

Welcome to WinTex 3.4

[Introduction to WinTex.](#)

Installation of WinTex

[Set up](#) of WinTex

List of [external](#) Tools supported.

Using [Level Editors](#) with WinTex.

How to [Set up command lines](#) to support external tools.

Tutorials

General [Overview](#) of WinTex.

Tutorial on [Textures Editing](#).

Tutorial on [Floors and Ceilings Editing](#).

Tutorial on [Sprites editing](#)

Using WinTex to browse WADs

Listing the [Contents](#) of a WAD.

How to [get information](#) on WAD contents?

How to [Display](#) graphics and sounds inside a WAD?

Listing the [Textures](#) used in the levels of a WAD

Using WinTex to combine and install WADs

How to [merge](#) two WADs

Generating [Installation Batch](#) for sprites&flats WADs

Calling DeHacked and HHE from WinTex

Tips on using WinTex to build new WADs

How to [decompose](#) a WAD into a set of entries.

Changing [End Screens](#) (rating: easy)

Changing [Sounds](#) (rating: easy)

Changing [Musics](#)(rating: easy)Changing [Graphics](#) (rating: quite easy)

Changing [Levels](#) (rating: not really easy)

Changing [Wall Textures](#) (rating: difficult)

Changing [Floors and Ceilings](#) (rating: difficult)

Changing [Sprites](#) (rating: fairly difficult)

How to [make](#) a new WAD

Editing WAD resources

[Lumps](#) (general purpose data)

[Craphics](#) (menus, screens, faces)

[Sprites](#)(monsters, player, objects, weapons)

[Sounds](#) (strange background noises)

[Levels](#) (levels, reject, blockmap, nodes)

[Textures](#) (the walls)

Patches (texture components)

Flats (floors and ceilings)

Music (background musics)

Select only the entry types you want to put in your WADs

Choosing the format for extracting WAD entries

Registering WinTex

I have spent quite few hours on WinTex, as you might guess. However, despite the fact that it's of relatively good quality, I don't feel like releasing it as shareware.

However, if you use WinTex regularly, I respectfully request that you send me a little postcard and drop me a few notes about what you think of Doom and Heretic playing, multi-playing and editing. Of course, the more postcards I'll receive, the more I'll feel motivated :-)

But the idea is rather to see how far this program can go, and how many people are likely to get interested in Doom or Heretic editing. If I receive enough data, I'll post a summary to the net, of course.

Don't send money! I'm not American, I value a few friendly lines much more than money!

If you really can't afford a small stamp to France, then send an e-mail! :-)

My address is:

Olivier Montanuy
Manoir de Keringant
St Quay Perros
22700 Perros Guirec
FRANCE

Thanks

Many thanx to all the WinTex users on the net, especially those who tried the early alpha test versions, that were absolutely lame (pure Visual Basic! real slow!)

Among the guys I'd like to thank:

Avery Andrews and **Drake O'brien** for writing the tutorial for textures and flats.

Kevin McGrail (HevKev) for waiting patiently for the Help file (and WinTex, and Deutex 3.3)!

Marc-Andre Bissonnette for the early support on WinTex.

Jean François Fortin for early bickering (and general support) on WinTex

Dan Teeter that made me redesign of the front screen (stop it, Dan, that hurts!)

Renaud Paquay for telling me he could use the WinTex alpha (saved me from Dan)

Steve McCrea for support, help, and adjustments of the interface.

Tony Muray for suggestions about the help file.

Dr Sleep and the Compuserve Action forum for waiting so patiently :-)

My thinkpad, for crashing so elegantly under intense sollicitation.

Me, for doing all the underpaid work, including typing that manual that I hate.

You, for reading that stupid stuff (and lending me your Spitfire).

Thanks list for DeuTex:

Matt Fell for the DOOM specs.

Brendon Wyber, **Raphael Quinet**, **Denis Moller**, **Bernd Kreismeier**, **Bill Neisius**, **Elias Papavassilopoulos** and **Keith Wilkins** for ideas or exchange of ideas.

Mark Mathews, **Per Allansson**, **Chuck Rossi**, **James Bonfield** for the ports of DeuTex.

About WinTex

WinTex, the ultimate bastard WAD tool from Hell!

WinTex is **an integration tool**: there are many DOOM and HERETIC editing tool around, each specialised in it's own domain. WinTex is here to *make them work together efficiently*. Of course, WinTex has *it's own some special capabilities*, like Texture composition and entry identification.

WinTex doesn't do the main job itself because it would be ridiculous to reprogram stuff that already exist, and because Visual Basic is slow, inefficient, and not always reliable. Instead, WinTex is here only for the cool presentation! The serious business is done by specialised tools, which are all optimised in their own domain, and generally run in DOS.

If I was leading a team of ten programmers, I sure would have built a complete DOOM editing tool. But it would have been far beyond your budget!

However, by using WinTex, it should be real easy for you to build a great **DOOM, DOOM2** or **HERETIC** projects, containing not only custom levels, but also **custom textures, graphics, flats, sounds, musics and sprites!**

Of course, **WinTex is mainly built for use with DeuTex**, one of the best and of the most reliable WAD composer and decomposer. But it can also support all the tools you need to build your WAD, provided they run under your version of **MicroSloth Windoze**.

About this manual

This file is the result of long hours of work from me, Avery Andrews, Drake O'Brien and other cool guys. But it's still imperfect.

For more insight about DeuTex, read the **DeuTex manual by HevKev**.

For more insight into DOOM, see the **DOOM UNOFFICIAL SPECIFICATIONS V 1.666** or the book (in preparation): **Tricks of the Doom Guru** by **SAMS Publishing**.

Data Utility License

DOOM(tm) and HERETIC(tm) are trademarks of **Id Software**, Inc.

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

Data inutility license

The above statement is introduced by request of Id Software. Most of the legal grounds of this statement are may be found in the daydreams of Jay Wilbur and his lawyers.

However, if you intend to infringe the terms of this statement, you better get a good legal advisor, and keep my poor utility out of your dirty business.

WinTex SET-UP

To run WinTex, you must have **VBRUN300.DLL** in your **C:\WINDOWS\SYSTEM** directory. Vbrun300.dll should be included in your WinTex release. If not, then it is available in all serious ftp sites (ftp.cica.idiana.edu, warchive.wustl.edu), BBS and CDs. The size of that file is around 400k. If you ever used another Visual Basic tool, chances are that you already have vbrun300.dll on your system.

Create a new directory where you will put **WINTEX.EXE**, **DEUTEX.EXE** and possibly all your DOOM and HERETIC tools and all the associated **.PIF** files. This directory must be readable and writable, it cannot be on a CD. This is because files **WINTEX.INI**, **OUTPUT.TXT** and **ERRORS.TXT** will be read and written here. Put in the WinTex directory, or in your **WINDOWS\SYSTEM** directory, the files **LBDEUTEX.DLL**, **THREED.VBX** which are libraries for WinTex.

Never ever launch two WinTex at the same time. For some reason I cannot understand, my code cannot prevent you from doing that stupid mistake. Sorry.

If you want to use some special tool,s you may wish to configure the Command lines.

List of critical files in this release

WINTEX.EXE	The Visual Basic code.
LBDEUTEX.DLL	Some DeuTex code, Windoze DLL fashion.
THREED.VBX	Visual Basic 3D controls. (should be in your system directory)
DEUTEX.PIF	To run DOS DeuTex from WinTex.
DEUTEX.EXE	DeuTex, DOS version.
DEUSF.EXE	DeuSF, DOS version, to install your WADs on other PCs.
WINTEX.HLP	This help file.
*.PIF	DOS/Windoze Program Information Files.

WinTex External Tools

WinTex is designed to support many external tools.

The standard Windows tools are called in the standard way, you don't need to configure them.

But the tools which are specific to DOOM or HERETIC editing must be configured in

WINTEX.INI File because there is no standard way to invoke them.

Note that any tool that takes a somewhat reasonable standard line can be called by WinTex.

Supported level editors:

WinDEU	(windeu 5.3)
DEU	deu53.zip (and after, if any in that century)
DCK	DCK 2.1
EdMap	EdMap 1.81
WadEdit	WadEd 1.42
DoomCAD	dmcad51.zip

Supported WAD tools:

DeuTex	3.4	deutex34.zip (included in this release)
WadTools		wt2.zip
NewWadTools		nwt103.zip
Reject		reject11.zip (can run under windows)
RejectMapBuilder		rmb11.zip (can only run full screen)
End Text Editor		ted.zip (included)
Midi2Mus		midi2mus.exe (included)
Mus2Midi		mus2midi.exe (included)
BSP12x		if you can make it run in windows.

WinTex Initialisation file

WinTex uses a file called **WINTEX.INI** to store information about the current configuration, and the way to invoke the external tools.

For most of the tools, WinTex will ask you to designate the executable file and that should be enough.

However, it could be that the command line that WinTex will generate is in fact unsuitable for your tool. In that case, you'll have to edit WINTEX.INI (Menu **File | Edit WinTex.ini** from inside WinTex). All the commands are in the **[EXTERN]** section, and there is one command line per tool type.

Command lines structure

The command line look like ordinary DOS commands, except for some names like **%WAD%** or **%FILE%**. Those are placeholders that get substituted by actual indications, when the tool is to be called by WinTex.

For instance, **MIDI2MUS %MIDI%** will become **MIDI2MUS C:\MYMUSIC.MID**

<u>Variable:</u>	<u>Example:</u>	<u>What it represents</u>
<i>for most tools:</i>		
%WAD%	D:\DATA\MYWAD.WAD	A WAD file.
<i>For MUS and MIDI tools:</i>		
%MUS%	D:\MUSIC.MUS	A music file (MUS format)
%MIDI%	D:\MUSIC.MID	A music file (MIDI format)
<i>For level editors, reject and BSP:</i>		
%LEVEL%	E2M3 MAP04	A level name.
%DOOM%	C:\DOOM\DOOM.WAD	name of the main WAD.
%RWAD%	D:\DATA\MYWAD (.WAD)	A WAD file (root name, no extension)
%OLD%	D:\DATA\MYWAD.OLD	A WAD file (old version).
%DATA%	D:\MYTEX.WAD	A WAD with custom textures and flats file.

If your command line is incorrect, then WinTex will discard it.

When you call a new tool the first time, a default command line is generated. It may not work. It's also a good idea to provide a **.PIF** file in the WinTex directory, for that tool.

See the section about [level editors](#) for more information about the utilisation of your favourite editor in WinTex.

Using level editors with WinTex

DeuTex is a tool to help you build completely customised DOOM and Heretic conversion. It will help you build PWADs with custom textures, patches and flats. Of course, you might wish to use those custom textures and flats in your level editor.

See the sections about [Textures](#) , [Patches](#) and [Flats](#) to know how to build custom floors, ceilings and wall textures.

The basic idea is to first build a WAD with custom flats, textures and patches, so as to use it during level editing.

All the tricks of using a level editor in WinTex are explained in in the tutorial on [Textures Editing](#).

Overview of WinTex editing principles

The WinTex help service

- The **Blue Text** usually mean that when you click on it, an information box will pop-up.
- If you **double click on some frames**, then explanations concerning their contents should pop-up.
- If you **press F1** anywhere, it should provide you in-context help for this help file.
- When you press a command, a popping **Tutorial Box** will tell you what's gonna happen. but you can disable the tutorial mode.

And before you ask: WAD editing is non-trivial, so don't expect WinTex will make you understand it all in 20 seconds. Be patient, read the help, read the DOOM specs, and soon you'll be a DOOM and HERETIC guru like everyone else.

Tutorials available

Tutorial on [Textures Editing](#).

Tutorial on [Flats Editing](#).

Tutorial on [Sprites Editing](#).

