ Not shown on Automap
☐ Shown on Automap at start

● Doors ○ Raise Floors ○ Moving things
○ Ceilings ○ Lower Floors ○ Special

Sector tag: 0

[0]	--	Normal
[1]	DR	Open door, closes after 4 seconds
[26]	DR	Open blue door, closes after 4 seconds
[27]	DR	Open yellow door, closes after 4 seconds
[28]	DR	Open red door, closes after 4 seconds
[63]	SR	Open door, closes after 4 seconds
[29]	S1	Open door, closes after 4 seconds
[90]	WR	Open door, closes after 4 seconds
[4]	W1	Open door, closes after 4 seconds
[31]	DR	Open door
[32]	DR	Open blue door
[34]	DR	Open yellow door
[33]	DR	Open red door

Vertices number: First: 13 Second: 12
SideDefs number: First: 17 Second: 21

First SideDef: XOff 0 YOff 0 Sector 4
Above wall: -
Normal wall: -
Below wall: -

Texture	Width	Height
-	0	0
AASHITTY	64	64
ASHWALL2	64	128
ASHWALL3	64	128
ASHWALL4	64	128
ASHWALL6	64	128
ASHWALL7	64	128
BFALL1	64	128
BFALL2	64	128
BFALL3	64	128
BFALL4	64	128
BIGBRIK1	64	128
BIGBRIK2	64	128
BIGBRIK3	64	128

Second SideDef: XOff 0 YOff 0 Sector 1
Above wall: -
Normal wall: -
Below wall: -

Doors must have the same floor and ceiling height. And that height should usually match the Sector they're being accessed from. That is, our map so far has a floor height of 0 and a ceiling of 128. The door Sector needs to have both the **floor height** and the **ceiling height** set at **0**. That's because a door is a moving ceiling that is in the ***down*** position.

1 Enter SECTORS MODE. Select the door Sector (6) and hit ENTER to bring up the SECTOR EDIT dialog box.

2 Set the FLOOR and CEILING HEIGHTS to **0**. Click OK.

You've created a door! The LineDefs should now show green, which WinDEU uses to indicate LineDefs with assigned ACTION TYPES.

Since a door is a moving Sector in the down position, the FIRST SIDEDEFS will need *upper textures*. In WinDEU, they're shown as ABOVE WALL in the FIRST and SECOND SIDEDEF areas of the LINEDEF EDIT dialog box.

Doors have unusual attributes. *DOOM draws normal textures from the* ***top down***. The door Sector's ceiling meets the floor in the down position. When the door rises, the ceiling of the door Sector rises, and the textures to the side of it will appear to move up with the door. This looks goofy (and I'm sure you've seen this happen in some amateur levels). To fix this, the sides of the doors - the door tracks - need to have the *lower textures* **unpegged**. Unpegging the lower textures causes them to be drawn from the *floor up*, instead of from the ceiling down.

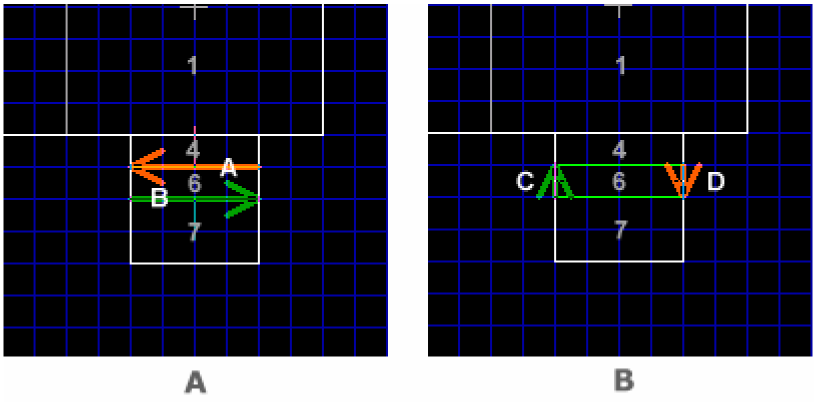
But first, you need to put textures where they're needed.

1. Enter LINEDEFS MODE. Mark LineDefs **A** and **B** and hit ENTER. In the LINEDEF EDIT dialog box, select **SW1LION** in the texture list. (Remember, if you double-click on a texture name, the texture viewer will come up.) Click on the arrow button and transfer the texture to **ABOVE WALL** of the **FIRST SIDEDEF**. Click OK.

2. The CONFIRM COPY dialog box comes up. Click OK.

3. Press **C** to clear the marks and mark the LineDefs **C** and **D** on the side of the door Sector (see Figure 2.30). Press ENTER to bring up the LINEDEF EDIT dialog box.

Figure 2.30.
Apply textures to the front and side SideDefs.



4. In the LINEDEF EDIT box, choose the texture **DOORTRAK** and transfer it to the **NORMAL WALL** box of the **FIRST SIDEDEF**.

5. In the FLAGS area on the left, select the check box next to **WALL BELOW IS UNPEGGED**. This will keep the textures from moving. Click OK.

Your door is now complete!

Now, this was just your plain vanilla DOOM door. Doors can be made to operate from *switches*, close or far away. Or they can be *triggered* by the player walking over an invisible LineDef, called a *trigger*. These kinds of doors are called REMOTE DOORS, and their ACTION TYPES are called REMOTE ACTIONS.

We'll be making a remote door. But first, let's look at some of these different LINEDEF ACTION TYPES, and we'll explain what some of the functions mean.

2.4.1 LineDef Action Types

LINEDEF ACTION TYPES (also called FUNCTION TYPES) control various effects like doors opening, ceilings and lifts lowering and raising, etc. There are 143 different types (not counting the many more created for DOOM ports such as ZDOOM and JDOOM), many of them performing similar functions but at different speeds or height increments, or *activated* in a different manner.

A LINEDEF ACTION TYPE is assigned to a LineDef and the LineDef is given a **TAG** number. Any Sector given the same TAG number will perform the function dictated by the ACTION TYPE. TAG numbers are not ACTION TYPES, nor are they the LineDef or Sector numbers. They're an arbitrary identification number assigned by the level designer.

LINEDEF ACTION TYPES may be *activated* by the player (or sometimes a monster). The player may activate it either by *using* a ***s**witch* (pressing the spacebar or *use* key), by ***w**alking* over it, or by ***g**unfire* – shooting at it. The *manual **d**oors* are a special type of their own category.

A LINEDEF ACTION TYPE is either *repeatable* or *non-repeatable*. In WinDEU, the ACTIVATION TYPES are listed in the LINEDEF EDIT dialog box:

Trigger	Activator	Repeatability	Category	Type
S1	Switch	Once	Switch	Local
SR	Switch	Repeatable	Button	Local
W1	Walkover	Once	Trigger	Remote
WR	Walkover	Repeatable	Trigger	Remote
G1	Gunfire	Once	Impact	Local
GR	Gunfire	Repeatable	Impact	Local
D1	Door (no tag required)	Once	Manual	Local
DR	Door (no tag required)	Repeatable	Manual	Local

A **TRIGGER** is the condition that causes the function to be activated. It's the acronymic symbol for ACTIVATOR (**S**, **W**, **G**, **D**) and REPEATABILITY (**1** and **R**). An **ACTIVATOR** is the event that initiates the LINEDEF ACTION TYPE. **REPEATABILITY** is either *once* or *more than once*. **TYPE** indicates which of two broad functions the LINEDEF ACTION TYPE is: *local* or *remote*. And **CATEGORY** is the term generally accepted and used by all enlightened designers and coders.

Trigger is also widely used to indicate a two-sided transparent LineDef laid out across a Sector to act as a "trip-wire" that – when crossed – initiates the ACTION TYPE assigned to it in whatever Sector has been paired with its TAG number. *Trigger* in this context always refers to a *remote action*. S*witch* always refers to a *local action*, even though it may activate a Sector far away.

NOTE: There are only two types of LINEDEF ACTION TYPES: ***local actions*** (*switch*) and ***remote actions*** (*trigger*). These are sometimes referred to by others as *switch-activated* actions and *pressure-activated* actions, but not in this manual.

2.4.1.1 Local and Remote Actions

For ***local actions***, the LineDef you're activating is usually part of the Sector performing the action. That is, its SECOND SIDEDEF – or LEFT SIDE – is shared by the Sector. This is so because all *local actions* must be activated on the FIRST SIDEDEF – or RIGHT side – of the LineDef. A *local action* is always activated by the spacebar (or default *use* key): therefore, it's usually a switch, button, or manual activator. A switch may invoke an action in a Sector far away, so this is the exception where the LineDef may not actually be part of the moving Sector; but the switch is still a *local action*, because it is the *activator*, and therefore must be activated from the FIRST SIDEDEF.

Remote actions – such as *trigger*S (LineDefs that are *walkover*) – may be crossed from either side to activate. This applies to any mover activated by a two-sided *walkover* LineDef. Therefore, just as doors can be *local* or *remote*, lifts can be activated locally – from the FIRST SIDEDEF, or RIGHT SIDE – or remotely – walkover from either side, even if the LineDef is part of the Sector being activated. (The one notable exception to this rule is a **TELEPORTER**. A TELEPORTER **must** be entered from the FIRST SIDEDEF, or RIGHT SIDE. See Chapter 2.6 for more information.)

All actions, *local* and *remote*, require a shared TAG association between the LineDef and the Sector (the *trigger* and the *mover*) – but for one more notable exception: MANUAL DOORS with the trigger **DR** or **D1**.

For instance, the first door we made was a MANUAL (or *local*) DOOR which did not require a TAG. This is a special condition used only for this type of door, indicated by its trigger **DR** or **D1**. It's easy to confuse a MANUAL DOOR with the way a lift works, because in both cases you seem to be activating the Sector itself. That's because these are *local actions*: the LineDef you're activating is part of the Sector (its SECOND SIDEDEF – or left side – is shared by the Sector performing the action). Many lifts require you to stand right up against them and press the spacebar to make them lower. These lifts still need a TAG number. It just so happens that this is a *local action* – so it appears that it operates in the same manner as a MANUAL DOOR.

NOTE: All *local actions* must be activated on the FIRST SIDEDEF – or RIGHT SIDE – of the LineDef.

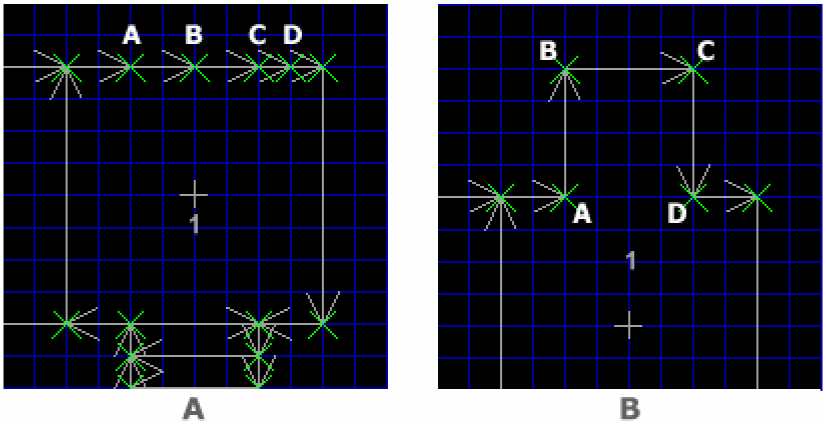
Remote actions using *triggers* are not limited to doors and lifts. One of the things that made DOOM so unforgettable (and terrifying) was walking down a hallway and crossing a transparent trigger that opened a secret door or room that suddenly gushed forth a horde of monsters you weren't expecting. Using this type of ambush is always guaranteed to get a frenzied reaction from an unwitting player.

A complete compendium of the LINEDEF ACTION TYPES, sorted by function and by number, is provided in Appendix C: LineDef Action Type Reference. This is a must-have reference for the level designer.

2.4.2 Remote Doors

To make a REMOTE DOOR - one opened with a *switch* or an invisible *trigger* - let's go back to our map and make some adjustments. We need to add a Sector, and the quickest way is to split some LineDefs and add some Vertices.

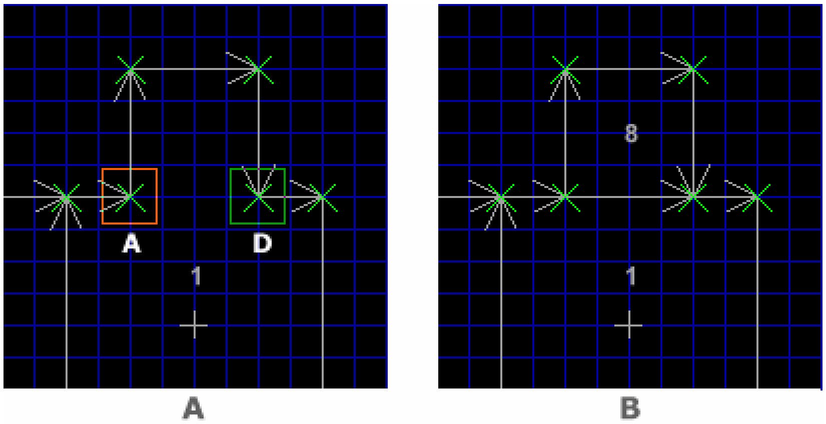
Figure 2.31.
Split the LineDef to get four Vertices and then make a 64x64 room.



Let's start with the LineDef opposite our door.

1. Enter LINEDEFS MODE. Right-click on the LineDef and choose SPLIT (ADD VERTEX) four times till you get four Vertices as in Figure 2.31a.
2. Position the Vertices to form a room 64x64, like the one opposite (see Figure 2.31b).

Figure 2.32.
Mark Vertex D first, then Vertex A.



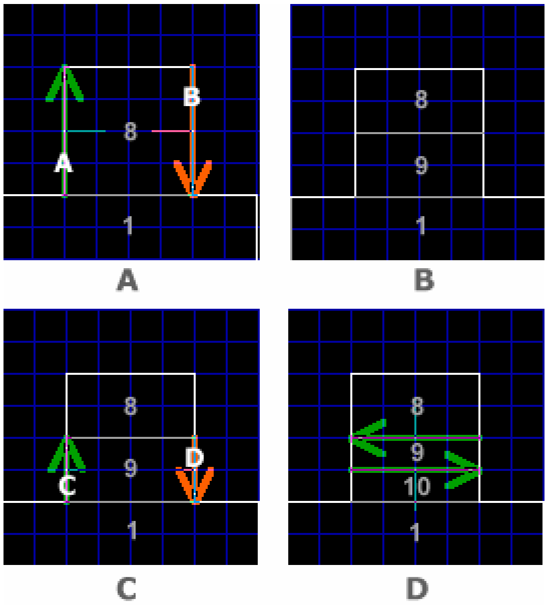
3. Mark first Vertex **D,** and then Vertex **A**. Right-click to bring up the FUNCTION MENU and choose SPLIT SECTOR.

You've split the Sector and added a new Sector (8). Now we need to split this Sector to form our new door.

1. Enter LINEDEFS MODE. Mark LineDefs **A** and **B** (Figure 2.33a), right-click and choose SPLIT (SECTORS) from the FUNCTION MENU.

2. Mark LineDefs **C** and **D** (Figure 2.33c). Right-click for the FUNCTION MENU, choose SPLIT (SECTORS).

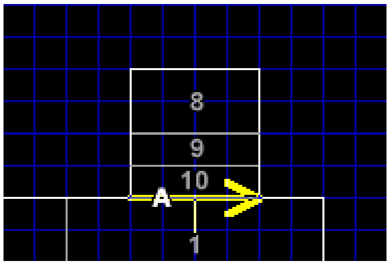
Figure 2.33.
Split the LineDefs to get three Sectors. Sector 9 will be the door. The LineDefs marked in green have their FIRST SIDEDEFS facing *out*.



In LINEDEFS MODE, make sure your vectors are facing the right way. The top LineDef vector should be facing north, the lower facing south (Figure 2.33d). If they are not, fix them with the FLIP LINEDEF function.

Now, we're going to make Sector 9 a REMOTE DOOR, and we'll use LineDef **A**, right in front of it (see Figure 2.34), for the *trigger*.

Figure 2.34.
Select LineDef A for the *trigger*.



We need to note the number of our Sector, which is **Sector 9**. (Select it with the mouse and look at the INFO WINDOW in the bottom left of the screen.)

1. Right-click on LineDef **A** and press ENTER. In the LINEDEF EDIT box, choose **[90] WR OPEN DOOR, CLOSES AFTER 4 SECONDS**.

2. In the **SECTOR TAG** area in the upper right hand corner, click on **NEW**. This will always choose the *next unused Tag Number*. Since we haven't used any yet, this is **SECTOR TAG 1**. Click OK.

Note that the LineDef *trigger* has turned purple. WinDEU shows all *triggers* as purple, and all tagged Sectors as green.

NOTE: If you pass your mouse over a *trigger*, the affected Sector will light up. This is how you know which trigger is tagged with which Sector.

We've assigned the ACTION TYPE to the LineDef to open a door when someone walks over it, with a SECTOR TAG of 1. Now we have to assign that tag number to the Sector we want to have perform the function, which will be Sector 9.

1 Go into SECTORS MODE. Select Sector 9, and press ENTER. The SECTOR EDIT dialog box comes up. Right above the OK button is the **LINEDEF TAG** box. Enter number **1** (don't click NEW, or it will choose number 2, because we already used this function in the LINEDEF EDIT box).

Since this is a door that opens by rising, we need to make it start in the *down* position.

2 Enter a CEILING HEIGHT of **0**. Click OK.

And now to finish up. We need to assign textures to the door and door tracks.

1 Enter LINEDEFS MODE. Mark both of the facing LineDefs of the door and hit ENTER. Choose **SW1LION** for the **UPPER TEXTURES**, which are shown as **ABOVE WALL** for the FIRST SIDEDEF. Click OK.

2 Mark the door tracks on the side of the door, hit ENTER and choose **DOORTRAK**. Transfer the texture to the **NORMAL WALL**.

3 And remember to check **WALL BELOW IS UNPEGGED** in the **FLAGS** area. Hit OK.

That's it! You can use remote triggers for all kinds of actions: crushing ceilings, rising floors and stairs, and lifts.

Now would be a good time to save your map. (Remember to do this often and always make backups.) You may want to test your map to see if the doors work. In the next chapter, we'll modify the map a bit and then make a lift.

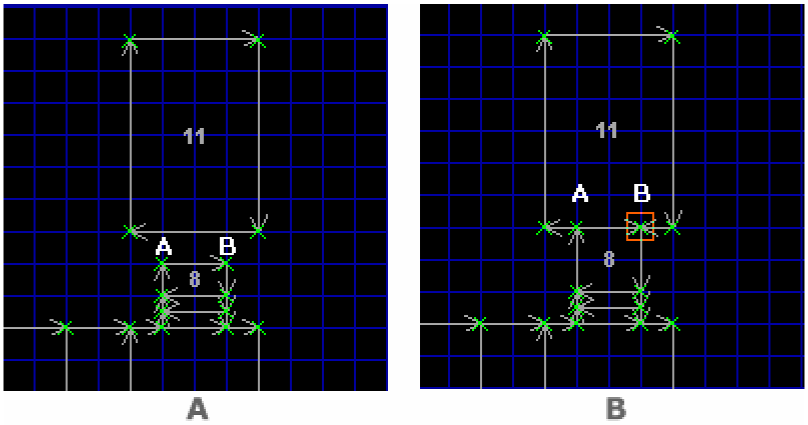
2.5 Making a Lift

WinDEU has a menu function for making lifts automatically, which we'll explore at the end of this chapter. So that you understand how lifts work, we're going to make one the old-fashioned way. We'll need to add a couple rooms to the level.

Let's add a room off the area where we just made the remote door.

1. Enter VERTICES MODE. Click the INSERT RECTANGLE button in the TOOLBAR. Enter a WIDTH of **128** and a HEIGHT of **192**. Click OK. Place the cursor well above the door Sector (see Figure 2.35) and left-click to create the Sector.
2. Drag Vertex **A** and Vertex **B** to the positions shown in Figure 2.35b to join Sector 8 with Sector 11.

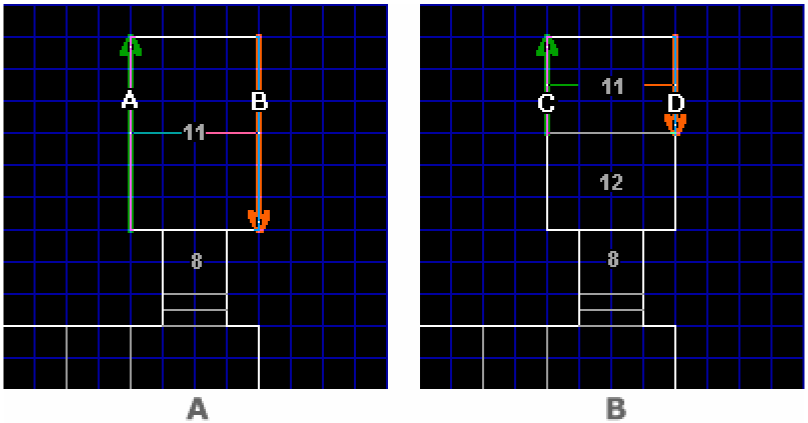
Figure 2.35.
Drag Vertices A and B to join Sector 11.



Now we need to split Sector 11 a couple of times until we have three Sectors.

3. Enter LINEDEFS MODE. Mark LineDef **A** and LineDef **B** as shown in Figure 2.36a. Right-click to bring up the FUNCTION MENU and choose SPLIT (SECTORS).
4. Mark LineDef **C** and LineDef **D** (Figure 2.36b) and right-click. Choose SPLIT (SECTORS).

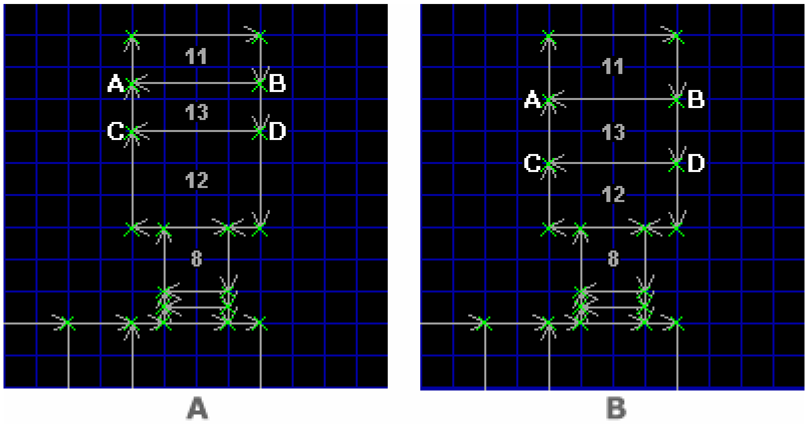
Figure 2.36.
Split Sector 11 to create two Sectors.



We now have three Sectors. Let's adjust the Vertices so that each Sector is 128 pixels wide and 64 high.

5 Enter VERTICES MODE. Move Vertices **A**, **B**, **C**, and **D** to the positions shown in Figure 2.37b.

Figure 2.37.
Adjust the Vertices so that each Sector is 128x64.



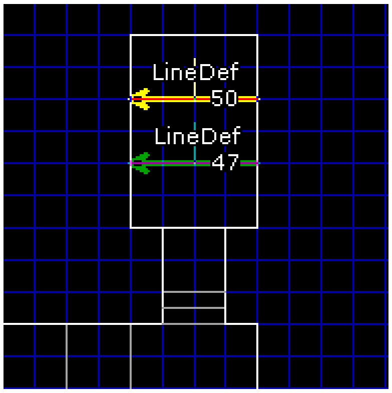
Now we'll transform Sector 13 into a lift. We'll need to change the floor height for Sector 11, add textures here and there, and assign a LINEDEF ACTION TYPE for lifts to both LineDefs.

1 Enter SECTORS MODE. Double-click on **Sector 11** to bring up the SECTOR EDIT dialog box. Change the FLOOR HEIGHT to **-128** and the CEILING HEIGHT to **0**. Click OK.

Changing the Sector heights has exposed upper and lower textures on LineDef 50. Let's fix that and also give the lift a STEP flat for the floor.

2 Enter LINEDEFS MODE. Right-click on **LineDef 50** (see Figure 2.38) to bring up the LINEDEF EDIT dialog box. Find **SUPPORT3** in the texture list. Transfer it to the BELOW WALL (lower texture) box for the FIRST SIDEDEF and the ABOVE WALL (upper texture) box for the SECOND SIDEDEF. Don't exit yet.

Figure 2.38.
Assign lower and upper textures to LineDef 50 and a switch to lower the lift. A walkover trigger on LineDef 47 will lower the lift from the other side.



Now we'll assign an ACTION TYPE to lower the lift. This will be a *switch* activator, and the FIRST SIDEDEF needs to be facing away from Sector 13. (Remember, all LOCAL ACTIONS must be activated from the RIGHT SIDE (FIRST SIDEDEF) of the LineDef.) If you split the Sectors according to the instructions, the FIRST SIDEDEF should be correct. If it isn't, right-click on it and choose FLIP LINEDEF from the FUNCTION MENU.

3 Select the MOVING THINGS option button at the top of the box and select **62 SR LOWER LIFT FOR 3 SECONDS**. Click OK.

Now we need to set the trigger to lower the lift when you cross over into it. This will be a *walkover* trigger, and it doesn't matter which direction the FIRST SIDEDEF is facing.

4 Double-click on **LineDef 47** (see Figure 2.38) to bring up the LINEDEF EDIT dialog box. Select the MOVING THINGS category and choose **88 WR LOWER LIFT**.