The [Picture](#) tricks

Why WinTex is similar to DeuTex

WinTex is derived from DeuTex, and so it is organised **exactly like DeuTex**, my powerful but uneasy to use tool for DOS, Linux, OS/2 Unix(Sun Sparc, Silicon Graphics and DEC Alpha). Keep in mind that behind the WinTex screen, it's often DeuTex itself which is running (it will say: "DeuTex at work").

Don't expect this program will behave like the various WAD tools available all around. WinTex is safe. It doesn't allow you to fiddle directly with the inside of WADs, but rather it takes the burden of creating the WAD file with the right structure.

Basically, DeuTex uses a directory where they will store various data file, in subdirectories. A special WAD Information file will be in that directory, and will describe all that needs to be known about the entries that shall be put in the WAD to be built. WinTex uses exactly the same file structure.

WinTex/DeuTex directories

WinTex and DeuTex first need to know **the directory where you put your DOOM or HERETIC** files, unless you set the environment variable DOOMWADDIR (like under Linux/Unix DOOM).

WinTex also needs a [Working Directory](#), where it will store all the relevant data for the project.

This directory will contain a very important file [wadinfo.txt](#) that **describes what is supposed to be in the WAD you will build**. The [subdirectories](#), will contain all the data relevant to build your WAD.

Tips for using WinTex

If you're new to DOOM or HERETIC editing, I recommend you get a great patch WAD,

create a Working Directory, select that WAD as the WAD to be examined and decompose it by pressing extract. Then you can freely invoke the different Editing menus, and see how things go in this patch WAD.

Don't extract under a given game a patch WAD made for another game, or it will look real strange!

Once you're ready to create your own patch, **for each PWAD** you intend to make, You should provide to WinTex a specific Working Directory

Editing WADs entries under WinTex.

You will edit the contents of your WAD in the relevant **Editing panels** (available under the Edit menu).

Then you will **build** your WAD from the main window: you will first select which entry types shall be put in your WAD, then you will just press Build, and DeuTex will do the tricky job. Then, if you managed to build your WAD, you will check your WAD by pressing **Check**. and you'll read all the complaints that DeuTex will send to you.

Then you will build an installation batch for your files, if they contain Sprites or Flats or a de hacked patch.

Calling DOOM or HERETIC from WinTex

It is not possible to call DOOM directly from WinTex, because DOOM runs very slow under Windows, because of memory swapping, so I believe it isn't worth the bother. I could not manage to exit windows, play doom, and come back to windows. This should be possible, from the API calls, but it doesn't work with DOOM: the game just locks.

Feel like rewriting WinTex in pure C? contact me! I don't have time to do it...

Editing Lumps under WinTex

See the [Changing End Screens](#) section for general principles.

Lumps represent various resources that DOOM and HERETIC need to work, but that you don't usually need to edit.

End text of DOOM and Heretic, boot screen of Heretic

The lumps ENDOOM, ENDTEXT, LOADING are editable under WinTex. What happens is that WinTex calls TED in a full screen windows. You better know how to use TED, check the txt file. ATL-X to exit. ATL-W to write your changes (that will overwrite the previous lump, no back file, you've been warned).

Colour palette and colour maps

It is most unwise to edit the DOOM or HERETIC palette, PLAYPAL. Beware that if you spoil the first colour palette, and put insane values at colours 247 (DOOM) or 255 (HERETIC) then DeuTex won't be able to determine the transparent colour, and will refuse to work. Simple enough.

Future extension of WinTex might include a way to modify some colours.

The COLORMAP you might wish to recalculate yourself, if you're a brain in picture editing. WinTex won't help you there. see the doom specs.

Gravis Ultra Sound configuration

It's not real optimised, so they say. You might wish to replace that one with a file that is circulating on the net. Though it's text, WinTex can't let you edit it, because it's a Unix text format, not DOS. text editors would miss the newlines (blast them all).

Game demos

The lumps DEMO1 DEMO2 DEMO3 represent game demos, that you can record yourself with DOOM or HERETIC. See the doom command line reference. must be DOOM -record demo1.lmp or something.

Doom and heretic run lame under windoze, so WinTex doesn't even attempt to call them.

Other lumps

Let me know if there is something else we should care about.

Editing Levels under WinTex

WinTex will enable you to extract levels from a file, using DeuTex, to manage your list of levels, and to modify the levels in that list, by calling your favourite level editor on that level. Normal!

Moreover, if you've built an external WAD redefining textures, patches and flats, WinTex will call the level editor requiring it to load that WAD too (but some editors can't).

And of course, once you've built your level, you can call your favourite node builders and reject builders to build the missing data.

Building levels

From the Main Window, select menu **Edit | Level**. If you have [set up your level editors](#) correctly, you should be able to select a WAD in the list, and press on the button that corresponds to your editor, and then this level editor will be called with your level WAD as parameter.

Building Nodes and Reject

Most editors don't build the nodes and reject data themselves, because it's a very specialised job.

It's a good idea to use a dedicated and reliable tool for that. WARM, RMB, REJECT, BSP and IDBSP are examples of such tools. Most don't run under windows easily, alas.

Using WinTex, you should be able to always use the latest release of your favourite tools.

Always build **Reject** (and, maybe, optimised **Nodes**) at the last moment, after all the level editing, because those tools take a lot of time to run.

Using Custom flats, textures and patches

See the [Editors](#) section for the way to configure your editor for external texture and flat WADs.

Editing Graphics under WinTex

What are the graphics?

See the [Changing Graphics](#) section for general principles.

The graphics, are those pictures that are not related to the play field, as opposed to sprites, patches and flats. They are composed of the title screens, the help screens, the end screens, the menu bars, the end titles, everything in the status bar, the fonts...

WinTex will present you a list of graphics, with, in that order:

ENTRY NAME	X OFFSET	Y OFFSET	= FILE NAME
(name, in WAD)			(name on disk)

Special graphics

Under HERETIC, the title screens and the help screens are special. they are raw 320x200 pictures. WinTex can show them but can't load them into a PWAD.

Adding or removing graphics

Press [Add](#), and indicate a file a file somewhere in your directories, where you edit your pictures and it will be copied to the relevant subdirectory, and declared in the list. Press the **dust bin** to delete a graphic from the list, and possibly from the directory.

Reusing old graphics

Select an old graphic in the list of old graphics, and press [Import](#). Usually, you'd better replace a graphic by a graphic of the exact same size.

Editing your graphic

Double click in the graphics list and you will invoke your favourite graphic editor. See also the [Graphic Tricks](#) before you start composing a graphic.

Setting the graphic entry name

The name of the graphic entry is only meaningfull to DOOM or HERETIC, it bears no relation with the graphic file name. You can change it by **selecting a graphic entry name in the combo box** and pressing [Rename](#). You'll find it easier, in general, to have a file name equal to the entry name.

Setting the offsets (insertion points)

Most graphics don't need offsets. But the faces graphic, on the status bar of DOOM, need offsets. See the [Offsets](#) section for more insight.

Identical Graphics

if you don't want to compose a new picture for every graphics you can use the same picture for a serial of graphics.
The first graphic of the serial will indicate the picture, the other just repeat the previous one.
Select your graphic, and click on the Repeat previous check box.

Editing Sprites under WinTex

See the [Changing Sprites](#) section for general principles.

For more advanced topics, take a look at the [Using WinTex to build Sprites](#) section.

Description of screen

The sprite editing windows is composed of a **list of sprites**, and above that a **sprite display** frame. On the left you find a frame to set the [Offsets](#) of sprites, and above that a list of old sprites, that you can view, [Import](#) and modify

The stuff at the lower right corner indicates the [Phase](#) and [ViewPoint](#) that indicate when your picture will be used to represent the sprite.

WinTex will present you a list of sprites, with, in that order:

```
ENTRY NAME      X OFFSET      Y OFFSET      = FILE NAME
(name, in WAD)                                (name on disk)
```

The viewpoints

See the [ViewPoint](#) section.

Every sprite, at a given phase, can be seen from 8 different viewpoints (for instance, you can see a walking player from front, right-front, right,...). But some items will display the same, whatever your position toward them, because they have only one viewpoint (for instance, the barrels).

Warning: Omission of a sprite viewpoint is a very likely cause of game failure!

The phases

See the [Phase](#) section.

The phase indication doesn't adapt to the sprite, so check in the IWAD list that the phase you're defining effectively exists. You can also define phases that don't exist, but you'll have to modify the game program, else it won't take your new phase into account. (DeHacked and HHE are the programs to do that, for DOOM and HERETIC respectively).

Adding/deleting sprites

Press [Add](#), and indicate a file a file somewhere in your directories, where you edit your pictures and it will be copied to the relevant subdirectory, and declared in the list.

Press the **dust bin** to delete a sprite from the list, and possibly from the directory.

Reusing old sprites

Select an old sprite in the list of old sprites, and press [Import](#). This sprite will be added to the sprite list, and then you can redefine it.

Editing sprites

Double click in the sprite list and you will invoke your favourite graphic editor.

See also the [Graphic Tricks](#) before you start composing a sprite.

Setting the sprite entry name

The name of the sprite entry is only meaningful to DOOM or HERETIC, it bears no relation with the graphic file name. You can change it by **selecting a graphic entry name in the combo box** and selecting a viewpoint and phase for your sprite, and then pressing Rename.

Setting the Offsets (insertion points)

See the Offsets section.

Editing Sounds under WinTex

Changing sounds

See the Changing Sounds section for general principles.

it's easy, provided you can create WAVE sounds (WAVE is the default Windows format).

WinTex will present you a list of sounds, with, in that order:

ENTRY NAME	=	FILE NAME
(name, in WAD)		(name on disk)

What are the sounds?

You know that, don't you? Ever heard the scream of the horny imp, when it's about to screw you?

oink! oink! Well, now you know:

All you listen and that doesn't sound like music is a sound. easy to guess, no?

Adding/removing sounds

Select a Name for your sounds and press Add sound and select a sound file somewhere in your directories. it will be added to the list, and the file copied to the **SOUNDS** subdirectory of your **Working directory**.

Changing the sound name

you can change a sound name so that it corresponds to another sound effect.

select a name in the combo box, and press Rename.

Sound formats

use **WAV** only, **11025** sample/second. **non** compressed. **8 bit-sample**.

Do **not** use Stereo sound. there is no Stereo in DOOM or HERETIC.

Note: DeuTex can handle VOC and SUN Audio, but not WinTex.

The PC sounds

these are obsolete junk that is played on PC speaker, when no sound card is present.

Double click for a lovely text file! What can those numbers possibly mean?

Well, I didn't invert much on that part. Figure by yourself. Who cares for PC sounds?

Editing Music under WinTex

What are the musics?

See the [Changing Musics](#) section for general principles.

The musics are associated to a level and are played continuously when you play the level.

Musics are under very different format from sounds, and it is not possible to reuse a sound as music. Music is only a set of notes to play. Music files are usually MIDI or MOD. Only MIDI is supported, but there is a MOD to MIDI converter on the net.

WinTex cannot play MUS files, and it will invoke your favourite MIDI tool everytime you want to play a MIDI file. No, I'm not supposed to write a MUS driver for Windows, so you will need to convert your MUS files to MIDI.

Of course, every tools you can need to convert MUS to MIDI and inversely is included in the WinTex package, and WinTex will call them when needed.

WinTex will present you a list of Musics, with, in that order:

ENTRY NAME	=	FILE NAME
(name, in WAD)		(name on disk)

Identifying the musics

Use the [Name](#), as usual.

For Heretic, all the music names are not here, because there is not one music for each level. some music repeat, but I don't know how.

Adding/removing music

Select a [Name](#), press [Add](#) and select a **MUS** file. It will be put in the **MUSICS** subdirectory of your **Working Directory**.

Or select a [Name](#), press **Import MIDI file** and select a **MIDI** file, the converter will automatically be called on that file to convert it to MUS.