There's one more thing we need to do that might not seem obvious and is easy to miss. Although WinDEU will not tell us that we need lower textures on LineDef 47 (it has no way of knowing), we do. When the lift lowers, it will reveal the visible lower textures area of the FIRST SIDEDEF. Let's add a texture there.

5 Find **SUPPORT3** in the texture list and transfer it to the BELOW WALL box of the FIRST SIDEDEF. Click OK.

The lift is complete!

I mentioned at the beginning of this chapter than WinDEU has an automatic lift creation command. If you enter SECTORS MODE and right-click on Sector 13, the FUNCTION MENU has the command MAKE LIFT. If you were to select that, it would make a lift exactly like the one you just constructed. LINEDEF ACTION TYPES 62 and 88 would have been assigned, and it would then have put SHAWN2 on the lower texture of the FIRST SIDEDEF of LineDef 47. When you make your own lift, WinDEU can't predict the action of the Sector and that it will reveal upper or lower textures. (No editor I know of is this smart.) So you have to keep on your toes and consider all the possibilities of what might be revealed when you're constructing moving Sectors.

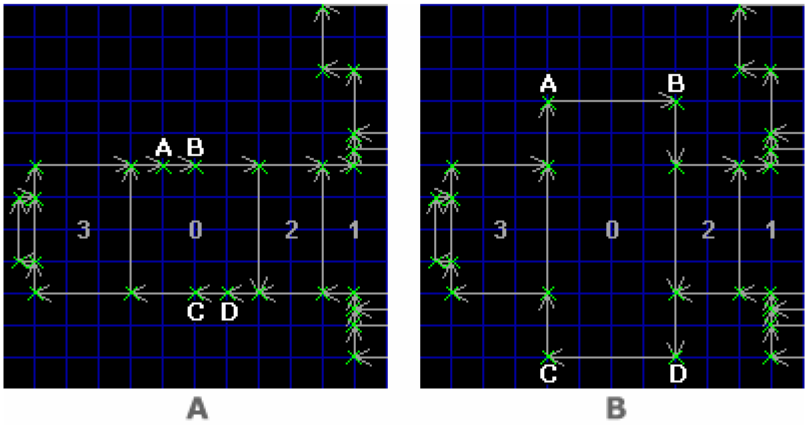
2.6 Making a Teleporter

Making a teleporter is extremely satisfying when you get it right, mainly because teleporters are very easy to get wrong. The reason they're easy to get wrong comes down to a simple matter of which direction the FIRST SIDEDEFS are facing on the teleporter pad. As mentioned in Chapter 2.4.1.1, LOCAL ACTIONS such as *switches* require that a LineDef be activated from the RIGHT SIDE, or FIRST SIDEDEF. Not so REMOTE ACTIONS, such as *walkover* triggers – EXCEPT in the case of teleporters, which must be crossed from the RIGHT SIDE, or FIRST SIDEDEF.

Let's make a traditional teleporter pad in Sector 0. First, we need to expand the Sector a little.

1. Enter LINEDEFS MODE. Select **LineDef 1** and right-click to bring up the FUNCTION MENU. Select SPLIT (ADD VERTEX) to split the LineDef. Select **LineDef 51** and split it again so that we have two Vertices (see Figure 2.39a).
2. Do the same with **LineDef 3**. Split it twice.
3. Enter VERTICES MODE. Move the Vertices to the positions shown in Figure 2.39b.

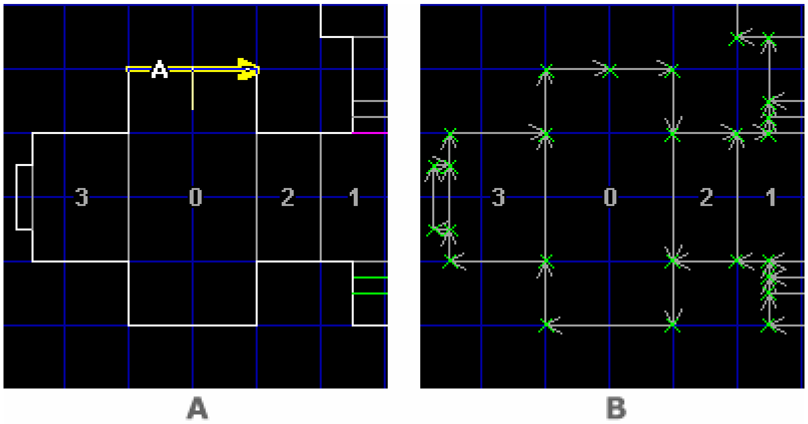
Figure 2.39.
Split LineDefs to expand Sector 0.



We're going to put the teleporter in the upper left corner of Sector 0. We need to create a Sector there, and we'll do it by splitting another LineDef and then by splitting the Sector.

1. Enter LINEDEFS MODE. Select LineDef 51, right-click, and choose SPLIT (ADD VERTEX).

Figure 2.40.
Split LineDef A to prepare a teleport Sector.



Remember that flats cannot be manually aligned. DOOM presupposes a 64-pixel fixed grid along which floor and ceiling textures are aligned. So when you construct a teleporter pad, you need to keep in mind that if you plan to use the GATE textures, you'll need the pad to be constructed on a 64-pixel grid. Change your grid scale to 64 (press G).

2 Enter VERTICES MODE. Mark Vertices **A** and **B** as shown in Figure 2.41. Right-click and choose SPLIT SECTOR from the FUNCTION MENU.

3 Enter LINEDEFS MODE. Select the new LineDef **A** and right-click. Choose SPLIT (ADD VERTEX).

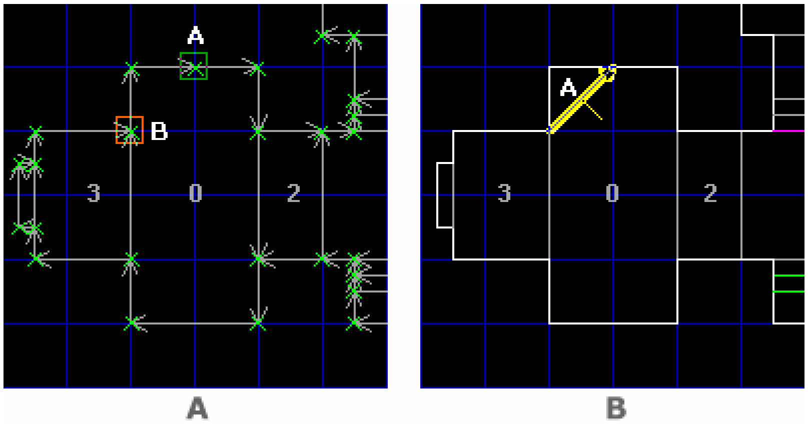


Figure 2.41.
Add a LineDef between Vertices A and B to split the Sector.

4 Enter VERTICES MODE. Move Vertex **C** to the position shown in Figure 2.42 to make a square.

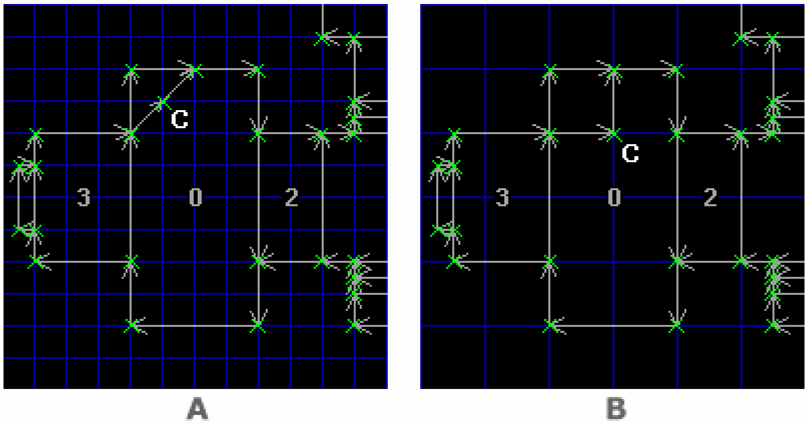


Figure 2.42.
Drag Vertex C to form a square. Note that the teleport pad is built on the 64 grid.

Let's dress the pad up a bit with textures and teleporter flats.

1 Enter SECTORS MODE. Double-click on **Sector 14** to bring up the EDIT SECTOR dialog box. Choose **GATE3** from the flats list and transfer it to the FLOOR TEXTURE. Find **TLITE6_1** and transfer it to the CEILING TEXTURE.

2 In SECTOR SPECIALS, choose **8 LIGHT GLOWS BETWEEN SET LEVEL AND NEARBY LEVEL**. Change the LIGHT LEVEL to **208**.

3. Change the FLOOR HEIGHT from 0 to **8**. Change the CEILING HEIGHT from 128 to **120**. Click OK.

Changing the Sector heights has revealed *upper* and *lower textures*. Let's fix those. First, make sure the LineDefs of the new Sector have their FIRST SIDEDEFS facing *out* and *into* Sector 0. They should, if you marked the previous Vertices A and B in a clockwise manner. If the FIRST SIDEDEFS are facing the wrong direction, mark them, right-click, and choose FLIP LINEDEF.

1. Enter LINEDEFS MODE. Mark both LineDef **A** and **B** (as shown in Figure 2.43) and double-click to bring up the LINEDEF EDIT dialog box. Find **METAL** in the texture list and transfer it to the ABOVE WALL and BELOW WALL boxes on the FIRST SIDEDEF.

2. Select the option box next to SPECIAL in the LINEDEF ACTION TYPES area. Find **97 w2 TELEPORT TO SECTOR** and select it. Click NEW (or enter **3**) for the SECTOR TAG. Click OK.

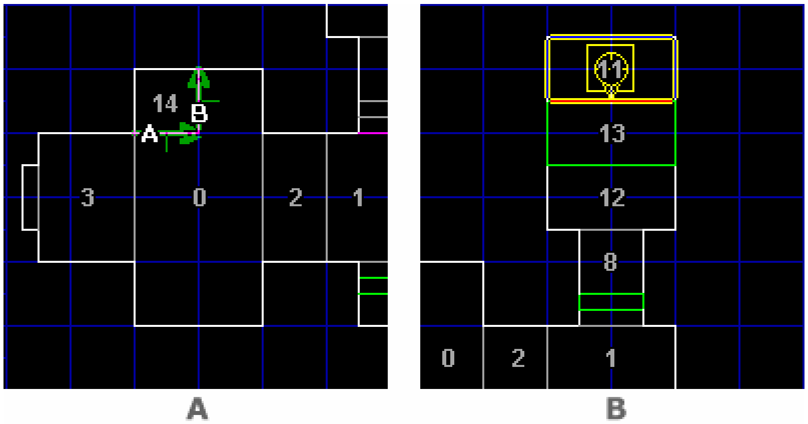


Figure 2.43.
A TELEPORT DESTINATION must be placed in the Sector you teleport to.

Now we need to choose a Sector where you want to teleport to. Let's use Sector 11 below the lift.

1. Enter SECTORS MODE. Double-click **Sector 11** to bring up the SECTOR EDIT dialog box. Enter **3** in the LINEDEF TAG text box. Click OK.

This links the Sector with the teleport LineDefs. If you highlight Sector 11, you'll see that the teleport LineDefs glow red, showing that they're successfully tagged. Now we need to add a TELEPORT DESTINATION Thing to Sector 11. This is like a PLAYER 1 START. We'll need to set the ANGLE so that the player faces toward the lift.

2. Enter THINGS MODE. Position your cursor in the center of **Sector 11** and press INS. Double-click on the Thing you added to bring up the EDITING THINGS dialog box. In the THING TYPE area, click the option box next to PLAYER POSITION. Choose **TELEPORT EXIT** from the list. Click the ANGLE option box next to **SOUTH**. Click OK.

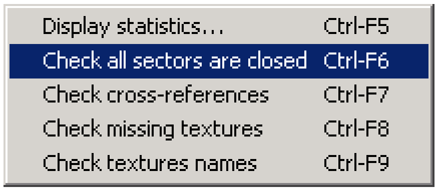
You've created a teleporter and destination. Teleporters need not be so visible. The TELEPORT MONSTER ONLY ACTION TYPE is a lot of fun to play with.

> **NOTE**: Teleporters must be entered from the RIGHT SIDE, or FIRST SIDEDEF. (The vector must be facing *out* of the Sector.) Teleporters are REMOTE ACTIONS, but are the exception to the rule.

2.7 Checking Your Level for Errors

Here's another tool you should use often to absolutely make sure your Sectors are valid. WinDEU has a set of error checking tools that examines your Sectors, LineDefs, Vertices, and Textures. Normally, if there's an error, WinDEU will tell you when you try to save your level, as the nodes builder may have trouble with an invalid level. But not always! So check your level often. To do this, you can click on **CHECK** in the MENU BAR and a drop-down box appears with a list of choices.

Figure 2.44.
WinDEU comes with a set of error checking tools.


Display statistics... Ctrl-F5
Check all sectors are closed Ctrl-F6
Check cross-references Ctrl-F7
Check missing textures Ctrl-F8
Check textures names Ctrl-F9

WinDEU runs through the tests fairly quickly, depending on the size of the level. Only when you have an error will you get a notification message. Do this often, because if you do have a Sector error, it can quickly lead to errors in other Sectors, because of the LineDef references.

CHECK ALL SECTORS ARE CLOSED is the most valuable error checking tool. An unclosed Sector can cause DOOM to crash. Unclosed Sectors are those whose LineDefs do not all reference the Sectors they belong to. LineDefs can have one or two sides, called the FIRST SIDEDEF and the SECOND SIDEDEF (or the RIGHT SIDE and LEFT SIDE). A one-sided LineDef has nothing on the other side of it, so its RIGHT SIDE, or FIRST SIDEDEF, will be facing into the Sector it is part of. The RIGHT SIDE must always be the side visible to the player. When a LineDef is shared between two adjacent Sectors, one side will face one Sector, the other the second Sector. Each side is given a Sector reference. This is the number of the Sector that it faces.

Figure 2.45 shows two Sectors that share a single LineDef. LineDef D therefore has two sides. The small line perpendicular to the LineDefs is called a **VECTOR** and always indicates the FIRST SIDEDEF, or the RIGHT SIDE. LineDef D's FIRST SIDEDEF would have a Sector reference of **0**, while its SECOND SIDEDEF would have a Sector reference of **1**. LineDefs **A**, **B**, and **C** are one-sided, so their FIRST SIDEDEFS face into Sector 0 and have the same Sector reference. LineDefs **E**, **F**, and **G** are also one-sided, and their FIRST SIDEDEFS face into Sector 1. Their Sector reference is therefore 1.

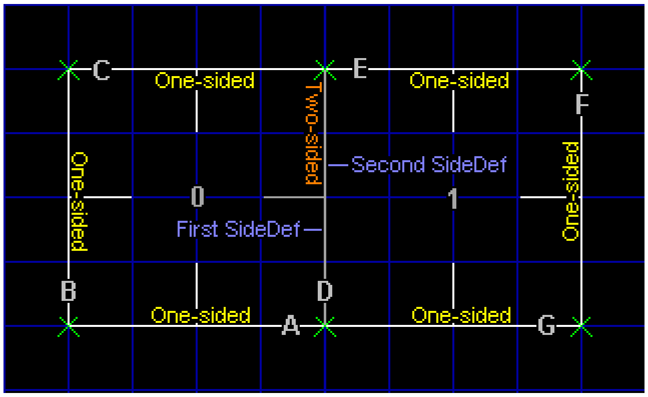


Figure 2.45.
LineDef D is shared between Sector 0 and Sector 1. It has two sides: the FIRST SIDEDEF and the SECOND SIDEDEF.

If a SideDef has the wrong Sector reference, the Sector is not closed. This can happen when you split Sectors incorrectly. The SideDef has to be manually edited in the EDIT SECTOR dialog box and given the correct Sector reference.

CHECK CROSS-REFERENCES makes sure that all of your Vertices reference their proper LineDefs, and that your SideDefs reference their proper Sector assignments. It also clears any unused Vertices and LineDefs that may be transposed with or buried under others. Run this as your map gets more complex.

CHECKING TEXTURES looks for missing *upper*, *lower*, and *normal textures*.

CHECKING TEXTURE NAMES looks for invalid texture names. This may only be a problem with your level if you've edited it with another level editor that allows manual insertion of texture names. (WinDEU v5.24 forces you to choose from a list of valid DOOM1 or DOOM2 textures. Later beta versions have boxes for manual entries.)

DISPLAY STATISTICS brings up a list of the number and sizes of all the entities in your map: Things, Vertices, LineDefs, SideDefs, and Sectors.

Few editors have such powerful error checking tools as WinDEU's. Unfortunately, WinDEU doesn't correct your errors. You still have to do the hard part.

2.8 Saving Your Level

Now, the most important thing: **SAVE** your level. Click on the first button on the TOOLBAR and then enter the name of your level. Click **OK**.

You may have noticed that a window with three progression bars opened up very quickly and then closed. This was the **Nodes Builder**, which runs a **BSP** (Binary Space Partition) compiling program on your level. This must be done to make your level playable. Later you may want to switch to other BSP programs (and other utilities for optimizing your map), which we'll get into later. For now, remember to **SAVE** your level often. WinDEU is very stable; I've never had it freeze up or crash on me. But things can happen, and there's nothing more heart-breaking than having added a bunch of cool stuff to a level you're working on, only to have Windows crash or freeze or your power go out – and BAM: You've lost all of your work because you forgot to save. It's happened to all of us.

So again: **SAVE your level often!**

Now go and test your level. (If you're completely new to all of this and don't even know how to use add-on levels, see Appendix E: How to Play Add-On Levels in DOOM for complete instructions.)

That's it for the basics of DOOM editing. I've shown you enough to get you up and running to make a few boxes in WinDEU. But you've still only just scratched the surface. Visit my website at Dr Sleep's DOOM Apothecary for more tips on level design, downloads of terrific DOOM utilities, add on levels, and other helpful documents.

You may want to check out Section 3.0, which explains some of WinDEU's advanced functions.

SECTION 3

WinDEU’s Advanced Functions

- *The Advanced Functions Toolbar and File Menu*
    - Grouping PWADs
    - List Opened WAD Files
    - Build New IWAD
    - Extract Object
    - Insert RAW File
    - Dump Entry
    - View Sprites

3.0 WinDEU's Advanced Functions

3.1 The Advanced Functions Toolbar and File Menu

WinDEU's advanced functions can be performed with the icon toolbar that is present when you first start WinDEU. It cannot be accessed while you have a map open. If you're currently editing a map, save it and close it by clicking on the X at the top of the window bar or choosing QUIT EDITOR in the FILE MENU.

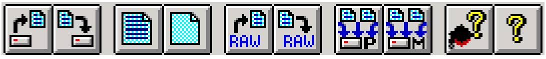


Figure 3.1.
The advanced functions toolbar.

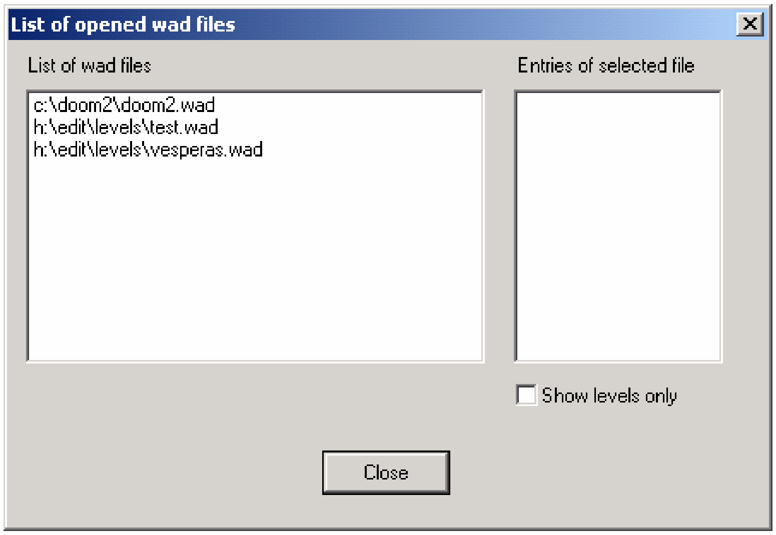
3.1.1 Grouping PWADs

Many of DEU's original command-line functions can be accessed through WinDEU's menus and icons. WinDEU allows you to merge or **group** two PWADs together so that you have two maps in one WAD file. WinDEU will only do two at a time, but conceivably if you had many levels you wanted to include in a single PWAD, you could just keep grouping two separate maps, then group the two map WAD with another two-map WAD and so on.

To group PWADs, first you must **read** the WADs into WinDEU by opening them.

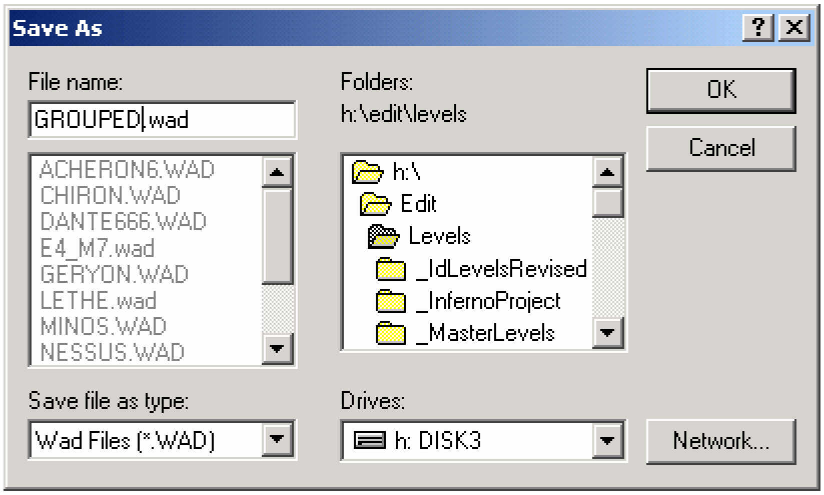
1. Open WinDEU and click the **OPEN** icon to load your first PWAD.
2. Click **OPEN** again and load the second PWAD.
3. Press **W** (or go to **FILE | LIST OPENED WAD FILES**) and WinDEU will show the LIST OF OPENED WAD FILES dialog box. (The DOOM2.WAD is shown by default, so don't worry about it being included.) Make sure your maps are listed.

Figure 3.2
WinDEU displays a list of all the opened WAD files, or WADs that have been *read*.



c:\doom2\doom2.wad
h:\edit\levels\test.wad
h:\edit\levels\vesperas.wad

4 Click the **GROUP PWADs** icon (or press **G** or choose **FILE | GROUP WAD FILES**). WinDEU will prompt you to enter a name for your new PWAD. Enter it and click OK.



File name: GROUPED.wad

ACHERON6.WAD
CHIRON.WAD
DANTE666.WAD
E4_M7.wad
GERYON.WAD
LETHE.wad
MINOS.WAD
NESSUS.WAD

Folders: h:\edit\levels

h:\
Edit
Levels
_IdLevelsRevised
_InfernoProject
_MasterLevels

Save file as type: Wad Files (*.WAD)

Drives: h: DISK3

Figure 3.3.
Give your grouped PWAD a new name in the SAVE AS dialog box.

You have grouped your PWADs.

WinDEU will only group two PWADs at a time. If you want to group a large number of maps into a single PWAD, you could group two at a time into single PWADs then group them to create one that contains four maps, and so on.

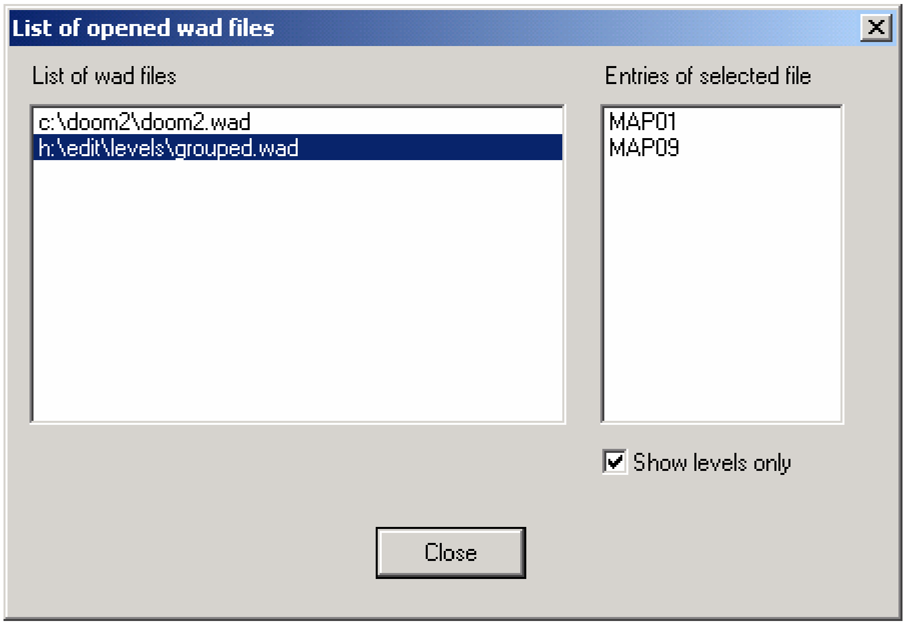
3.1.2 List Opened WAD Files

To make sure your maps have been grouped properly, WinDEU can show you a list of the currently open WAD files. When you select a PWAD, WinDEU will then display the directory entries in that file. PWADs with multiple maps will be shown with all of their *lumpnames* or map numbers.

1. **OPEN** your new PWAD.

2. Press **W**, (or choose **FILE | LIST OPENED WAD FILES**). The LIST OF OPENED WAD FILES dialog box appears. Click on your new map name.