To delete an entry, press the **dust bin**, or the delete key.

Changing the music name

Select a music name in the drop box, and press on [Rename](#).

Converting MUS to MIDI

Warning: MUS2MIDI sometimes hangs...or so it seems... in fact, it's lost in one helluva loop that won't terminate before you finish your coffee, so go get one.

I'm not criticising, because I looked at the code and understood nothing at all, so I can't make it better.

Editing Textures under WinTex

What is a texture?

See the [Changing Wall Textures](#) section for general principles.
See the [Tutorial on Textures Editing](#) for more detailed explanations.

A texture under DOOM or HERETIC, is **the pattern defining what is displayed on the walls**. It is not a single graphic file, but rather a combination of graphic files that are called patches. So a Texture is no more than a patchwork... that makes sense, doesn't it? Why a patchwork? so that you can reuse the same graphic file in different situations, and thus generate many kind of wall textures, from a limited amount of graphic patches.

The two texture lists TEXTURE1 and TEXTURE2.

Textures under **DOOM2** are all in a single entry. But under registered DOOM and HERETIC they come in two entries, that you will need to define separately. That's why there is a **TEXTURE1** editing window, and a **TEXTURE2** editing window! Beware **not** to redefine the same texture name in **both** windows. Beware **not** to use **TEXTURE2** if you're doing a **DOOM2** level!

Choose your texture file

Under WinTex, like under DeuTex, the textures are stored in one or more **text files**. These files have a very simple text format, that can be edited by hand if really needed. But It's easier to edit it with WinTex!
If you use NWT, you may have noticed that NWT also can store textures under a Text format, and this is exactly the same as that of DeuTex, for compatibility reasons. So if you're not happy with the texture composer of WinTex, you can still use NWT...

The default name for a texture file is **TEXTURE1.TXT** or **TEXTURE2.TXT**.

Organisation of the screen

The screen is divided in no less than 6 part, which is real confusing the first time.

- The upper part is the **texture windows**. it contains the texture being composed.
- The lower right part is the **patch frame**, it contains patches that can be used to compose texture.
- The middle part is the **texture definition**, it contains the list of patches currently composing the texture, and buttons to fiddle with those patches.
- The lower part is the **texture file contents**, it contains the list of textures in the currently edited file.
- The lower left part is the **texture file list**. there can be many texture files, but preferably you should only use one, the default one.
- The middle left part is the **list of old textures**, as found in DOOM or HERETIC.

Creating or Editing a Texture file

Before you start editing textures, you must first select a text file where your texture definitions will be stored. got to the lower left part, and type a name in the text box, and click on it or press return to add this name to the list. If there already are files in the list, then just click on a name of the list. Then press the **Edit button**.

Loading the contents of another texture file

The **Read File button** enables you to read a texture file, and append the contents to your currently edited file. Typically you will use this option to merge two different texture files, or read some of your old texture files generated by DeuTex or by NWT.

Selecting a Texture to view or edit it

Just **Click** on a texture in the **texture list** and it will appear on the screen.

Changing the name of a texture

You can enter or modify the name of your texture in the text box situated in the upper part of the texture windows. Beware that if you change the name of a texture, you will create a new texture, not replace the previously existing texture.

Selecting the Height and Width of the texture

You can change the height and width of a texture by moving the scroll bars at the top and left of the texture windows.

Any Height between 4 and 128 can be selected, but beware that if you are composing a wall, that texture will be repeated every 128 pixel, so a texture of height below 128 will leave a very ugly mess on the screen. **Use 128 always.**

For Width, only the powers of 2 can be selected, i.e. 4,8,16,32,64,128,256,512, and there is no 1024 option yet (because 1024 is only useful for sky textures, and a real mess to display on a 640x400 screen! hand edit the texture if you need 1024.). **128 or 64 are recommended.**

Composing a texture

The section about [Patches](#) will tell you more about the patches you can use to compose your textures.

Warning

The transparent areas of the patches are **not transparent** when moving the patches. This is because there is no transparent Bitmap under Visual Basic, and the Windows call that is used for moving transparent stuff is much too unreliable to be used here.

So you roughly place your patch, then precisely adjust it's position once it is in the texture.

Adding a patch

Browse through the list of new patches, or the list of old patches, at the lower right corner, and select the patch you wish, and it appears in the texture screen, as a floating picture.

Left click and your patch will **drop** on the texture. **Right click** and your patch will be **discarded**.

Grabbing a patch

The patches only move when you're in the texture picture, and you hold it by the middle point.

If this is unsuitable, **click on the DeuTex icon** (the Devil) and you will **grab your patch** by the point you just selected on the icon! This is most unnatural to you, but so is the power of

the Devil!

Moving a patch

Select a patch in the texture contents list, and use the Up/Down/Right/Left/Bring Front buttons.

If you want to move the patch to a far away place, then click on the list, it will grab that patch. place it where you want. and then delete the previous instance of the patch in the list. easy!

Removing a patch

Select a patch in the texture contents list, and press DELETE.

Viewing the old textures

You might wish to display or modify textures that were already in DOOM or HERETIC. You must press the **Load Old button** to get the list of old textures, because that huge list is slow to read, so it's not done automatically.

Modifying the old textures

you need to import them into the texture file. press the **Import** button

Editing Patches under WinTex

See the [Changing Wall Textures](#) section for general principles.
See the [Tutorial on Patch Editing](#) for more detailed explanations.

You need to be in the **Patch editing** windows (Menu **Edit | Patches**) where you can add patches to your existing patches.

Warning

If there is a file in the New Patches list, that **has the same name as an existing** patch, then **it will redefine that patch**, even if it's not what you desire. This is because for DeuTex the loading of patches is entirely automatic.

If you want to avoid this, un-select the patches when building your WAD.

Defining new patches

Before using new patches in your textures, you must provide a graphic file that will contain your patch. Else WinTex won't allow you to compose a texture with that patch.

To define a new patch, click on the **Add Patch button**, and select a graphic file that will be copied into the new patch list.

Deleting new patches

If you want to get rid of a patch in the new patch list, select it and press **Delete**.

Beware that if your patch was used in a texture, WinTex will not even realise that, but DeuTex will then look for it in the main WAD. If it was really a new patch, and it was really needed, DeuTex will immediately stop building your WAD and report an error.

Editing patches

Double-click on the patch name you want to edit.

See also the [Graphic Tricks](#) before you start composing a patch.

Reusing old patches

Under WinTex, you can select a patch from the list of all graphic entries in the WAD. However, all those graphic entries shall not be reused. WinTex automatically removes the from the list the Flats, the Fonts, and the Status bar items, because they would crash DOOM. You can reuse Sprites (as was done in the famous TRINITY WAD) but if you want to reuse fonts, then you must provide your own fonts as separate patches, with names different from the names of the font in DOOM or HERETIC. Suggestion: use names FNT_A to FNT_Z. Of course, you can also reuse the old patches, that are situated at the very end of the list.

Now you can learn how to compose your [Textures](#).

Editing Flats under WinTex

What are the Flats?

See the [Changing Floors and Ceilings](#) section for general principles.
See the [Tutorial on Flats](#) for more detailed explanations.

Flats are the patterns that DOOM and HERETIC use for Floors and Ceilings. Changing those patterns is possible, and much simpler than changing the textures, because these are mere 64x64 bitmaps. Alas, it is also much more tricky, as DOOM or HERETIC can't read external Flats, unless you include all the flats into an external WAD. Same situation as for sprites.

WinTex will present you a list Flats, with, in that order:
ENTRY NAME = FILE NAME
(name, in WAD) (name on disk)

How can you change Flats?

Wintex will allow you to specify a list of new Flats. You then will use them for floors and ceilings in your levels. Then you will have to distribute your Flat WAD with DeusF, and an [Installation Batch](#).

How to use Flats in your levels?

Level editors should recognise the external Flat PWAD, and then you should be able to use your new Flats as well as the original Flats.
Alas, most editors won't recognise the external flats. Then the only option is to [MergeDTX_MERGE](#) your Flat WAD into the main WAD, do your level editing, and then restore your main WAD when you've finished all the level editing business.

Warning

DOOM flats are **64x64** bitmaps. Heretic also has some **64x65** bitmaps for flats, I don't know why and I don't think the last line is used in fact.
DOOM can take external Flat WADs, but HERETIC 1.0 cannot.

Animated flats

If you want to redefine an animation with flats, be sure to select (in the identification drop box) one start and one end of animation that exists in your doom.exe or heretic.exe. Then redefine those start and end with the patches you damn please, and put the intermediary steps **in between**.

Warning

If you redefine only the start or only the end of animation, you will crash doom or heretic. There will be no warning.

The Flat texture F_SKY1, indication of SKY

The flat named F_SKY1 is not a real Flat. It is never displayed. It only says to the DOOM or HERETIC engine 'please draw a sky texture instead of a floor and ceilings'.

So F_SKY1 a Flat that displays a texture! but a very special one, which looks like a fixed landscape very far away. There is no way to animate a sky texture.
If you want to redefine it, proceed like with any other textures, but beware that it needs to be 256x128 minimum, and only 1024x128 will give you a non-repeating sky texture.

Viewing WAD sounds and Graphics under WinTex

This panel is real easy to use, and you should figure it all by yourself. The only important thing is that you can save your entries to a directory, that you will specify at the top of the **Viewers window**, in the center. On the top left, you will find stuff to change the **scale** of the picture displayed.

Limitations of the entry identification

You can select to view the sounds, the graphics, the flats or the sprites. This should work with almost every WAD, but as they don't have a correct format, like the main WAD, there must be an entry identification first. The viewer that is included in WinTex will provide you a list of entries, but the entry identification is not as well crafted as in DeuTex, because it would be too slow. Thus sometime you will find names in the list that don't relate to any sound or graphic entry. Don't worry, this is normal, and the viewer will ignore them if you try to display them.

The loading of **sounds** under HERETIC is **real slow**. This is normal, because have to read a little bit of all the sound entries, and much more, to be sure to catch most of the sounds.

Displaying information about the WAD contents

The **Show** menu enables you to get some information about the contents of a WAD. First you should select the **WAD to examine**, by using the directory and file browser.

Listing the contents of a WAD

This invokes a WAD directory browser that will display the contents of the WAD directory. The name, offset and size of each entries are indicated. You will mostly care about the names and size. The offsets are meant for guru only (they tell you where to find the entry in the WAD...let WinTex and DeuTex handle this). Due to the limitations of windows, only 4000 entries can be displayed (this is not enough for the main WAD of DOOM2).

Identifying entries in a WAD

This calls DeuTex -wadir command, which give information about each entries in the WAD. The format is as follow:
Entry **name** Entry **size** in bytes Entry **identification** (Sound, Sprites, Level....).
This can help you understand how the WAD is organised.
This can also let you check that DeuTex identifies all your entries correctly

Viewing and listening entries in WAD

See the [Entry Viewer](#) section.

Listing space wasted in a WAD

See the [Wasted space](#) section.

Listing the textures used in a WAD

See the [Texture used](#) section.

Checking a WAD

will check that all the necessary patches and textures are present. It is slow because non optimised: but simple and secure.
Stuff **not checked** yet:
- multi-sprite viewpoints
- missing flats

Checking a Target WAD

Same check as before, but it is applied to the WAD you've just **Built**.

Decomposing a WAD

Extracting only one entry

For sounds and graphics, the Viewer enables you to extract single entries, as **WAV** or **BMP**, from the **WAD to examine**. They are written to the directory of your choice.

Extraction of single entries is not implemented for other entry types, because it's mainly useless.

Note that under most **Editing windows**, you can also import single entries from the main WAD.

Extracting all entries from an existing WAD

WinTex is especially meant to let you decompose, modify and rebuild existing WADs. This is very useful when you want to modify a WAD or convert it from, say, DOOM to HERETIC.