The directory entries will be shown in the ENTRIES OF SELECTED FILE box on the right. You should see complete directory entries for both maps now. The DOOM2.WAD is listed by default. If you select SHOW LEVELS ONLY check box, only the *lumpnames* – or map numbers – will be shown (see Figure 3.4).



c:\doom2\doom2.wad
h:\edit\levels\grouped.wad

MAP01
MAP09

☑ Show levels only

Figure 3.4. Select the new PWAD to show the directory entries for the maps you merged.

3. Click OK to exit.

This function is useful for keeping track of which maps you may have loaded and merged.

3.1.3 Build New IWAD

This function will rebuild the entire DOOM.WAD to whatever filename you give it. Be sure to give it a different filename, by the way, or your original IWAD will be over-written! You may want to do this if you're replacing entries in the DOOM.WAD for some reason.

1. Click on the **BUILD MAIN WAD** icon (or press **B** or choose **FILE | BUILD NEW**) and enter the new filename. Click OK.

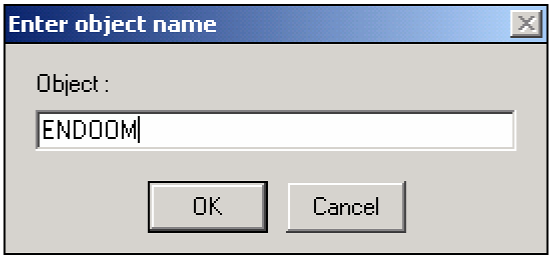
That's all there is to it. This function is recommended for folks who've studied the DOOM.WAD and know what they're doing. You should read *The Unofficial DOOM Specs v1.666* by Matt Fell.

3.1.4 Extract Object

To extract any entry from the DOOM.WAD is easy. You can extract textures, sprites, music or sound entries, patches, individual menu entries, or the ENDOOM data. In fact, you can extract just about anything. WinDEU will save these objects as *lumps* or *RAW* files. These can later be modified and imported into other PWADs.

1. To extract an object, click on the **EXTRACT OBJECT** icon (or press **X** or choose **FILE | EXTRACT OBJECT**).

The ENTER OBJECT NAME dialog box comes up. You need to know the exact entry name from the DOOM2.WAD. Let's extract the ENDOOM data. Always use ALL CAPITAL LETTERS for entry names.



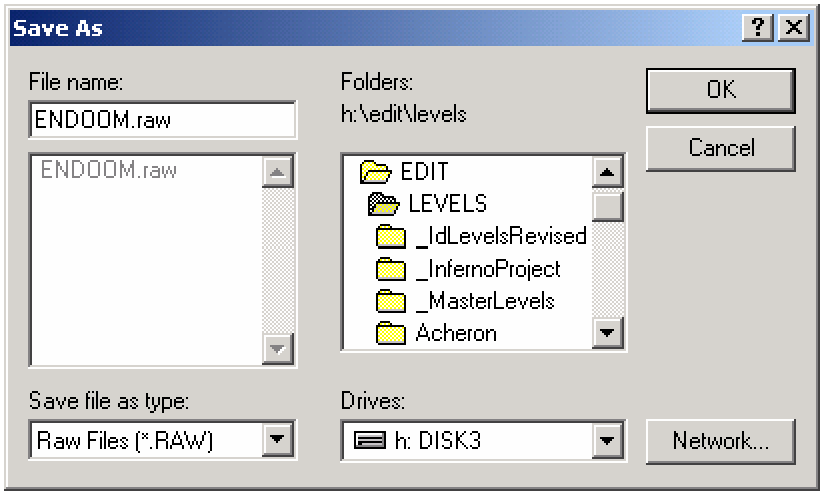
Object : ENDOOM

Figure 3.5.
Always use ALL CAPITAL LETTERS for the name of object you wish to extract.

2. Enter **ENDOOM** in all caps in the object text box. Click OK.

Now the **SAVE AS** dialog will appear. You can save the object as a WAD file (a PWAD will be created with the entry ENDOOM in it. There will be no other entries such as you normally see in map PWADs), or you can save it as a RAW file.

3. Click the LIST FILES AS TYPE drop-down arrow and choose .**RAW**. You should use the same name as the object to save as, so enter **ENDOOM** and click OK.



File name: ENDOOM.raw
Folders: h:\edit\levels
Save file as type: Raw Files (*.RAW)
Drives: h: DISK3

Figure 3.6.
Use the same name as the extracted object to name your RAW file.

You've extracted and saved your object.

The ENDOOM resource is the screen at the end of DOOM and that tells you what you're playing, how to order, and lists the credits for the folks at id Software. There are utilities out there - ENDOOM editors - designed specifically for altering the ENDOOM data. Many level designers alter the text and add their own name and the name of their level and other bits of trivia.

But before you can edit the ENDOOM data, you need to extract it from DOOM as you've just done. These utilities are made to import and edit lump files (files with the .RAW or .LMP extension), then resave them as LMPs or RAWs for insertion into a PWAD. That PWAD can then be merged with an existing PWAD to include the altered ENDOOM data. When someone loads your PWAD, the new ENDOOM screen will appear instead of the one in the DOOM.WAD.

A terrific utility for editing the ENDOOM data is **XWE**, the *eXtendable WAD editor* by Csabo. It not only edits the data, it can extract and insert data just as WinDEU does; though frankly, it is much easier to use than WinDEU, since it was designed primarily for this purpose. (You can download **XWE** from my web site, Dr Sleep's DOOM Apothecary, on the editors page in the downloads section.)

3.1.5 Insert RAW File

WinDEU doesn't insert a RAW file into your PWAD directly. It first must be inserted into a new PWAD and then merged or grouped with your existing one using the GROUP PWADS function.

1 Click on the **INSERT RAW** icon (or press **I** or choose **FILE | INSERT RAW FILE**). The **OPEN FILE** dialog box comes up with the default .WAD type.

2 Click on the LIST FILES OF TYPE drop-down arrow and change the file type to **.RAW**. Select **ENDOOM.RAW** and then click OK.

3 The **OBJECT NAME** dialog box comes up again. Type **ENDOOM**. Click OK.

Your file will be saved as ENDOOM.WAD. Confirmation of this is indicated in the STATUS BAR at the bottom of WinDEU.

WinDEU will not insert objects that are in graphic format. They have to be converted to RAW files or lumps. There are other utilities that do this. (In fact, there are other utilities that will insert objects into your PWAD in graphic format, and do it a lot easier and with fewer steps than WinDEU. **XWE** (as mentioned earlier) or **WinTex** are the most popular tools for doing this. But that's another lesson.)

Let's group the ENDOOM.WAD you've just created with TEST.WAD that you've been making.

1 Open TEST.WAD.

2 Open ENDOOM.WAD.

3 Press **W** just to see that the PWADs are open in WinDEU. Click OK.

4 Press **G**, or press the **GROUP PWADs** icon.

5 The SAVE AS dialog box comes up. Enter **NEW.WAD**. Click OK.

6 Open NEW.WAD and then press **W** to list the open PWADs.

7 Click on NEW.WAD in the LIST OF OPENED WAD FILES dialog box. (Click OK when finished.)

You should now see ENDOOM among the directory entries (usually at the very top of the list). You've successfully inserted a RAW file into your PWAD.

3.1.6 Dump Entry

The only advanced function left is the **DUMP ENTRY** function. This is for information purposes only, for the most part.

1 Under the EDITOR MENU choose **DUMP ENTRY** (or press **D**). The DUMP ENTRY dialog box comes up.

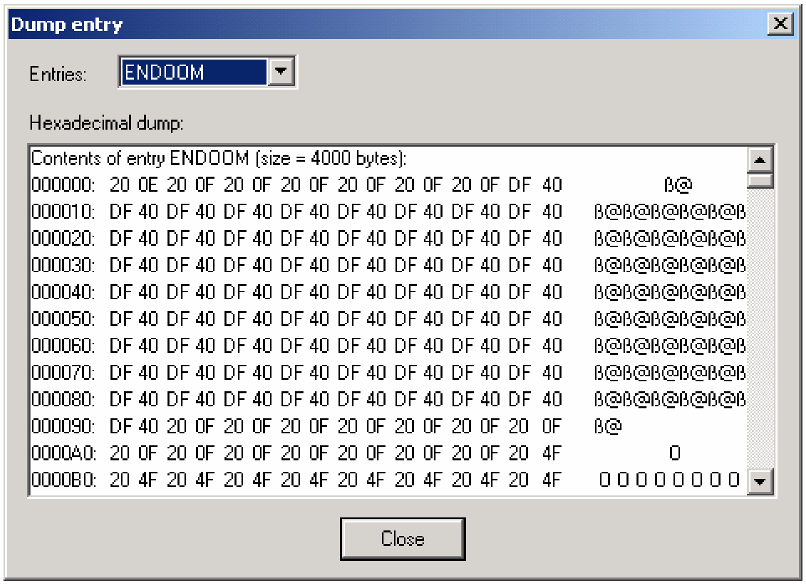


Figure 3.7.
The dump entry dialog box shows the entry in hexadecimal format.

Entries: ENDOOM

Hexadecimal dump:

```
Contents of entry ENDOOM (size = 4000 bytes):
000000:  20 0E 20 0F 20 0F 20 0F 20 0F 20 0F 20 0F DF 40                 ß@
000010:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000020:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000030:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000040:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000050:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000060:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000070:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000080:  DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40 DF 40     ß@ß@ß@ß@ß@ß@ß@ß@
000090:  DF 40 20 0F 20 0F 20 0F 20 0F 20 0F 20 0F 20 0F     ß@
0000A0:  20 0F 20 0F 20 0F 20 0F 20 0F 20 0F 20 0F 20 4F                    O
0000B0:  20 4F 20 4F 20 4F 20 4F 20 4F 20 4F 20 4F 20 4F      O O O O O O O O
```

2 Click on the ENTRIES drop-down list box and choose the **ENDOOM** entry from the DOOM.WAD.

WinDEU will display the information in hexadecimal format in the window below. You can cut and paste the information, if you know what you're doing with it, but there's no way otherwise to save it to a file, per se.

3.1.7 View Sprites

WinDEU has an added little treat: the sprite viewer. Under the EDITOR MENU, select **VIEW SPRITES** (or press **V**). This viewer is similar to the VIEWING WALL TEXTURES and VIEWING FLOOR/CEILING TEXTURES dialog boxes (which are also available in the EDITOR MENU). If you select one of the entries under **CHOOSE SPRITE NAME** and use the down-arrow key, the sprites will run through their animation frames.

Note the ZOOM FACTOR and GAMMA CORRECTION tools on the right.

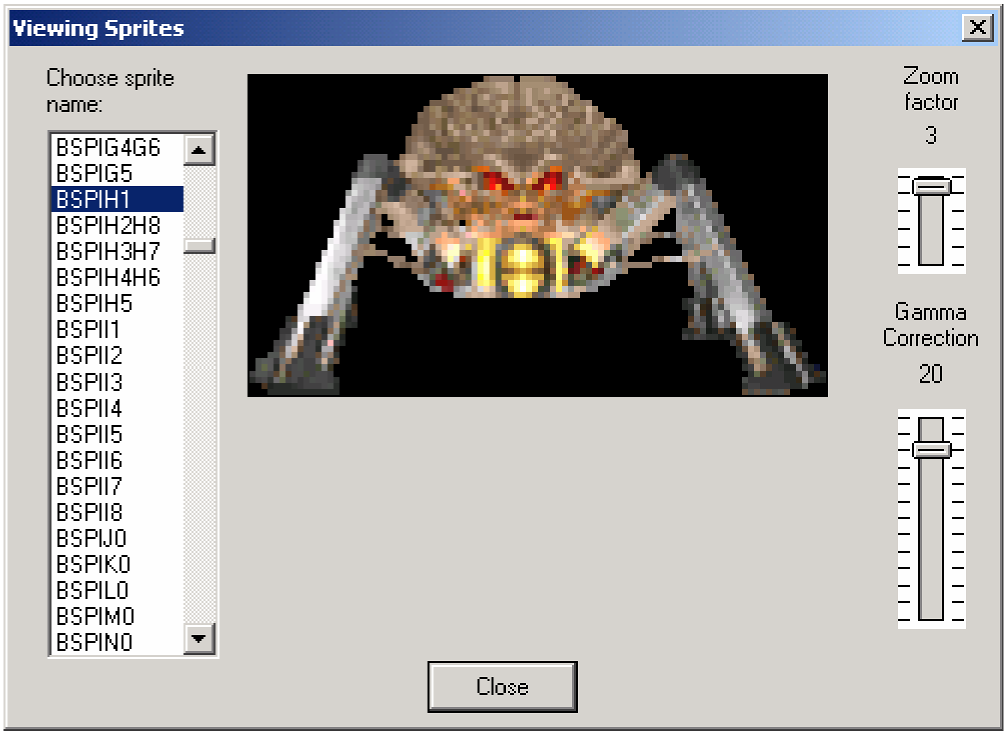


Figure 3.8.
The viewing sprites dialog box. Use the down-arrow key to make the monsters dance.

That concludes the end of Section 3.0 and the end of this manual. Good level design comes from practice. A tool is only as good as the person using it. WinDEU is a fabulous tool, and can provide you with the flexibility you need to stretch your imagination.

Acknowledgements

I hope this book has proved useful. It is not meant as an advanced lesson in level design, but merely as an introduction to getting around in the WinDEU editor's interface, and understanding how Sectors are created, for the most part. Many other editors allow line drawing to create Sectors, which is certainly easier than placing Vertices all over a grid. But this is a good way to learn how PWADs are constructed, since the new editors merely create the Vertices for you. They're still an integral part of a PWAD.

Many thanks to Raphaël Quinet for creating DEU, and to Renay Paquay for WinDEU. I made my first DOOM levels – *Dante's Gate* and *Crossing Acheron* – with DEU, and it lead to my getting hired by id Software to design *E4M7* "And Hell Followed" for **The Ultimate DOOM**, and five more levels for **Master Levels for DOOM II**. I'm therefore eternally grateful to Raphaël for providing the tool that led to my career in the games industry.

Thanks, of course, to id Software for creating DOOM.

Thanks also to Matt Fell for *The Unofficial DOOM Specs v1.666*, Hank Leukart for **The DOOM Hacker's Guide** (MIS Press 1995), and David Bruni for his insightful chapters on DOOM objects in **Tricks of the DOOM Gurus** (SAMS Publishing 1995).

WinDEU (and DEU, from whence it came) is now almost ten years old. It was a godsend when it came out in early 1995 because many level designers learned level design with DEU, but wanted an editor like it that worked in the then new Windows 95. Booting into DOS mode was a pain, and the DOS editors wouldn't run in a DOS box in Windows 95. Sure, there were other Windows editors like DoomCAD, but once one learns DEU, it's hard to part with it and learn a new editor all over again. WinDEU is still a powerful level editor, even ten years down the road.

If you have any questions, suggestions, or need more help, feel free to email me. Other tips for level designers are available in the Editing Clinic at my website, listed below.

John W. Anderson
04.04.2004
drsleep@newdoom.com

Dr Sleep's DOOM Apothecary
http://drsleep.newdoom.com/

APPENDICES A-E

- *Appendix A: WinDEU Hot Keys*
- *Appendix B: DOOM Metrics*
- *Appendix C: LineDef Action Type Reference*
- *Appendix D: WAD Authoring Template*
- *Appendix E: How to Play Add-on Levels in DOOM*

Appendix A:

WinDEU Hot Keys

Keyboard Quick Reference

Hot Key	Command
Q	Quit
ESC	Exit (no save)
Arrows	Move Pointer
Scroll Lock	Autoscroll on/off
Space	Scroll slow/fast
+	Zoom In
-	Zoom Out
TAB	Next Mode
SHIFT-TAB	Previous Mode
N	Next Object
P	Previous Object
J	Jump to Object
M	Mark/Unmark Objects
C	Clear All Marks
O	Copy Object(s)
D	Drag Mode Toggle
G	Set Grid Scale
H	Hide/Show Grid
ENTER	Edit Object(s)
INS	Insert Object(s)
DEL	Delete Object(s)
1, 2,…9, 0	Direct Zoom Setting
T	Thing Mode
V	Vertices Mode
L	LineDef Mode
S	Sectors mode
>	Next Object
<	Previous Object
#	Jump to Object #
SHIFT+H	Grid Scale=0
I	Info Bar On/Off
SHIFT+INS	Insert Lines and Close Sector
SHIFT	Move Pointer without Selecting New Object
SHIFT+M [or Left Mouse Button}	Box Object Selection
F1	Help
F2	Save
F3	Save As
F4	Search
F5	Preferences
F8	Miscellaneous Operations
F9	Insert Predefined Objects
F10	Check Consistency
CTRL+F5	Display Statistics
CTRL+F6	Check All Sectors Are Closed
CTRL+F7	Check Cross-References

CTRL+F8	Check Missing Textures
CTRL+F9	Check Texture Names

Pull-Down Menu Keys	Command
ALT+F	File Menu
ALT+E	Edit Menu
ALT+S	Search Menu
ALT+M	Mode Menu
ALT+I	Misc Menu
ALT+O	Objects Menu
ALT+C	Check Menu
ALT+H	Help Menu

Keyboard Commands Full Description

Q	**QUIT**, saving changes. You will be asked for the name of the PWAD file. **NOTE**: If you load a PWAD with multiple levels (or sounds, demos, graphics, etc.) and edit a level, **THE SAVE COMMANDS WILL ONLY SAVE THAT MAP**, not any other levels, graphics, etc. **DO NOT** name it the same as the multilevel PWAD.
ESC	**EXIT** without saving. If you have unsaved changes, a warning message will be displayed. May also be used to **CANCEL** accidentally selected functions**.**
Arrows	Move the **POINTER**.
Scroll Lock	Turn on/off the **AUTOSCROLL FEATURE**.
SPACE	Toggle slow/fast movement speed and the **SCROLLING SPEED**.
+, -	**ZOOM IN** or **OUT**. (Change the map scale.) ZOOM levels range from 1/20 scale (smallest) to 4/1 scale (major magnification).
1,2...9,0	Set **ZOOM LEVEL** from 1 to 10 directly. (1/1 - 1/10).
TAB	Switch to the **NEXT EDITING MODE**.
SHIFT-TAB	Switch TO THE **PREVIOUS EDITING MODE**. If objects are marked, the objects they are built from remain marked.
T	Switch to the **THINGS MODE**.
V	Switch to the **VERTICES MODE**.
L	Switch to the **LINEDEFS MODE**.
S	Switch to the **SECTORS MODE**.
N, >	Select the **NEXT OBJECT**. This will only work if the pointer is not on an object.
P, <	Select the **PREVIOUS OBJECT**. This will only work if the pointer is not on an object.

J, #	**JUMP** to a specified object (enter number).
M	**MARK/UNMARK** current object. (See also the SHIFT key below). **OBJECTS STAY MARKED UNTIL YOU UN-MARK THEM.**
C	**CLEAR** all marks.
O	Copy objects. After pressing **O** move the copy to where you want it and press **ESC** to drop it there.
D	Toggle **DRAG** Mode.
G	Show the **GRID** and change its scale. Press it again to increase the number of grid lines thus decreasing the scale. Use **SHIFT-G** to increase the scale.
H	**HIDE/SHOW GRID**. This only controls whether or not grid lines are displayed.
SHIFT+H	Set grid scale to 0.
I	Show or hide the **INFO BAR** at the bottom of the screen.
ENTER	Edit current object or group of selected objects. A menu will pop up and you will be allowed to change attributes in the object(s).
INS	**INSERT** a new object at the current cursor position. This will copy the last selected object or insert a default object. There are two special cases: When a group of Vertices are selected and you press INS, the editor will add new LineDefs between the vertices and will put you in the LineDefs editor. The editor will create LineDefs for all but the last line. (Use SHIFT+INS if you want to close the polygon). When a group of LineDefs are selected and you press INS, a new Sector will be created and one SideDef in each LineDef will be bound to this Sector and the edit mode will switch to the Sector editor.
SHIFT+INS	Use this when selecting groups of Vertices and you want the editor to close the polygon.
DEL	**DELETE** the current object or group of objects. All objects bound to the current object will also be deleted. (That is, if you delete one Vertex, this will also delete the LineDefs that used this Vertex). Except for Things, you will be asked for confirmation before the object is deleted.
CTRL	Hold the **CTRL** key while moving the cursor to prevent the pointer from selecting a different object.
MARK	Hold the **SHIFT** key while pressing **M** (or the **LEFT MOUSE BUTTON**) to drag a selection box around several objects and select them all at once. This is great for mass selection of any objects!
F1	**HELP** screen.

F2	**SAVE LEVEL** in a PWAD file.
F3	**SAVE AS**... (Change Episode and Level Number) This will allow you to reassign the episode and level number of a map. Enter a file name, then select the episode/level number.
F5	**PREFERENCES**. Use this to change the default values for wall, floor, and ceiling textures, and floor and ceiling heights. These defaults are for your current WinDEU session only. You may change the defaults for all WinDEU sessions by editing WINDEU.INI.
F8	**MISCELLANEOUS OPERATIONS.** The options that appear on this key vary depending upon which editing mode you are in. However, the first option is constant, no matter which editing mode you are in. **1. FREE TAG NUMBER** **2. ROTATE AND SCALE** **3. DELETE VERTEX AND JOIN LINEDEFS** **4. MERGE SEVERAL VERTICES INTO ONE** **5. ADD A LINEDEF AND SPLIT SECTOR**

Appendix B:

DOOM Metrics

By Scot Amspoker

With Addendum and supplementary material updates by Dr Sleep

Creation of a successful DOOM level requires that the PWAD designer be aware of the "physical" sizes and limitations of the Doom world. On a less critical level, the choice of wall, floor, and ceiling textures will have some influence on room dimensions. This document is primarily concerned with physical limitations. This document is not complete, and as I explain below, it is not absolute either. I regret that I haven't determined the map limits of a DOOM level. The data structures would appear to allow grid coordinates as well and floor/ceiling heights from -32768..32767. However, map editors impose stricter limits.

To determine physical limitations, I created a simple level with three sectors resembling this:

```
*-----------*         *------------*
|           |         |            |
|           *---------*            |
|           |         |            |
|           *---------*            |
|           |         |            |
*-----------*         *------------*
```

By manipulating the height and width of the center sector I was able to determine minimum clearances. While it was easy to perform tests with "the player", it was a little trickier testing the various enemies. I had to get an enemy to chase me through the center hallway. Unfortunately, this does not lead to definitive results. If the width of the hallway is near the minimum limit, the enemy is far less likely to enter the hallway within a short period of time. This would give the false impression that the hallway is already too narrow. I advise PWAD designers to allow extra clearance if it is intended that enemies move easily from one sector to the next.

After performing most of my tests, I received the April 13 version of *The Unofficial Doom Specs* by Matt Fell. Unlike the earlier version I had been using, this version provided the actual width and height of each object. I found this useful in confirming most (but not all) of my test results and for filling in the missing data on the Spider Mastermind which I hadn't tested at all. (Many thanks to Matt for his invaluable contributions). It appears that the minimum width through which an object will pass is greater than the actual diameter of the object. For example, The 32-wide player requires a width of 33 (although Fell reports a minimum width of 34). I will list the physical dimensions as well as the minimum clearances I have actually seen work. I left a value empty ("-") if I hadn't satisfactorily established a minimum.*

	Actual		Minimum Clearance	
Object	**Height**	**Width**	**Height**	**Width**
Player	56	32	56	33
Trooper/Sergeant	56	40	56	44
Heavy Weapon Dude*	56	40	56	44
Wolfenstein SS*	56	40	56	44
Imp	56	40	56	44
Demon/Spectre	56	60	56	64
Cacodemon	56	62	56	64
Lost Soul	56	32	56	36†
Hell Knight*	64	48	56	52
Baron of Hell	64	48	64	52†
Arachnotron*	64	128	64	132
Pain Elemental*	56	62	56	66
Revenant*	56	40	56	44
Mancubus*	64	96	64	100
Arch-Vile*	56	40	56	44
Spider Mastermind	100	256	100†	264†
Cyberdemon	110	80	110†	84†
Boss Brain*	16	32	-	-
Commander Keen (Billy)*	72	32	-	-

NOTE: Rising doors stop 4 units short of the neighboring ceiling. In order to ensure the minimum doorway height of 56, the neighboring ceiling should have a clearance of at least 60.

ANOTHER NOTE: When passing from one sector to the next, an object must fit between the higher of the two floors and the lower of the two ceilings.

Steps and Stairs

It appears that any creature with legs can step up as many as 24 units at a time. (I haven't verified this with the Spider Mastermind). As we all know, the player can step down from any height. Enemies appear to be limited to descending at most 24 units at a time. (Floating enemies aren't constrained by steps).

Although a simple step up or down is easy to handle, stairways are far more complicated. I figured that the depth of each step (the distance front of one step to the front of the next step) had to be a factor. The test results confirmed this but were confusing at best.

The player can climb steps with a depth as ***narrow as 1***! This could be creatively used in PWADS since it would appear that the player is scaling a sheer wall.

Enemies are much more picky about steps. For example, I discovered that a trooper would descend stairs that had a minimum step depth of 33 with the maximum riser height of 24. When I changed the riser height to some smaller value, the minimum required depth got smaller too. I gasped in horror realizing the amount of work it would take to figure out the magic formula, by trial and error, for each enemy. For what it's worth, the following table shows the minimum observed depth for 24-unit risers. You can assume these to be a safe value for any smaller riser. If your creatures appear to be afraid of the stairs, try adjusting the riser to depth ratio to create a kinder and gentler stairway. (I did not test the Cyberdemon or the Spider Mastermind in this area.)**

Object	Minimum Step Depth with 24-Unit Riser
Trooper/Sergeant	33
Imp	33
Demon/Spectre	51
Cacodemon	N/A
Lost Soul	N/A
Baron	41
Cyberdemon	-
Spider Mastermind	-

Wall/Room Dimensions

Unlike the physical limits described above, wall/room dimensions are determined more by aesthetic factors than by numerical requirements. However, in the absence of hard rules, there are magic numbers and standard situations that come up from time to time. It is not practical to cover every situation here. When in doubt, you should always examine any of the standard DOOM levels for guidance. It is helpful to always think in powers of 2. DOOM level editors should provide a user set able snap grid adjustable down to 8 units.