If you want to get all entries from a WAD, WinTex needs three indications:

- The name of Game directory, where you have put your main WAD.
- The name of the WAD file you want to examine.
- The name of Working Directory where all the data extracted from the WAD will be put (in subdirectories)

Generally, you will want to create a new directory as the working directory.

All you need to do once you've chosen those three items, is to press on **extract**.

If you already extracted files into this working directory, you will have to choose if you want to overwrite the existing files. Otherwise, information concerning the data extracted will be appended to your WAD information file, and no data file will be overwritten.

Selecting only some entries

You might not wish to extract all the entry types in your WAD.

Then you can select only some types. see the Entry Selection section.

So if you don't want to extract levels, don't select levels.

Beware that extracting all the entries uses a lot of disk space.

Using NWT or WT

WinTex can call New Wad Tool and Wad Tool. These tools are DOS tools that are convenient to browse WADs, and that you may find easier to understand than WinTex, because they provide less functionalities and assume you already know how to organise a WAD.

But they are an excellent introduction to WAD editing, and a tool like NWT is complete enough to be used as a complement to WinTex.

Efficient use of NWT

If you want to do little modifications directly to your WAD, use New Wad Tool, don't use obsolete tools. This is especially useful if you want to check your sprite animations directly in DOOM. With NWT, you can quickly insert your pictures in the WAD, even if they are not inserted right or cleanly, it doesn't matter, for experimental purposes. Use a PWAD to which all the sprites already have been appended.

Think of WinTex as a compiler, and NWT as an interpreter. They complete each other. Collaboration between DeuTex and NWT has made some of the batch mode function look similar, but of course that NWT approach to WAD editing is completely different from that of DeuTex and WinTex.

Building a new WAD

First you must select a **Target WAD** and a **Data Directory** that will contain the necessary informations to build a wad

Entry selection

You might not wish to put all the entry types in your WAD.
Then you can select only some types. see the [Entry Selection](#) section.

Build PWAD

the WAD will be build with the special PWAD convention.

Build IWAD

the WAD will be built as if it was a main WAD. It had better contain all the necessary entries!

Check WAD

Check the WAD you have just built.

Build a Project File

See [Makefile](#).

Build Install batch for PWAD Distribution

See [Install Batch](#).

Building a Project File (MakeFile)

This will put a MAKE.BAT file in your **Working Directory**. This batch file can be used to rebuild your WAD, by double-clicking on it from WinFile:: it will directly call DeuTex, you won't need to use WinTex. This is convenient once you have decided of the structure of your WAD, and make only minor changes that don't require WinTex.

But this file will also store the name of the **DOOM or HERETIC main directory**, depending on the game you built your WAD for, and it will store of the name of the **WAD to build or modify**.

Building an install Batch

This will create an **INSTALL.BAT** file that can install your WAd if it contains Sprites or Flats, or if you used a DeHacked or HHE patch. First you must tell the name of your WAD in **WAD to build or modify**, and your Dehacked patch in the menu **File | Select EXE patch**.

You then just need to distribute **INSTALL.BAT**, **DEUSF.EXE** and your **PWAD**.
If you use a **DEHACKED** or **HHE** Patch, you will also need **DEHACKED.EXE** or **HHE.EXE**.

Merging WADs

DeuTex was the first ever tool to implement that WAD merging feature. If DOOM and HERETIC did their job correctly, you could add as many PWAD as necessary, to replace all the entries you want, and that merging feature would be irrelevant. But alas DOOM has a bug there, and HERETIC seem to have an even greater bug! The merge functions are so necessary to play DOOM and HERETIC files, that a special utility, called DeuSF, has been derived from DeuTex: it's DeuTex limited to the merge functions! And you will often need to distribute it with your installation files (but it's small).

What's the use of the merging functions

the merge functions of DeuTex enable you:

- to install your WADs containing sprite and flats, for all versions of DOOM and HERETIC, without having to distribute all the internal sprites and flats of DOOM and HERETIC.
- to combine WADs together, especially WADs that redefine textures, sprites and flats (otherwise, most utilities can do that, even DEU. Use DeuTex because it also handles the tricky stuff).
- to install your wad inside your main wad, so that you replace DOOM or HERETIC by your favourite patch (but it's reversible).

There are many way to reach the same general goal, depending on whether you're paranoid about your main WAD, or about the wasted space on your disk, or if you prefer the old DMADDS way of working out this problem.

Warning:

Don't mix DOOM files with DOOM2 files: it could work, but don't expect miracle.
Don't mix DOOM files and HERETIC files: it will almost never work.

Merging a WAD into the main WAD

select a source WAD, **unlock** the main wad directory (so that it appears red) and use the menu **merge - merge PWAD into IWAD**. or press the merge button

Restoring the main WAD

unlock the main WAD directory (security) and as menu **merge - restore IWAD**, or press restore.

Joining two PWADs

This combines two PWADs together, merging their sprites, flats and even textures. sprites and flats is not a problem
select a source WAD, and a target WAD.
the source WAD will be merged into the target WAD. only the target WAD is modified
If entries are conflicting, those in the source WAD have priority

Restoring a PWAD

Whatever modification you do to a PWAD, you can restore your PWAD to it's original state, like the main WAD. **Even if you use some other tricky tool** on the WAD, DeuTex **might** be able to restore the PWAD. There is a **_DEUTEX_** directory entry that contains data for the

restoration. as long as this data is preserved, and the beginning of the WAD is not modified, DeuTex can restore.

Sprite completion

When you've build a sprite WAD that only replace a subset of the sprites, you shall complete that WAD with all the sprites in DOOM before attempting to play it. the sprite completion command is done for that.

Warning: HERETIC version 1.0 can't handle external sprite WADs. use the merge command.

To build sprites, See [Using WinTex to build Sprites](#)

Flat completion

same idea, but for flats.

Selecting only some entry types in a WAD

This is easy: you just check each entry name, on the down-left part of the main screen. there is a special button to select them all, or to select none, or to select only textures, patches and flats, so as to build custom texture and flat PWADs.

The patch trick: automatic loading

The patches are loaded automatically if present in the patches\ subdirectory of your working directory.

un-check patches if you don't want them

Output formats

There is no much of a choice if you're using WinTex. Only the Windows format are supported by WinTex, i.e. Bitmap and Wave sounds.

However, if one of your tools needs something else, like GIF, PPM, AU or VOC, then you can still request that format. but WinTex won't be able to handle it.

BMP

windows bitmap format. Only the 8-bit bitmaps are supported by WinTex. Only the 8 bit and 24 bit format are supported by DeuTex.

PPM

A trivial graphic format, to be used with the freeware package PBM PLUS popular under Unix. Only the P6 (24 bit raw RGB) format is supported by DeuTex.

GIF

Compuserve's Graphic Interchange Format. Popular everywhere.

WAVE

Window's sound format. Only the 8 bit PCM, non-compressed, single channel format is supported.

VOC

Creative voice format. Only the simplest format is supported: 8 bit PCM, non compressed, single channel, single chunk of sound data, no silences, no comments.

AU

SUN System format. Only the simplest format is supported: 8 bit PCM, non compressed, single channel.

DOOM or HERETIC game files

Why WinTex needs to know where your game files are?

WinTex needs to read the main WAD in the game, first to check that you are using a registered version, then to retrieve some basic information like directory contents, color palette and the like, and finally to give you the opportunity to reuse or modify those entries (import functions).

Selecting the game files

The game files can be selected from the WinTex main window. If your game file window says 'no game files', press the **change** button or select menu **Files | Set Doom directory**. Then a directory search box will pop-up, and you'll just have to point to the directory where you store your DOOM, DOOM2 or HERETIC files. WinTex will then tell you that it has found either DOOM, DOOM2 or HERETIC.

Source WAD

What is the source WAD?

This is a WAD that you wish to examine, to decompose, or to merge into another WAD. It will be used as a source of data, for most DeuTex commands. It will never be modified by any WinTex commands, only read.

selecting a source WAD

You select the source WAD from the main WinTex windows. As the source WAD often changes, there is a directory and file list box directly on the main window. It makes browsing for new WADs more convenient. Select the WAD file of your choice, by default the first WAD of the directory is selected, if there is one.

Target WAD

What is the target WAD?

When you're under the WinTex editing windows, you just describe what you'd like to put in your WAD. Then you'll need to go back to the main WinTex windows, and ask to build your WAD, from the files you provided to WinTex.

This WAD you're gonna build is called the target WAD.

More generally, it is used as the destination of some DeuTex commands.

For instance, when you merge two PWADs together, it's the Target WAD that gets modified: the Source WAD is appended to it.

Also, in the special case when you want to build new textures and flats for use with your level editors, it's also the Target WAD that will contain those new textures and flats.

If your level editor is able to support external textures and flats, just go under the level editing section, select a level and press on your favourite level editor, and if the Target WAD exists, it is fed to your level editor, so that it can get some textures or flats there.

Selecting a target WAD

In the main WinTex windows: Click on the **change** button, or select the menu **Files | Set Target WAD**.

Heretic WAD naming

The convention is that all WADs made for heretic should have a name beginning with **h**, for rather **h_**, so that they are not confused with the DOOM WADs.

For Strife, it will be the same idea: **s_**.

The Working Directory

What is the Working Directory?

This is the directory structure where all the data files describing your WAD are to be stored, generally in sub directories.

There should be **one** working directory **for every project** you make.

How to select a new working directory

double click on the directory name, or click on the change button.

The directory selection window appears.

If you type a directory name, just above the directory list, a new directory will be created.

Contents of the Working Directory

The working directory contains a main file, called WadInfo that describes what is supposed to be present in the WAD you just decomposed, or the WAD you are trying to build.

It will also contain the **batch file for installation**, if you make one.

Usually a sub directory will be created for each entry type that DeuTex needs (for instances, the sub directories TEXTURES, PATCHES, LUMPS ... are created if you edit textures, patches, lumps...)

Direct access to data in the subdirectories

for some operations, you might wish to gain direct access to your data file. You will find them in the directory. This is especially true if you work with external tools that generate a lot of data, like the raytracers (3D Studio, POVray, AutoCAD). see Making Sprite. With WinTex.

Here is the list of the sub directories, and what they contain.

The names of the subdirectories match the names of the sections in wadinfo.txt.

levels	*.WAD			Single level WADs
lumps	*.LMP			Lumps of raw data (format varies)
musics	*.MUS	*.MID		music files (MUS or MIDI format)
sounds	*.WAV	*.VOC	*.AU	sound files (WAVE, VOC or SUN AUDIO format)
textures format)		*.TXT		Text files (special DeuTex/NWT texture
patches Pixmap)		*.BMP	*.GIF *.PPM	Graphic files (Bitmap, GIF, Portable
graphics	*.BMP	*.GIF	*.PPM	Graphic files (Bitmap, GIF, Portable Pixmap)
sprites	*.BMP	*.GIF	*.PPM	Graphic files (Bitmap, GIF, Portable Pixmap)
flats	*.BMP	*.GIF	*.PPM	Graphic files (Bitmap, GIF, Portable Pixmap)

WadInfo file

What is the WadInfo file

This is the directory where all the data files that DeuTex needs are to be placed. There should be one data directory for every project you make

How to select a new WadInfo file

you can't. Wintex always use wadinfo.txt and wadinfo.bak as the previous file, before modifications.

Contents of the WadInfo file

That file describes what is supposed to be present in the WAD you just decomposed, or the WAD you are trying to build.

The content of that file should match the contents of the Data Directory.

Trick

The patches are never mentioned in **wadinfo.txt**.

The loading of patches automatic. all the graphic files that match a path name will be included in the WAD, unless you unselect the patches with the Entry Selection section.

Format of wadinfo.txt

You don't need to know that format, unless WinTex goes crazy and you have to modify that file by hand.

for each **subdirectory** of the Data Directory there is a correspondent entry in **wadinfo.txt**, that describe the contents of that directory.

for instance, the SPRITES contained in the WAD are described in the [SPRITES] section, and the graphic file that define those sprites are in the SPRITES subdirectory of the data directory.

Sections format

It's the same as what is displayed in the list boxes.

ENTRY [**Xinsrpt**] [**Yinsrpt**] [= **FILENAME**] [*]

ENTRY is an **entry name** (8 characters max, no space).

Xinsrpt is an optional **X insertion point** only relevant for graphic and sprites.

Yinsrpt is an optional **Y insertion point** only relevant for graphic and sprites.

= FILENAME is an optional **= sign**, followed by the **root name of the file** defining the entry.

If this sign and a root file name is not specified, the entry name is used as root file name.

***** is an optional *** sign**, it indicates that this entry is exactly the same as the previous one.

Don't put this indication on the first entry defined in a section!

Example of sound definition:

DSOOF

DSPISTOL = BANG
DSSHTGUN *

It means that the file DSOOF.WAV or .VOC will define the DSOOF sound entry, that the file BANG.WAV or .VOC will define the DSPISTOL sound entry, and that the DSSHTGUN sound entry is exactly the same as DSPISTOL, so there is no need to specify a file name.

Example of sprite definition:

PLAYA2A8 45 80 = HOMERA2
PLAYA3A7 *

it means that the BMP or GIF file HOMERA1 will define the entry PLAYA2A8, and that this entry will be placed at insertion point (45,80). Then the entry PLAYA3A7 will be the exact same entry as PLAYA2A8, so there is no need to specify an insertion point or a BMP or GIF file.

Valid sections

[LUMPS]	the list of raw data files
[LEVELS]	the list of WAD files that contain one single level (others are ignored)
[TEXTURES]	the list of files that contain textures definitions (the last one has priority)
[SOUNDS]	the list of sound files
[MUSICS]	the list of music files
[GRAPHICS]	the list of graphic entries
[SPRITES]	the list of sprites
[FLATS]	the list of floors and ceilings

there is no [PATCHES] section: Patches names are determined automatically by DeuTex)
there is no [PNAMES] section: Patch names list is built automatically by DeuTex)

Browsing the WAD directory

Pressing the **Contents** button will list the WAD directory of the file you selected as **WAD to examine**.

The WAD directory is a special part of a WAD file that describes the entries contained in that file, the size of those entries, and where to find those entries in the file. Consider it as the simplified equivalent of a file structure.

Listing the WAD directory is useful to check what is supposed to be in the WAD, and to check if the WAD structure is not perverted. If WinTex can't read the WAD directory, chances are that the WAD file is corrupt.

If you look at the main WAD of DOOM2, you will notice the end entries are missing. This is a bug of Windows: a list box cannot contain more than 4000 entries, and there is more than 4000 entries in DOOM2.WAD. You are using Microsoft Windows, keep smiling.

See [Identification List](#) for more information.

Listing the textures used in a WAD

This function will take a WAD, scan all the levels contained in that WAD, and list all the textures used in those levels.

The resulting list is not sorted. It lists the texture in the order that they appear in the levels.

Don't ask why there is such a function in WinTex. If you try to make a level conversion from DOOM to HERETIC, you will know.

Listing the wasted space in a WAD

This function will take a WAD, scan all the entries, and determine where some data is lost.

When you use tools that compose WAD directly, like DMGRAPH, DMAUD, NWT, WT, there is often some space wasted when entries are added or replaced. This is because the WAD format is not suitable as a modifiable filesystem.

Have a GOOD LAUGH! Look at OLD WADs with this!

WinTex will only show you if there are wasted space.

Tools like NWT have a special function to clean the mess they do.

Under WinTex you will prefer to **Extract** and then re-**Create** the WAD. That way you're sure that you only put in your WAD exactly what you want: unused WAD entries are deleted, especially those that DMADDS require and that are useless for DeuSF, and you will reform correctly the sound entries that DMAUD creates (and that contain sometime a lot of wasted data).

Warning:

On a WAD generated by DeuTex, or modified by DeuTex or NWT there are sometime some little amount of data that will be reported unused. This is normal! don't delete those data! DOOM and HERETIC can't use them, but DeuTex and NWT use them to restore the modified WAD.

Listing the identification of entries in a WAD

This function will use DeuTex to list all the entries in a WAD, like [Directory List](#) but will report:

- the Name of entries
- the Size
- the Identification of entries.

It is wise to take a look at this before using one of the Merge functions, to check if deutex identifies all the entries correctly. If it doesn't the Merge function could produce an incorrect result.

Of course, DeuTex should not fail, but in some special circumstances, if you used some reserved entry names, it is possible to confuse DeuTex (and probably DOOM also).

Using the Repeat Check box

If you have entries that are **the exact same as other entries**, you can use one crazy WinTex trick: declare an entry normally, and for all the entries that follow **declare that they repeat the previous entry**. So that in the WAD only the data for the first entry is effectively inserted, but DOOM and HERETIC will believe that the other entries exist too. *This trick is achieved by setting all the WAD internal pointers to the same data.*

You will know when an entry repeats the previous one, because there will be a * at the end of the line, in the **Entry list**. And if you select that entry, the **Repeat** check box will be **checked**.

Checking that box, if it isn't, indicates to WinTex that this entry repeats the previous one. As a consequence, you won't be able to edit that entry, because there is no file that is supposed to define it.

Using the Picture display

The Graphics, Sprites, Patches, and Flats editing windows have a picture display. by double-clicking on this display, you can invoke the bitmap editor.

Using the Add button

The Add button is to be used when you want to define a new entry, using an already existing file.

Under the Sprites, Sounds, Flats, Music editors, press the Add button and a new entry is added to the list. You are prompted for the file that will define that entry, and that file will be copied to the relevant subdirectory of the working directory. But the entry name will be taken from the Entry Identification Box because the name you want to give to a file had better make sense to you, whereas the entry name is hard coded in the game, so only a subset of names is valid.

Under the Patch editing windows, the Add button will add a file to the patch list. The patch name and file name is the same, because any patch name is valid, provided it doesn't correspond to a previously existing entry. The name of patches is not hard coded in the game.

Using the Entry Identification Box

An entry identification box is present in every editing windows where the game expects that you use a **certain subset of names**, that have a special meaning. For instance, BOSSA1 in the Sprites Editing windows means the picture represents a Baron of Hell (and it's the first walking phase, seen from the front).

WinTex help you by providing generally the **list of valid names**, with an indication of what they are used for.

But because sometime such a list would be too huge, some '?' characters are left, and you must **replace them by valid characters**.

However, in the Sprites editing windows, you just need to select a Phase and Viewpoint and press on **Main View** or **Inverted View** to replace the '?' by valid indications.

The good thing is that you won't need to browse the DOOM SPECS for the name of the entries that define stuff like animated walls, animated floors, sounds, monster names, level names because there is a list associated to the combo box, and you can get entry names from there.

Using the Rename Button

If you made an error, or want to change your mind about the way a given entry should be used in the WAD, select that entry, choose another name in the [Entry Identification Box](#) and press on the **Rename** button. Then the entry name will change, but of course the file name will not change.

Using the Import Button

You will use the **Import** button when you want to reuse a modified game entry in your new WAD.

The import button imports the entry you have selected in the [List of main WAD entries](#).

Note that if you want to import DOOM entries into HERETIC, you just need first to select DOOM as the main WAD, import the entries from DOOM, and after that use HERETIC as the main WAD.

Using the Import List

You will use the **Import List** to browse through the list of old entries, so as to know what kind of entry DOOM or HERETIC expects, when you intend to replace an existing entry. You will also use the **Import List** when you want to browse through game entries, and possibly import them, to modify them.

Using the Save button

You will use the **Save** button when you have made changes to your list of entries, and want to preserve those changes.

Note that when you **Quit** the **Editing** window with unsaved changes, WinTex will also ask you to save your changes.

Setting the Offsets

Some basic facts about the offsets

You will generally have to set the **Offsets** of your Sprite and Graphic pictures.

The offsets insidcate to the DOOM engine how to place a given picture on screen. This is quite tricky, so pay attention:

- For **Players, Monsters and Objects Sprites**, the offsets are **positive**. They indicate how the sprite should be placed in regard to the actual position of the sprite. This position of the sprite is indicated in WinTex by a cross hair, that only appears when offsets are positive.

-For **Weapons Sprites** (the weapon as they appear when you use them) the offsets are **negative**, and they represent the position of the sprite relatively to a 320x200 game screen. WinTex will place the weapon on the background screen, exactly as it would appear in the game.

- For the **Marine faces Graphic**, on the Status Bar of DOOM and DOOM2, the offsets are a bit special. Note that setting those offsets to bad value is a sure way to crash the game (A funny way to disable God cheat mode in your WAD? put bad offsets on STFGOD)

- **Mixing Positive and Negative offsets is not recommended**. If WinTex only display one of the crosshairs, you are mixing offsets, and you're bound to be in trouble.

Predefined offsets

To facilitate your job, WinTex includes several predefined offsets:

Creature Walking just the feet on the actual position, as if walking..

Missile, Explosion centred on the position. Ideal for missiles, explosions, flying monsters.

Weapon small suitable for centred weapons like the pistol, crossbow, chainsaw...

Weapon big suitable for centred weapons like the chaingun, staff...

Actually, no one seems to know why there are two possible offsets types for weapons

No offsets offsets to zero.

Normal suitable for wall patches and most ordinary graphics.

Marine face suitable only for the marine face on the status bar of DOOM .

Actually, the ideal marine face offsets may slightly differ, depending on the picture.

Modifying the offsets

If you want to set your offsets by hand, you can use the horizontal (X) and vertical (Y) scroll boxes. They will modify respectively your horizontal (X) and Vertical (Y) offset, and this offset will be reported on screen.

press on **Apply** to validate your changes. Unless you do that, all changes to the offsets will be discarded (because most of the time you don't want to fiddle with offsets).

The ViewPoint indications

The ViewPoint indication is pretty much like the Phase indication.

Explanations on the Viewpoints

When in the game you see the sprite from the left, from behind or from another viewpoint, you don't expect to see it the same way. As DOOM doesn't use actual 3D graphics for sprites, it needs to define generally 8 pictures for each sprite, depending on the direction you're looking from.

Of course, only 8 directions isn't much but it's enough to fool you, provided the sprite is moving. On static objects, this isn't always a wise idea, so most static objects, like cadavers, will look the same when seen from any direction.

This indication of the picture to be used, depending on the position of the player toward the monster, is called here the **viewpoint**.

This viewpoint is coded in the 6th and 8th character of the sprite name as:

0 = the picture is the same for every direction.
1 = the picture used when the sprite is seen from front.
2 = the picture used when the sprite is seen from front-left.
.and so on, by turning around the sprite, until:
8 = the picture used when the sprite is seen from the front right.

One or two viewpoint indicated per sprite name?

Another special trick of the game: when you indicate a viewpoint as the 6th character of the name, the picture is displayed normally, when you indicate it as the 8th character, the picture is displayed inverted. This is to code sprites that look the same when seen from left or right.

Safety rules

If viewpoint 0 is defined, no other viewpoint shall be used

If any other viewpoint (1 to 8) is used, then all viewpoints 1 to 8 must be defined, and 0 must not be defined.

HERETIC artifacts are the only sprite pictures that have no viewpoint defined.

Using Viewpoints in WinTex

To spare you the trouble of guessing the viewpoint, WinTex displays it directly on the Viewpoint indicator. it consists of 8 check box in circle, with a ninth one at the center, and a tenth one to indicate artifacts.