Room sizes are, of course, entirely arbitrary. However, hallway and corridor widths are commonly 64 or 128 units wide.

Most general wall textures are 128 x 128 units. The manner in which the wall textures are rendered is described in a separate document, *Managing Textures and the "Unpegged" Attribute*. The use of "unpegged" attributes and X, Y offset modifiers enables the PWAD designer to satisfactorily use these wall textures on any wall space.

Some of the available wall textures have more specific uses such as doors, switches, decorations, and wall thickness filler. Doors are usually 64 or 128 units wide. Door heights can vary. 72 units seems to be a common door height in the standard Doom levels.

Decorations and switches normally appear on wall textures that are 64 units wide (a few are 128). When placing these textures on non-standard wall widths, the decoration/switch will not appear centered without adjusting the texture X-offset.

Wall thickness filler is used on the insides of door tracks, windows, and the occasional alcove. Textures intended for such use normally 8 or 16 units wide but are designed to tile well with non-standard wall thicknesses and heights.

Special stair riser textures are 8 and 16 units high. However, any texture may be used on a stair riser.

Floors and Ceilings

The textures used for floors and ceilings are all 64 x 64 units. Unlike wall textures, floor and ceiling textures are aligned and oriented with respect to the underlying map coordinate grid. Since these textures are tiled globally, they will always create seamless transitions from one sector to the next (assuming the adjoining sector has similar floor/ceiling textures).

Some floor/ceiling textures have specific markings on them that are important. For example, some ceiling textures have a grid of lights. To ensure that the ceiling lights in a sector tile correctly, not only must the sector size be a proper multiple of a single light, but the sector itself must be placed on the map at an appropriate grid line.

A transporter pad has a special design. In order for the image on the transporter pad to appear correct, the pad must be aligned on a 64 x 64 map grid.

There are some special stair step textures (STEP1 and STEP2). These could be used for the top surface of each stair. If you wish to use these textures for your stairs, make sure you pick the texture with the proper north/south or east/west orientation and snap your stairs to a 64-unit width and a 32-unit depth.

Real World Scale

Occasionally, a PWAD designer wishes to create a DOOM level based on an actual place (such as his office building or home). I undertook such a project using the actual blue prints as my reference. This creates the problem of equating DOOM measurement units to real world units. There is no magic formula. The DOOM engine does not appear to render things with consistent aspect ratio. For my project, I found that the following scale produced a satisfying result that "felt right" when walking through the level:

- 16 Horizontal Doom units = 1 horizontal foot
- 10 Vertical Doom units = 1 vertical foot

You may wish to experiment with these values to suit your tastes. The vertical scale seems to be the hardest to pin down. Heights look different in DOOM depending on whether the player is near or far from the feature in question.

Addendum

DOOM Metrics was published before the December 15th revision of *The Unofficial DOOM Specs* by Matt Fell. This later edition included all of the updates for DOOM II, which had just been released, and therefore is missing from Scott's article. I've corrected spelling and some punctuation, but otherwise left the article exactly as it was written.

*I've updated the height and width table with all new entries on the monsters from DOOM II.

**I haven't tested the step depth distances.

†These figures were originally left null ("-") by Scott. I've since tested and updated the clearance figures.

John W. Anderson
drsleep@newdoom.com

Appendix C:

LineDef Action Type Reference

By Jim F. Flynn

LineDef Action Types by Function and Number

DOORS | CRUSHERS | CEILINGS | LIFTS | STAIRS | FLOORS | LIGHTING | TELEPORTERS | END LEVEL | MISC | TAGS

Credits

Special thanks to Matt Fell who's The Unofficial Doom Specs provided much new information for this revision, and [which] has been invaluable to WAD writers and DOOMers all along.

Thanks to Dr Sleep (aka John W. Anderson) for providing early information on the 1.666 LineDefs and for advice and encouragement all along. If you're looking for HERETIC information you want his htlndf11.txt file.

Thanks to Neil Bonner for pointing out the 667 Sector tag I was missing and thereby motivating this current revision.

Legend

Abbreviations

N. = neighbor
MIN = minimum
MAX = maximum
INC = inclusive
EXC = exclusive

A ***** appears on the left of any description line that only works for **DOOM 1.666** engine and above.

LineDef Description Headers

Example:

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
59	5	7	W1&	Open/wait 4/close	T/SEC/DMG	slow	(mover)

is the LINEDEF FUNCTION TYPE number 0-143

U1 and **U2** are the number of occurrences of a TRIGGER in DOOM1 and DOOM2.

TRIGGER represents the conditions that cause the function to be activated.

- The TRIGGER symbol may start with **n** if the function does not require a Sector TAG to operate.

The basic trigger symbol letters are as follows:

- **S** = SWITCH/door, FIRST SIDEDEF must be used with spacebar to activate.
- **W** = WALKOVER only teleport LineDefs require approach from FIRST SIDEDEF.
- **M** = MONSTER WALKOVER, activated only by monster walking over line.
- **G** = IMPACT TRIGGER, activated on hit: fists, chain, bullet, shell.
- **-** = no TRIGGER required (like animated wall).

The next letter in the TRIGGER symbol is the repeatability of the function: **1** for ONCE only (per LineDef) and **R** for REPEATABLE. This may be **-** if the repeatability does not apply, as for End Level.

The final letter in the TRIGGER symbol can be **m** or **&**. The **m** indicates that a monster can activate the function. The **&** indicates that once activated, all other functions on the Sector are locked out even after the **&** function is completed.

BRIEF DESCRIPTION attempts to state in English what the function does.

QUALIFIER shows which key is required, crusher attribute, texture changes.

SPEED is the rough relative up/down velocity involved in the operation.

SOUND is the name of the sound associated with the action.

Texture Change Descriptions

Texture changes involve copying attributes from another sector to the one that is changing. If the sector being copied is based on the line triggering the change, the description is prefaced with **T**: to indicate a trigger model change. This means that the sector on the FIRST SIDEDEF of the triggering LineDef is the one copied in the change. If the sector being copied is based on the sector being changed then it is the sector that is on the other side of the lowest numbered two-sided LineDef in the changing sector. Such a change is prefaced with **N**, for a numeric model change.

The remainder of the texture change description lists the attributes copied. **0** means that the changing sector type is set to 0 and only the floor texture is copied. **SEC** means that the secret attribute of the model sector is copied. **DMG** means that the damage attribute of the model sector is copied. Floor textures are **ALWAYS** copied. Ceiling textures, lighting attributes and heights are **NEVER** copied.

Floor Motion Directions

Floor LineDefs described as **DOWN TO** functions will move the floor at the speed indicated if the target height is lower. The height change is instantaneous if motion is in the other direction. Floor LineDefs described as **UP TO** act in a similar fashion but in the opposite direction.

Door Functions

A door function described using **COMMAS** only works when the door is stable in the opposite state to the function. A door function described using **SLASHES** will work anytime to toggle the opening/closing state of the door.

LineDef Types by Function

CEILING MOVERS DOORS, CRUSHERS, CEILINGS
DOORS **MANUAL DOORS** (no Sector Tag required)

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
1	281	220	nSRm	Open/wait 4/close		med	(door)
26	22	14	nSR	Open/wait 4/close	BLUE KEY	med	(door)
27	26	12	nSR	Open/wait 4/close	YELLOW KEY	med	(door)
28	10	9	nSR	Open/wait 4/close	RED KEY	med	(door)
117	*0	47	nSR	Open/wait 4/close		turbo	(blaze)
31	76	45	nS1	Open		med	(door)
32	15	40	nS1	Open	BLUE KEY	med	(door)
34	19	27	nS1	Open	YELLOW KEY	med	(door)
33	14	24	nS1	Open	RED KEY	med	(door)
118	*0	8	nS1	Fast open		turbo	(blaze)

CEILING MOVERS DOORS, CRUSHERS, CEILINGS
DOORS **REMOTE DOORS** (Sector Tag required)

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
29	1	0	S1	Open/wait 4/close		med	(door)
63	38	15	SR	Open/wait 4/close		med	(door)
4	0	1	W1	Open/wait 4/close		med	(door)
90	21	17	WR	Open/wait 4/close		med	(door)
103	41	32	S1	Open		med	(door)
61	9	36	SR	Open		med	(door)
2	114	64	W1	Open		med	(door)
86	9	3	WR	Open		med	(door)
46	13	22	GR	Open		med	(door)
111	*0	0	S1	Fast open/wait 4/close		turbo	(blaze)
114	*0	51	SR	Fast open/wait 4/close		turbo	(blaze)
108	*0	0	W1	Fast open/wait 4/close		turbo	(blaze)
105	*0	23	WR	Fast open/wait 4/close		turbo	(blaze)
112	*0	4	S1	Fast open		turbo	(blaze)
115	*0	11	SR	Fast open		turbo	(blaze)
109	*0	99	W1	Fast open		turbo	(blaze)
106	*0	6	WR	Fast open		turbo	(blaze)
133	*0	0	S1	Fast open	BLUE KEY	turbo	(blaze)
99	*0	2	SR	Fast open	BLUE KEY	turbo	(blaze)
135	*0	16	S1	Fast open	RED KEY	turbo	(blaze)
134	*0	4	SR	Fast open	RED KEY	turbo	(blaze)
137	*0	6	S1	Fast open	YELLOW KEY	turbo	(blaze)
136	*0	4	SR	Fast open	YELLOW KEY	turbo	(blaze)
50	*0	0	S1	Close		med	(door)
42	6	1	SR	Close		med	(door)
3	2	9	W1	Close		med	(door)
75	6	0	WR	Close		med	(door)

113	*0	0	S1	Fast close		turbo	(blaze)
116	*0	1	SR	Fast close		turbo	(blaze)
110	*0	1	W1	Fast close		turbo	(blaze)
107	*0	0	WR	Fast close		turbo	(blaze)
16	3	2	W1	Close/wait 30/open		med	(door)
76	2	2	WR	Close/wait 30/open		med	(door)

CEILING MOVERS **DOORS, CRUSHERS, CEILINGS**

CRUSHERS

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
6	0	0	W1&	Start fast (non-fatal)		med	(crush)
77	6	3	WR&	Start fast (non-fatal)		med	(crush)
49	0	1	S1&	Start slow (fatal)		slow	(crush)
25	0	0	W1&	Start slow (fatal)		med	(crush)
73	17	6	WR&	Start slow (fatal)		med	(crush)
141	*0	1	W1&	Start slow silent (fatal)		slow	(quiet)
57	0	0	W1&	Stop crusher			
74	24	13	WR&	Stop crusher			

CEILING MOVERS **DOORS, CRUSHERS, CEILINGS**

CEILINGS

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
40	4	0	W1	Up to max ceil exc		slow	(mover)
41	0	0	S1	Down to floor		slow	(mover)
43	0	0	SR	Down to floor		slow	(mover)
44	1	0	W1	Down to floor +8		slow	(mover)
72	0	0	WR	Down to floor +8		slow	(mover)

FLOOR MOVERS **LIFTS, STAIRS, FLOORS**

LIFTS

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
21	1	0	S1	Lower/wait 3/raise		fast	(lift)
62	51	143	SR	Lower/wait 3/raise		fast	(lift)
10	1	0	W1	Lower/wait 3/raise		fast	(lift)
88	65	51	WR	Lower/wait 3/raise		fast	(lift)
122	0	0	S1	Fast lower/wait 3/raise		turbo	(lift)
123	0	162	SR	Fast lower/wait 3/raise		turbo	(lift)
121	0	0	W1	Fast lower/wait 3/raise		turbo	(lift)
120	0	58	WR	Fast lower/wait 3/raise		turbo	(lift)

FLOOR MOVERS **LIFTS, STAIRS, FLOORS**

STAIRS

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
7	11	6	S1	Raise 8		slow	(mover)
8	2	1	W1	Raise 8		slow	(mover)
127	0	6	S1	Fast raise 16	CRUSH	turbo	(mover)
100	0	1	W1	Fast raise 16	CRUSH	turbo	(mover)