The rule is as follow: the little cross below indicates that the lower check box on the circle represents the front view for the sprite. The other check box on the circle define the other views, as if turning around the sprite. The center box indicates that the viewpoint is the same for every direction. The artifact box indicates there is no Viewpoint (artifacts only, of course).

Of course, when you select a sprite, it's viewpoint will be indicated.

To set a Viewpoint for a sprite name, in the Entry Identification Box, select it on the viewpoint indicator, and press **Normal View**. If you also want to set an inverted viewpoint, press **Inverted View**.

Animating sprites according to the viewpoint

When one of your custom sprites is selected, you can press the other viewpoints, and WinTex will automatically seek for your custom picture that matches that viewpoint. This is great for testing animations, and for checking that all the needed viewpoints are present.

Using the phase indication

The **Phase** indication is pretty much like the [Viewpoint](#) indication.

Explanation on the phases

Most sprites of the game go through different phases, and a special (set of) picture is associated to each phase. For instance, a monster that walks generally go through four phases, as it's legs and body move while it walks.

The phases are coded in the sprite name as letters, from **A** to **Z** and a bit beyond, for the ArchVile.

Using phases in WinTex

Unfortunately WinTex cannot know, for a given sprite what phase represent, from simply the letter used. To know that, WinTex would need to read the game executable, and it just never does that, because the game executable has an undocumented and unstable structure.

There is an excellent program called DeHacked (for DOOM) or HHE (for HERETIC) that you will use to check that your sprites are used the way they should. This program can animate sprites because it can read the tables where the sprite behaviour is coded.

But of course, under WinTex, you can still use the [List of Old Entries](#) to guess what a phase corresponds to, for a given sprite.

Animation with the phases

If you are viewing one of your custom sprites, and more the phase scroll box, then WinTex looks for that of your sprites that corresponds best to the currently indicated phase and viewpoints. This is great for checking that the necessary viewpoints are defined for every phase, but of course that it a very poor indication of how the phases will effectively follow each other in the game.

Using WinTex to redefine Sprites

DIFFICULT

Redefining sprites: what for?

Sprites are the visual manifestations of things and events in the Doom environment- players, weapons, monsters, explosions,etc.

Using WinTex you can change the appearance of:

- **monsters**. Want to fight Barney or the Energiser Bunny?
- **objects**. Want to replace those Stimpack by a donnut?
- **weapons** and **missiles**. Want a to toast your pals with a flame thrower?
- **add viewpoints** to some flat objects. Want to replace barrel by a PC, and see the view change as you turn around it?

Generally, the sprites are used for all the stuff that can move, and for the decorations.

There are two basic types, player-weapon sprites, and the rest. Player- weapon sprites appear in a fixed position on the screen, while the others ('environmental' sprites) have a location in the Doom environment, so that where they are on the screen depends on where you are.

The behavior of all sprites is largely determined the Doom exe, so that the main thing you can do with WinTex is make graphic substitutions. It is however possible to add extra viewpoints to environmental sprites, and to put sprites onto walls (to scare the players!). What you can't do is add completely new ones - add a desk-lamp to the collection of decorations, for example.

The Sprites Tricks

See also the more general [Graphic Tricks](#).

- **Make smooth and natural animations** as even the best looking sprites will look mechanic, if it's arms and legs are rotated along only one axis, and only a few parts of it effectively move.
 - **Don't make front view** of your sprite, they look ridiculous. See that no DOOM sprite is ever seen directly from front, always a few degrees to the left or to the right.
 - **Don't be afraid of adding Viewpoint**: most sprites under DOOM or HERETIC only have a subset of the 8 viewpoints, but your replacement sprite, being usually asymmetric, might require more viewpoints than are available in the original sprite. **DeuTex and DeuSF will handle that for you, and they are the only tools that can do that!**
- Warning:** check you don't forget some viewpoints!

Sprite naming convention in DOOM and HERETIC

- the **sprites names** have 4 letter (except the artifacts).
- Then comes **a letter** that defines the Phase (A,B,...Y,Z...) when the sprite is to be displayed
- Then comes **a number** define the Viewpoint
- Then **optionally** comes another letter and number, with the same meaning. But the sprite will be displayed in reverse, **allowing to use the same picture for the left and right view of a sprite**, if it is **symetric**.

Declaring custom sprites to WinTex

You will first select a Working Directory. this is the place where all your sprites .BMP will be

stored by WinTex.

- Go under the **Edit** menu, and select **Sprites**.
- **Choose a name** in the Entry name drop box, Choose a Phase and a Viewpoint and press **Normal View** to complete your sprite name. All the '?' characters disappear, so that it makes a valid sprite name.
- Press on **Add** and select a BMP file that will represent the sprite, for the phase and viewpoint you've chosen. The file will be copied to the SPRITES\ subdirectory, and your list will be updated.
- Press on **Save** to save the changes to list, when it has been modified.

Improve the presentation of your sprite:

- For each sprite, select an Offset and press on **Apply**.
- double click on your sprite to edit it with the BMP editor.

Building your Sprite WAD

Press **Quit**, and back in the main menu, select a name for your **WAD to build or modify**, and and press **Create**.

Playing your sprite WAD

Due to a bug in DOOM and HERETIC, before you can play your Sprite PWAD, you must complete it with all the existing sprites. While it is still selected as **WAD to build or modify** press on **Add Sprites**. Here it is, you can play your WAD.

Distributing you WAD

Before distributing your WAD, check that the complete list of sprite is not included (it would be much too huge): press on **Restore**.

But, you ask, since anyone that want to use my WAD will need to complete it with all the sprites, shall I distribute WinTex with my PWAD? Of course not!

You will distribute only **a special version of DeuTex**, called **DeuSF**, that is small, portable (versions exist for every DOOM platform, be in DOS, Unix, Linux, OS/2 or anything) and that will safely do that job.

Moreover, **you won't even need to create the batch command file** to install your file with **DeuSF**: WinTex handles that for you! In menu **Build**, select Create Install Batch, and answer no to the question about a dehacked patch, answer that your file contains sprites, and no flats. And that's it. You have an **INSTALL.BAT** file in your **Working directory**.

What to do if you re-render all your sprites?

- **Later**, if you want to change the picture of your sprites (because you redraw them, or re-generated them with 3DS) just copy the new files to the **SPRITES** subdirectory of your **Working directory**. This will spare you the trouble of re-declaring them again to WinTex, and you will understand why DeuTex and WinTex don't store the data files as WAD!

See the Sprite Editing section, for the technical stuff.

Why use WinTex with sprites?

Because DeuTex is the only tool that can change the number of available viewpoints in the main WAD. If there is a sprite which has the same right view and left view, DeuTex enables

you to change it into two viewpoints, one for the left, one for the right!
Ideal if your sprite is not symmetric! ideal to make an 8-viewpoint barrel, for instance!
And of course, you can do the inverse, reducing the number of available viewpoints, to save space in your WAD, is always possible.

Why use WinTex rather than NWT with 3DS or AutoCAD?

Typically, when you generate a set of sprites, using AutoCAD or 3D Studio, you will make a first try, build a WAD with it, try it, notice that a few things are not correct, make modifications, and rebuild a new WAD, and this many times, before you reach perfection. If you use **NWT** you will need **each time** to re-load all the set of sprites, and type the offsets of course.

If you use **WinTex**, you declare your sprites, and set their offset **once and for all**. Then, for each new set of sprite you generate you will just press on the build wad button, and that's all!

Of course, you shall generate your sprites in the Data Directory, and more precisely in the SPRITES. Else, no problem, but you'll have to make a file copy.

Using WinTex to redefine Sounds

EASY

Redefining sounds, what for?

You can change the sound effects that are played when the **monster scream or die**, when the **weapons are fired**, when the **missiles explode**, when the **door open**.... Under heretic, you can also change the **ambience sounds** (random sounds). You cannot change the music here. It is not under a sound effect format.

The Tricks

Your sounds shall preferably sample at 11025 sample/second, and imperatively 8 bit per sample without compression. Remove the silences, they waste data.

Filter your sounds to reduce the noise level.

Take care to cut the eventual noise of the recorder. those clicks sound bad in DOOM!

Don't make too big sounds. the sound compress very bad with the archivers, and your WAD will be huge. also, DOOM cannot play big sounds (max is around 10 seconds).

Using WinTex with sounds

You will first select a Working Directory. this is the place where all your sounds .WAV will be stored by WinTex.

- Go under the Edit menu, and select **Edit sounds**.

- Declare new sounds one by one by choosing a name in the sound list, Rename the sound later if you made an error. Click on **new sounds** and select your WAV file. it will be copied to the relevant place.

- double click on your sound to play it, click on edit to edit it with your own program.

- when your list of sounds is complete, press **done**, save the file, and go back to the main menu.

- In the main menu, select only the sound (for security), select a name for your WAD to be built, and and press **build**. Then you can play your wad directly.

See the [Sound Editing](#) section, for the technical stuff.

Using WinTex to redefine Musics

EASY

Redefining musics, what for?

You can change the musics that are played in background as your blast through the levels.

The music tricks

The **musics** in DOOM and HERETIC are under a special **MUS** format, that few people ever bothered to understand. you can convert your **MIDI** files to MUS, by courtesy of IDsoftware, and back from MUS to MIDI by courtesy of Joachim Erdfeld (just_joe). These are external programs, because the source was not available or too strange to be integrated in WinTex. There is **no MOD converter** know to me, hence it's not called by WinTex. Many sound cards will crash if you try to play **a music with too many simultaneous notes**. A bad example is the music of TRINITY (the GUS is meant to play techno, not J.S. Bach).

Using WinTex with musics

You will first select a Working Directory. this is the place where all your sounds .WAV will be stored by WinTex.

- Go under the Edit menu, and select **Edit musics**.
- Declare new musics one by one by choosing a name in the music list, preferably a name that corresponds to the level the music will be played in. Rename the music later if you made an error. Click on **new music** and select your MUS file. it will be copied to the relevant place. Click on **new midi** and you can choose a MIDI file, that will go through conversion by MIDI2MUS.
- double click on your music to play it, but you had better converted it to MIDI before, because Windows can play MIDI sounds, but not MUS sounds (I'm not gonna write a MUS driver).
- when your list of musics is complete, press **done**, save the file, and go back to the main menu.
- In the main menu, select only the musics (for security), select a name for your WAD to be built, and and press **build**. Then you can play your wad directly.

See the [Music Editing](#) section, for the technical stuff.

Using WinTex to redefine lumps

VERY EASY

Redefining lumps: what for?

You can change the end screen of DOOM and HERETIC, and the start screen of HERETIC.

You can change the demos of the game.

You can also replace the lumps by anything you like, but you will need another tool to edit them.

later, the possibility to edit color palettes may be added

Using winTex with lumps

See the [Lump Editing](#) section, for the technical stuff.

Mainly DeuTex will call the relevant editor. TED is used for the end and start screen, because it's a good tool and then there is no need to rewrite it.

Using WinTex to change graphics

QUITE EASY

Redefining graphics: what for?

You can change the title screens, the names in the menu, then end level screen, and the status bar item, like the faces. This is much easier than changing sprites.

Under HERETIC, changing some big picture like the title and end screen is not possible, unless you treat them as lumps. This restriction should be removed some day... Still, you can view them.

Using WinTex

You will first select a **Working Directory**. this is the place where all your graphics .BMP will be stored by WinTex.

- Go under the **Edit** menu, and select **Graphics**.
- Declare new graphics one by one by choosing a name in the graphics list, and replace the '?' by relevant characters, so as to make a valid graphic entry. Rename the graphics later if you made an error. Click on **Add** and select your BMP file. it will be copied to the relevant place.
- **double click** on your graphics to edit it with your the .BMP viewer.
- when your list of graphics is complete, press **Quit** and save the modified file.
- In the **Main window** select a name for your **WAD to build or modify**, and and press **Create**. Then you can play your wad directly.

See the Graphic Editing section, for the technical stuff.

See the informations on Offsets before modifying them.

See the Picture Tricks before editing Graphics.

Using WinTex with Levels

NOT REALLY EASY

Why use WinTex with levels?

WinTex will allow you to call your favourite level editor **with your flat and texture PWAD** if you just built one. It will also allow you to **call external node builders** (like BSP) or **reject builder** (like RMB) on your levels, if your editor doesn't do that very specialised and time consuming job

You can also **change the level names**, or **group many levels** in your WAD.

The Level Tricks

- WinTex treats **each level independantly**, so you must **provide a single-level WAD** for each level you want to add in your final WAD.
- WinTex doesn't care about the internal name of your level in the single level WAD, but only one level (the first one found) will be read if there are many.
- WinTex will declare your level according to the name you specify (by clicking on **rename**) so it's easy to move a level!

Using WinTex

Nodes and reject need to be rebuilt each time you modify the level map.

You shall ask to build the nodes before you want to play a level.

Build the reject only on your definitive version, unless you want to experiment with some special effects (like blind monsters...see the RMB doc)

See the [Level Editing](#) section, for the technical stuff.

See the section about [Level Editors](#) to know how to use WinTex with your editor.

Using WinTex to redefine wall textures

DIFFICULT

See the [Texture Editing](#) tutorial.

Redefining textures and patches, what for?

You can **change the appearance of the walls** by changing the patches (parts of wall, or complete wall) and the textures (placement of the patches on the wall).

You can **change the animated walls**, and make actual films playing in the game (very short sequences only).

The texture and patch tricks

- To make **pass-through walls**, or **semi-transparent** walls (like windows, glasses) you must only use **one single patch** in the texture, or get the mother of all Hall of Mirrors.
- You can change the animations, but then enclose your new animation frames between the start frame and end frame of the animation you modify. Don't abuse on the number of frames, or the game may crash. 4 is safe, 16 is abusive, but seems to work. Don't invert the start and end frames of an animation, or the game will lock-up.
- All the patches you declare will be put in your PWAD, if they are referenced in any textures. loading of patches is automatic.
- Combining animations is a foolish thing. Happy experimenting.

Using WinTex with patches and textures

There are two specialised sections in WinTex: one for patches, one for textures.

First compose your patches, and declare them in the patche section.

See the [Texture Editing](#) and [Patch Editing](#) sections, for the technical stuff.

See the section about [Level Editors](#) to know how to use your texture WAD with your editor.

See the [Picture Tricks](#) before editing patches.

Using WinTex to change floors and ceilings (flats)

DIFFICULT

See the [Flat Editing](#) tutorial.

Redefining flats: what for?

You can add some **custom ceilings or floors** to your levels, or **change the animations** of the floors, so that they have more or less frames.

The Flat Tricks

- There must **not** be any **transparent pixels** in a floor or a ceiling
- The **sky is not a flat**, it's a special kind of texture. See the texture editing section.
- See the [Picture Tricks](#) before editing flats.
- When you redefine an **animation for flats**, you must **enclose your own frames between** the frames that are referenced as **start** and the one that is referenced as **end** of animation cycle, in that order. Those frames are indicated in WinTex. **The game will lock-up if you invert or forget one of start or end of animation. no warning!**

Using WinTex to change flats

You will first select a Working Directory. this is the place where all your sprites .BMP will be stored by WinTex.

- Go under the **Edit** menu, and select **Flats**.
- Declare new flats one by one by choosing a name in the sprite list, and typing a name instead of the '?' characters, or select an animated flat in the drop box. Your flats can have any name, preferably one that isn't already used by the game.
- Click on **Add** and select your BMP file. it will be copied to the relevant place.
- double click on your flat to invoke the BMP editor (you can change it).
- when your list of flats is complete, press **done**, save the file, and go back to the main menu.
- In the main menu, select only the flats (for security), select a name for your WAD to be built, and press **build**.
- when your WAD is built, you can distribute it. but **before you can play it** all the sprites in DOOM or HERETIC will need to be appended to that wad. click on **add flats**

See the [Flat Editing](#) section, for the technical stuff.

See the section about [Level Editors](#) to know how to use your floor and ceiling WAD with your editor.

Setting the Transparent Color

Since DOOM and HERETIC pictures contain transparent areas, and there is no way to code transparency in BMP format, WinTex use a trick:
A special color is sacrificed: all stuff of exactly that color will appear as transparent.

Changing the transparent color

You can set this in menu **Option | Transparency**.

Default transparent color is CYAN, Red=0, Green=255, Blue=255.

Parts meant to be transparent appear as blue or white

Note that if you chose Cyan, then a modified Cyan is not considered transparent (but matched to the closest color).

That means that Red=1 Green=255 Blue=255 is not Cyan

Tutorial on Texture Editing

Authors: Avery Andrews and Drake O'Brien.

Editor: Olivier Montanuy

Creating and inserting a new wall texture is easy with WinTex! Once you've learned how to insert one new texture you'll find that it's just as easy to insert 50 - or even more...

Assumptions

For this tutorial we will assume that you have extracted Wintex to its own directory on the **C:** or **D:** drive so the DOS prompt reads, eg., **D:\WINTEX** . We'll also assume you keep your pwads in a **D:\DOOM2\WADS** directory. During the texture tutorial, you will be playing with a Wad called **ROOM.WAD** provided with WinTex. You will also need DOOM2.

Menu

- 1/ A bit of theory on textures.
- 2/ Your first steps at textures editing.
- 3/ Making new patches for your textures.
- 4/ Creating new textures.
- 5/ Placing your textures in the levels.
- 6/ The Advanced topics of texture editing.
- 7/ Viewing the Textures used in a WAD.

Tutorial on Texture: Theory

What is a Texture?

A wall texture **is not a picture**. It is a **list of wall patches**, each with a position.

The patches are the actual graphic images. The textures are built by overlaying patches, by placing them one after another in their positions, the later ones in the listing sitting on top of the earlier ones (you can see this happening in the texture display routines of some editors, eg. Deu 5.2). This 2-level approach can provide considerable economy.

Examples: the TV front in the sample room.wad is produced by combining standard Doom2 patches; the grandfather clock combines standard Doom2 patches with one new patch.

Organisation of the Textures files

Doom2 defines its textures in a single directory entry **TEXTURE1**, while Doom1 and Heretic have **TEXTURE1** and **TEXTURE2** entries. This tutorial will use Doom2 and **TEXTURE1**. Even with DOOM or HERETIC, it is recommended to edit **only TEXTURE1**.

To avoid confusion later there are two things to be kept apart:

-internal files: the texture directory entries in a Doom wad (**TEXTURE1**, and **TEXTURE2**)

-external files: the files that Wintex uses to keep track of the textures.

By default Wintex uses an [**external**] txt file named **TEXTURE1** whose contents are inserted into the [**internal**] doom-wad **TEXTURE1** directory. Other [**external**] txt filenames besides the default **Texture1** may be used for inserting texture names with their pointers (to the patches).

The external texture file need only contain the new texture names and their pointers, although it does no harm if any or all doom texture names and pointers are also included, **provided the original DOOM texture names are listed FIRST**.

When we extract a pwad containing new textures we produce a **TEXTURE1.TXT** file containing all the original doom texture names with all the new texture names appended.

Tutorial on Texture: Introduction

Decomposing (Extracting) a PWAD.

The easiest thing to do is to take a pre-existing pwad and modify a texture in it. We'll start with a little wad called **room.wad** which comes packed with this version of WinTex. Copy it to **d:\doom2\wads** and play it in order to see what's there (no exit lines, no enemies, just a little room with some stuff in it). Now stop, fire up Windoze, and start Wintex.

The first step is to make all the entries in the WAD available for editing: You will want to specify a new working directory, with a name like **room**, dedicated to the editing of room.wad, and extract all the entries there.

Start from the **Main window** of WinTex. Go to the menu bar of the main Wintex window and click **FILES**, click **SET WORKING DIR.**, highlight (click) the main directory entry 'Wintex', press the **NEW DIR.** button, type ROOM in the pop up window and click **OK**. The screen will show that you have a new subdirectory called **ROOM**. Click **OK**.

At the center of the screen, under Wad to examine, is a list box which is probably blank. To the left of this the **D:** drive, Wintex directory and **ROOM** subdirectory will be listed in drop down lists. now change directory to **DOOM2\WADS**: Double-click **D:**, the **D:** directories will drop down; double-click **DOOM2** then double-click **WADS**. All your current pwads will be displayed in the Wad to examine box.

Highlight **room.wad** by clicking it. Below this box the Working directory box will show **d:\wintex\room**. Click the Extract button.

Examining the result of extraction

When DeuTex has stopped buzzing press the **Change** button on the Working directory panel. You'll see the **ROOM** subdirectory has itself sprouted subdirectories **LEVELS**, **LUMPS**, **PATCHES** and **TEXTURES**. These are the subdirectories where Wintex has stashed the respective informations contained in the pwad.

Now click on the Infos button. A file called **WADINFO.TXT** will be displayed. This is the file that tells DeuTex how to integrate the information that can be found in all the subdirectories. It will contain a section headed **[LEVELS]** and another headed **[TEXTURE1]**. You don't need to know exactly how to edit that file, because WinTex handles that for you. But you can read the DeuTex manual to learn more about it.

If you go to windows file manager you will see a .bmp file in the D:\WINTEX\ROOM\PATCHES directory, this being the new patch we just extracted. You will see a TEXTURE1.TXT file in the D:\WINTEX\ROOM\TEXTURES directory.

Modifying a Texture

Make sure you are in the working directory **D:\WINTEX\ROOM**. Go to the menu bar, click **Edit** and select **Texture1**. The Texture Window will appear. At the lower-left corner of the screen is a white window-let holding the name **TEXTURE1**. This is the name of the texture file that Wintex is currently editing. At the moment this file (**TEXTURE1.TXT**) is contained in the **TEXTURES** subdirectory of the **ROOM** directory (as first determined by the working directory specified in the main window).

The bigger window over the window-let will list all of the texture files involved in the current project, but it should now contain only **TEXTURE1**. You will seldom bother playing with many texture files, but as this was an option of DeuTex, it had to be in WinTex.

To the right of the window-let is an **Edit** button which we don't need, since the texture file is already read, and to the right of that is a scrollbox labelled **Textures in File** which lists all the textures defined in the current texture file, **TEXTURE1.TXT**.

Since you've just extracted the entries of **ROOM.WAD**, this will contain all the regular Doom(2)

textures, plus, at the end, the new textures defined in the **ROOM\TEXTURES\TEXTURE1.TXT**.

***Scholarly note:** whenever pwad redefines any textures, it has to contain the definitions of all the old textures in DOOM. This is not additive, maybe because some textures could redefine old textures. But DeuTex knows that it must include the old textures before the new ones, so that you only need to define your new textures.*

Now suppose you want to include textures defined in another WAD. For instance, you have also extracted **TRINITY.WAD** to the **TRINITY** subdirectory. Then press the **Read File** button, select the file **TRINITY\TEXTURES\TEXTURE1.TXT** and all the textures there will be appended to your texture list. This is especially convenient if you've already generated textures as text files with NWT, and want to import them in a single texture file.

Suppose you have simply renamed the **TEXTURE1.TXT** file extracted from room.wad as **ROOMPACH.TXT**. Setting **WINTEX\ROOM** as the working directory you will *not* get an immediate read-out of **ROOMPACH.TXT** when you pass to the **Texture Window**. In this case type **ROOMPACH** in the bottom window-let and press the **Edir File** button to get the read-out. But if you have also declared (as eventually you must) **ROOMPACH** in **WADINFO.TXT** (main window, working directory, Infos button) you will get the immediate read-out when entering the **Texture Window**. All txt files listed under the heading **[TEXTURE1]** in **WADINFO.TXT** will be listed in the bigger box above the window-let. If you highlight any texture name in the read-out list box and press the **Delete** button, it is deleted from the list, and if you now press the **Save** button it is really gone (deleted from the file). By using the **Read File** button you can browse through all the Wintex subdirectories for (variously named) texture files.