FLOOR MOVERS

LIFTS, STAIRS, FLOORS

FLOORS

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
58	5	0	W1	Absolute rise 24		slow	(mover)
92	0	0	WR	Absolute rise 24		slow	(mover)
15	0	1	S1&	Absolute rise 24	T:0	slow	(mover)
66	0	0	SR&	Absolute rise 24	T:0	slow	(mover)
59	5	7	W1&	Absolute rise 24	T:SEC/DMG	slow	(mover)
93	0	0	WR&	Absolute rise 24	T:SEC/DMG	slow	(mover)
14	1	0	S1&	Absolute rise 32	T:0	slow	(mover)
67	0	9	SR&	Absolute rise 32	T:0	slow	(mover)
140	*0	1	S1	Absolute rise 512		med	(mover)
102	14	22	S1	Down to max floor exc		slow	(mover)
45	0	2	SR	Down to max floor exc		slow	(mover)
19	11	22	W1	Down to max floor exc		slow	(mover)
83	0	7	WR	Down to max floor exc		slow	(mover)
71	0	24	S1	Down to max floor exc +8		fast	(mover)
70	3	2	SR	Down to max floor exc +8		fast	(mover)
36	8	27	W1	Down to max floor exc +8		fast	(mover)
98	4	0	WR	Down to max floor exc +8		fast	(mover)
23	16	12	S1	Down to min floor exc		slow	(mover)
60	0	4	SR	Down to min floor exc		slow	(mover)
38	23	37	W1	Down to min floor exc		slow	(mover)
82	6	0	WR	Down to min floor exc		slow	(mover)
37	31	11	W1	Down to min floor exc	N:SEC/DMG	slow	(mover2)
84	0	0	WR	Down to min floor exc	N:SEC/DMG	slow	(mover2)
20	13	12	S1&	Up to next floor exc	T:0	slow	(mover)
68	0	2	SR&	Up to next floor exc	T:0	slow	(mover)
22	3	18	W1&	Up to next floor exc	T:0	slow	(mover)
95	0	0	WR&	Up to next floor exc	T:0	slow	(mover)
47	0	2	G1&	Up to next floor exc	T:0	slow	(mover)
18	10	8	S1	Up to next floor exc		slow	(mover)
69	0	0	SR	Up to next floor exc		slow	(mover)
119	*0	13	W1	Up to next floor exc		slow	(mover)
128	*0	0	WR	Up to next floor exc		slow	(mover)
131	*0	2	S1	Up to next floor exc		turbo	(mover)
132	*0	0	SR	Up to next floor exc		turbo	(mover)
130	*0	0	W1	Up to next floor exc		turbo	(mover)
129	*0	0	WR	Up to next floor exc		turbo	(mover)
101	0	0	S1	Up to min ceil inc		slow	(mover)
64	0	0	SR	Up to min ceil inc		slow	(mover)
5	1	3	W1	Up to min ceil inc		slow	(mover)
91	18	1	WR	Up to min ceil inc		slow	(mover)
24	1	0	G1	Up to min ceil inc		slow	(mover)
55	0	0	S1	Up to min ceil inc -8	CRUSH	slow	(mover)

65	0	0	SR	Up to min ceil inc -8	CRUSH	slow	(mover)
56	5	0	W1&	Up to min ceil inc -8	CRUSH	slow	(mover)
94	0	1	WR&	Up to min ceil inc -8	CRUSH	slow	(mover)
53	0	0	W1&	Move min<->max floor inc		slow	(lift)
87	6	4	WR&	Move min<->max floor inc		slow	(lift)
54	0	0	W1&	Stop moving floor			
89	10	5	WR&	Stop moving floor			
30	2	2	W1	Rise by shortest outer lower		slow	(mover)
96	0	0	WR	Rise by shortest outer lower		slow	(mover)
9	2	0	S1	Donut function	N:SEC/DMG	slow	(mover5)

OTHER EFFECTS **LIGHTING, TELEPORTERS, END LEVEL, MISC**

LIGHTING

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
139	*0	2	SR	Light level to 0			(clunk)
35	3	14	W1	Light level to 0			
79	0	0	WR	Light level to 0			
138	*0	2	SR	Light level to 255			(clunk)
13	4	1	W1	Light level to 255			
81	0	0	WR	Light level to 255			
17	0	0	W1	Start 1 sec blinking (type 3)			
12	0	0	W1	Light to max n. light exc			
80	0	0	WR	Light to max n. light exc			
104	2	0	W1	Light to min n. light exc			

OTHER EFFECTS **LIGHTING, TELEPORTERS, END LEVEL, MISC**

TELEPORTERS

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
39	3	3	W1m	Teleport			(tport)
97	125	411	WRm	Teleport			(tport)
125	*0	4	W1m	Monster only teleport			(tport)
126	*0	20	WRm	Monster only teleport			(tport)

OTHER EFFECTS **LIGHTING, TELEPORTERS, END LEVEL, MISC**

END LEVEL

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
11	15	19	nS-	End level. Go to next level			(clunk)
52	19	55	nW-	End level. Go to next level			(clunk)
51	3	1	nS-	End level. Go to secret level			(clunk)
124	*0	4	nW-	End level. Go to secret level			(clunk)

OTHER EFFECTS **LIGHTING, TELEPORTERS, END LEVEL, MISC**

MISC

#	u1	u2	Trigger	Brief Description	Qualifier	Speed	Sound
48	99	103	n--	Animated wall			
0	0	0	n--	Null tag indicating no function is assigned			

Special Sector Tags

0 TAG used to specify no tag assigned. However, a LineDef assigned to tag 0 (none) will attempt to carry its function out on all 0 tagged sectors unless it also has 0 LineDef type. This usually causes the game to crash and is always weird - don't do it.

99 Artifact created by bug/feature of Id's BSP that collapses sectors that touch and have identical sector attributes. Used to prevent this. Has no effect on rising stairs other than this. Not necessary at all with Colin Reed's BSP.

666 In a transition level, when last Boss dies - Lower floor to min floor.

- DOOM I Boss = BARON, CYBERDEMON, SPIDER MASTERMIND
- DOOM II Boss = MANCUBUS, KEEN
- If **EXM8** and no 666 tag exists level ends on Boss death. The level cannot be ended on boss death in DOOM II, only Romero death.
- A transition level is **EXM8** in DOOM I, **MAP07** and an unknown list of others in DOOM II.

667 In a transition level, when last ARACHNOTRON dies - Floor rises 64. It is unknown if this tag has any effect or use in DOOM I.

A transition level is **EXM8** in DOOM I, **MAP07,** and an unknown list of others in DOOM II.

999 cf 99.

Appendix D:

WAD Authoring Template v2.0

Cut and Paste into Windows Notepad (or any text editor) and save as YOURWAD.TXT

```
==============================================================================
Title                     :
Release Date              : levelname.wad
Author                    : Your Name
E-mail Address            :
Web Site                  :
Other Files by Author     :
Misc Author Info          :
Additional Credits        :

Description               : (Set the mood here.)

Instructions              : (Where to unzip files, batch file operation, etc.)

==============================================================================

* Play Information *

Game                      : DOOM/DOOM II/Heretic/Hexen
Level/Map #               : E#M#/MAP##
Single Player             : Yes
Cooperative 2-4 Player    : Yes/No
Deathmatch 2-4 Player     : Yes/No
Difficulty Settings       : Yes/Not implemented
New Sounds                : Yes/No
New Graphics              : Yes/No
Demos Replaced            : None/1/2/3/All
DOOM Port Support         : JDOOM/ZDOOM/BOOM/Legacy DOOM/Other
Other Enhancements        : (DeHacked, new weapons, 3D models, etc.)

* Construction *

Base                      : New level from scratch/Modified ExMx/xxx.WAD
Build Time                :
Editor(s) used            : (Fill this in. Inquiring minds want to know.)
Other Utilities used      : (This too.)
Known Bugs                :

==============================================================================

* Revision History *

==============================================================================

* Copyright / Permissions *

This level is Copyright @ 2004 by yourname.

Authors MAY/MAY NOT use this level as a base to build additional levels.

(Use one of the following:)

You MAY distribute this WAD, provided you include this file, with
no modifications.  You MAY distribute this file in any electronic
format (BBS, Diskette, CD, etc) for non-profit purposes, as long as
you include this file intact (for instance, you may include it on a
```

CD that comes free with a book or magazine).

You MAY NOT distribute this WAD on any commercially released CD collection,
any shareware CD, or any other CD for which you charge a fee of any
sort without my prior permission. (Contact info here.)

You MAY not distribute this WAD file in any format.

===

* Where to get this WAD *

```
FTP Sites                 :
WWW Sites                 :
AOL/MSN/Other Online      :
BBS                       :
```

(This information form was originally provided as WADAUTHOR.TXT v1.4 with DEU 5.21.)

Appendix E:

How to Play Add-On Levels in DOOM

There are various ways to play an add-on level in DOOM. The parameters are

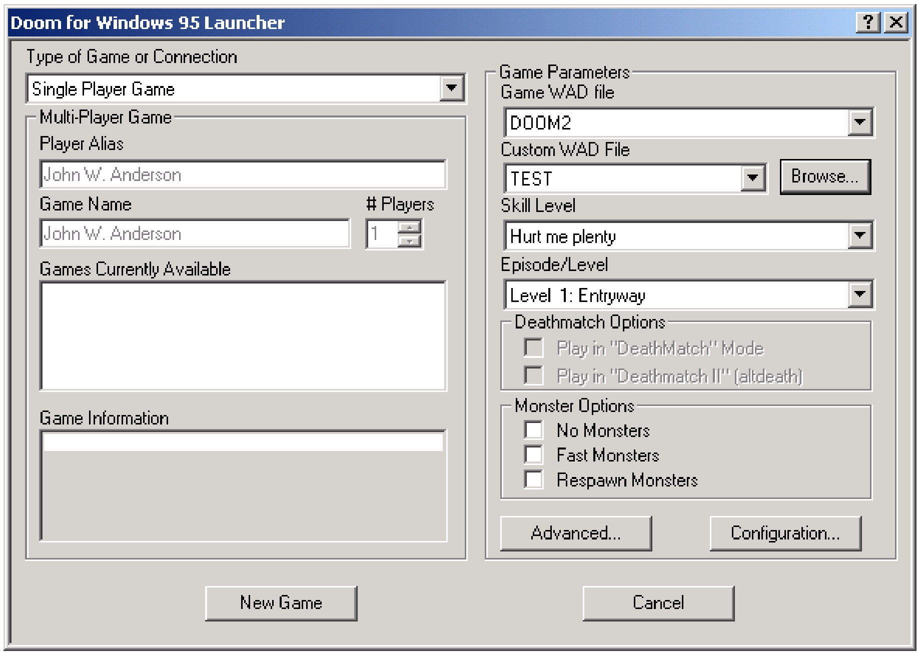
```
doom -file pwadname.wad -warp x y -skill n
```

Where **x** is the Episode number **1-3**, **y** is Level number **1-9**, and **n** is Skill Type **1-5.**

For DOOM II you only need one number **x** for the **-warp** parameter for Map number **1-32**.

```
doom2 -file pwadname.wad -warp x -skill n
```

There are Front End programs (such as KickStart for JDOOM and DLaunch for zDOOM) which enable you to add levels as you please. DOOM and DOOM II for Windows 95 come with a front end, which I'll dangerously assume you have. Double-click your DOOM95 icon to start the launcher.



Type of Game or Connection: Single Player Game
Player Alias: John W. Anderson
Game Name: John W. Anderson
Players: 1
Game WAD file: DOOM2
Custom WAD File: TEST
Skill Level: Hurt me plenty
Episode/Level: Level 1: Entryway

Figure E.1.
Click BROWSE to find the folder your level TEST.WAD is in. Click NEW GAME to start.

Just click **BROWSE** to find your level TEST.WAD. Make sure the **EPISODE/LEVEL** area is set to **Level 1**, since your PWAD is MAP01. Then click **NEW GAME**. That's it.

If you're running DOS versions of DOOM, you need to make a little batch file for adding PWADs.

How to Make a Batch File for Loading PWADs

If you're running DOOM in DOS or DOS Mode, here's a handy batch file containing the parameters you need to add a PWAD.

Open Windows Notepad or your favorite text editor. (Don't use Word or WordPerfect!) Enter the following, exactly as you see it:

```
doom2.exe -file %1.wad -warp %2 -skill 4
```

Now save it as **wad.bat** (or some easy and short name like **runwad.bat** or **addwad.bat**, etc.). Put it in your DOOM or DOOM2 directory and run it from there. When you go to run the file, type

```
wad [name of  PWAD] [MAPxx}
```

Note that there is a single space between the PWAD name and the MAP number. For instance, if you were loading DANTE.WAD which is MAP02, you'd type

```
wad dante 2
```

You can change the parameters to suit yourself, such as the Skill Level (if you don't enter the Skill Level, DOOM2 will default to Skill 3). You can remove the -warp parameter if most of the PWADs you run are MAP01, since DOOM2 will go to MAP01 by default. (The parameters are the same for shareware DOOM and The Ultimate DOOM. Just use **doom.exe** instead of **doom2.exe**.

The **%1** symbol stands for the PWAD name you type and **%2** the MAP number you type.

Read *The Official DOOM FAQ v1.666* by Hank Leukart for more parameters. The file is available with most versions of DOOM and DOOM2.

For more help, email me at drsleep@newdoom.com.