You can navigate this read-out in the usual manner, so whip down to the bottom where the new textures are and double-click on the texture MYCLOCK. Two things will happen.

- the texture will appear at the top left of the window, together with some info about its size.
- four names will appear in the **Patches in Texture** panel in the center of the window. These are the patches used to define the texture.

The **Patches in Texture** panel is the basic texture building tool. If you click on a patch to select it you can drag it around with the arrow keys. Two numbers, on top of the texture, that change when you move the patch indicate the coordinates of the patch. The origin of the coordinate system is the TOP-LEFT corner of the texture.

Rebuilding a modified PWAD

Now you want to Rebuild your modified PWAD.

Hit the **Change** button in the Wad to build or Modify panel and type a new name, **D:\DOOM2\WADS\MYROOM.WAD**. When this is done hit the Create button. DeuTex will buzz for awhile creating the new wad. Now play it to see the damage you've done to the clock.

Unless your changes really were improvements, clear out the room directory and re-extract the original room.wad into it.

Tutorial on Texture: Adding new Patches

Preparations before creating a patch picture

To make a patch you need some sort of paint program. Windows Paintbrush will do for illustrative purposes though for serious work you'll need something more substantial. DeuTex can handle **bmp**, **gif** or **ppm**, and WinTex can extract pictures to these formats (see the Option menu), but actually WinTex can only display **8-bit bmp**, the default Windows format. This is for reasons of speed, since gif needs decoding, and ppm has it's lines inverted.

First ensure, in the Option menu, by selecting Transparency, that your color for transparency is **CYAN** (the default).

Press **Default** in the Transparent color editing window.

The cyan color will be interpreted as transparent in all your pictures. This is because there is no special way to code transparency, in general format of pictures. Note that pale blue or cyan never appears in DOOM, that's why since the days of DMGRAPH cyan is used as transparent color.

Not also that the transparent color must be Exactly Cyan (R=0 G=255 B=255). Some lame editors only allow colors on 18 bit rather than 24, and you'll have to use R=0 G=252 B=252 for Cyan.

Creating a Patch picture

Suppose we want to deface the clock with a tag. This can be done by creating a **64x64** picture, the tag scrawled in any colors on a **CYAN** background.

Start up Windoze Paintbrush and under the menu **Options | Image Attributes**, click the radio button **PELS** (for pixels) and type in **64x64**. Create a cyan background by double-clicking the light blue picture block at bottom of the window. The **Edit Colors** dialog box appears and it should have Red=**0**, Green=**255**, Blue=**255**, which is pure **CYAN**. Click **OK** and roller the cyan background on.

Now create a masterpiece and save it to your **C:\WINDOWS** directory as a **256-COLOR** scrawl.bmp.

You will get more info about graphics in the Picture tricks section.

Declaring a new Patch to WinTex

Now (re)start Wintex. Make sure your Working directory is still **ROOM** and go to menu **Edit**, select **Patches**. In the Edit patches window hit the **Add** button and locate your new graphic in **C:\WINDOWS** with the resulting dialog. Highlight it and click **Ok**.

SCRAWL.BMP will appear in the **New Patches** box, which simply lists the contents of the **PATCHES** subdirectory of your working directory (**ROOM**).

Now, your patch is added to the list of new patches, and it can be reused during texture editing.

Click **Quit** to exit the **Edit Patches Window** window for the **Main Window**. Go now to the Texture Window (menu **Edit**, select **Texture1**). **SCRAWL.BMP** will appear in the **New Patches** list which of course displays the same content as the new patch list in the **Edit Patches Window**, the content of the **ROOM/PATCHES** directory.

The you can click on it and put it on your textures.

What you have done with **SCRAWL.BMP** is *copy* the file from **C:\WINDOWS** to the **D:**

WINTEX\ROOM\PATCHES. In fact you could have simply saved from Windoze Paintbrush to this directory and it would have worked the same, since patches need not be declared to WinTex in advance. But then you wouldn't have learned how to use the **Edit Patches** window of Wintex - a tool which will be useful in future.

In fact here is a safe good way to do it: create a new Wintex subdirectory **WINTEX\PACHTEMP**. Save your bmp's from your graphics program to the **PACHTEMP** directory with a temp name, eg. **TSCRAWL.BMP**, fire up **PaintShopPro**, quantize the **PACHTEMP** files and save it to your pwad subdirectories (in this case ROOM) with their final patch name. This way you keep your quantized/unquantized bmps separate, which is a wise thing to do, should the quantisation be incorrect.

Patch editing in WinTex

While in the **Texture Editing** window, you cannot edit any patch in the New patch list, because the patches are cached to speed texture composition, and as changes to a patch would not be taken into account it's not wise to let you edit it. For the same reason, you cannot edit the patches inside the main WAD.

But should you want to change those patches, go to the Patches Editing window highlight the DOOM patch you want to modify and press the Import button. this will copy that patch from the main WAD into the new patches list, and there you can edit it as you wish. But of you really want it to replace an existing patch, don't ever change it's name.

Adding your patch to the Texture

You are now in the **Texture Window** where **SCRAWL** has been added to the new patch list. Run down to the bottom of the textures list (bottom center of the screen), find **myclock** & bring it up, then select **scrawl**. Initially, it appears with the cyan background (this is an uncorrectable bug, as Visual Basic can't move transparent bitmaps easily), but when you left-click your patch is placed on texture and the cyan background disappears.

By pressing on the up/dn/lft/rgt buttons drag it precisely to a preferred position, press **Save**, **Save File** and **Quit** texture editing.

Now re-**Create** the wad (as **MYROOM.WAD**) and play it to admire your handiwork.

Note: you can recompile the wad with the identical name **ROOM.WAD**, and WinTex will ask you if yu want to delete the existing **ROOM.WAD** to save you possible heartache since overwriting is irreversible. I like to have at least one backup!

One thing you'll see is that it doesn't really look like vandalism of the front of the clock - the actual Doom textures are very cleverly chosen to given an impression of realism within the limitations of the engine - quickie custom patches tend to look like a cheap import from a cartoon, not like actual components of the Doom environment. If you want to make stuff look good, you'll have to learn a bit about palettes & color quantization.

The Viewer

A quick way to find out what each Doom patch looks like is to use the Wintex Viewer.

In the **Texture Window** if you click on the doom texture name **STARTAN3** the picture will be called up with its patch names **SW19_1** and **SW19_2**. You can find out more about the SW series by following these steps:

-- in the main window **Wad to examine** box call up **D:\DOOM2\DOOM2.WAD**

-- in the main window menu click **Show, Patches**, then type 'SW' in the **View Picture** window and you will jump to **SW19-1**. Highlight it to view. Highlight each name of the **SW** series to note the similarities. Very quickly the doom patch names will become more familiar. Check out the other viewers, like sprites and flats. They are mainly meant to display stuff that can be found in PWADs, but can also be used to become more familiar with the main

WAD.

Tutorial on Texture: Creating new textures

The next step is to create a new texture with or without new patches.

Examine the doom textures. The standard size for a doom wall texture is **128x128** pixels. The maximum height is **128**, because the DOOM engine repeats the textures vertically, with period **128**. The maximum width is **1024**, maybe more, but you seldom need more than 256.

The one exception to the rule is the famous Deutex wrap-around sky, first used in TRINITY.WAD, which is 1024x128, composed of four 256x128 patches in DOOM, and four 256x245 patches in HERETIC. The wrap-around sky is reason enough for learning Wintex! WinTex won't let you choose a texture size that is not suitable for the DOOM engine.

Let's start anew with **MYWAD.WAD**, which yet has no new texture names. Create a new working directory called **MYWAD**, and extract **MYWAD.WAD** to it. This will create a **LEVELS** subdirectory and a **WADINFO.TXT** declaring the maps in it.

Go to the **Edit Patches** window and click **Add**, and in the **ROOM\PATCHES** directory select your old **SCRAWL.BMP** and click **OK**. Likewise for **CLOCK.BMP**.

Go to the **Texture Editing** Window and in the **Texture** box click **New**.

The default **128x128** cyan texture space will appear and you will be prompted for a texture name, of 8 character. type **NEWTEXT**.

Always make sure this name is original, in the sense that it does not conflict with already existing texture names. Note that the name appears in a drop box, on the right of the **New** button.

You could also directly select a name here, especially if you want to redefine animated textures, since the name of the animated textures are defined here (or more precisely, the name of the extremities of the animation cycle).

Compose a texture from the patches listed in **New Patches** and **Patches in main WAD**, as before. When finished click the **Save** button in the picture box and **NEWTEXT 128 128** will appear in the **Texture in file** box. Click on **Save File** to save your work, this will create a **YOURWAD\TEXTURES\TEXTURE1.TXT** file, since **TEXTURE1** is the default name for that file. Click **Quit** to leave the **Texture editing** Window, and click Infos in the **Main window** to take a look at the **WADINFO.TXT** file. You will find a section like:

```
[TEXTURE1]
TEXTURE1
```

This section tells WinTex that when the WAD is to be built, to create the **TEXTURE1** entry (the first list of textures) it will have to read a file called **TEXTURES\TEXTURE1.TXT**. There could be many texture file names listed here (without the **.TXT** extension).

Now change the **Wad to Build or Modify** to **D:\DOOM2\WADS\MYWAD.WAD** (I assume you already have a backup since it's your current wad) and press **Create**. When Deutex has finished the resulting wad will have a new texture name, **NEWTEXT**, pointing to the patches you used, and when you load it in your editor, if your editor can load new texture names, you will find **NEWTEXT** added to the list of textures.

Having created one new texture you can now create as many as you want, simply adding them to the list.

Tutorial on Texture: Editing the levels

(this part of the tutorial was highly edited by the WinTex author. Sorry)

At this writing I'm sure that DoomCAD 5.1, DEU 5.21, WINDEU 5.23, DCK 2.1 and DMapEdit 4b7, will load new texture names. Edmap1.4 and latest DoomEd will also have this feature.

Editing levels with new textures ought to be very simple, but right now it isn't, because at the moment, no editor known to us handles custom textures correctly when they are defined in the same wad as the maps - many will read them, but none will save them properly! But maybe this has already changed by the time you are reading this. So if you're using an editor dating from later than March 10, 1995, try the following experiment: copy room.wad to a different filename, say, room2.wad, and then use your map editor to make a minor change to room2.wad, save it, and try to play it. If it works, you don't have to read the rest of this section (though you might want to, if your textures begin taking up a great deal of space in your wad). If doom crashes, read on.

WinTex in fact was designed to work with editors like DEU, that accept multiple WAD as input.

The idea was that first you would select your new patches, new textures (and possibly new flats), and you would create a temporary WAD containing only those new textures (and Flats).

Then WinTex would manage to call your level editor so as to let it know about those new textures (and flats), and you can freely place them to one or more of your levels.

Then, and only then, would you build your complete WAD, with textures (and flats) and levels.

You can use WinTex in an other way, but I think it would be quite inconvenient.

Favoured method for level editing

- First, Create your patches and your textures. You know how to do, now, by using the **Patch Editing** and **Texture Editing** windows.

- Create a WAD containing only your textures and patches. Press Change in **Wad to Build or Modify** and select for instance the name **ROOMFX.WAD**. This will be your temporary WAD. In the **Select Types** panel, press **Tex+Flat**, and then press **Create** as usual. Since you only selected Textures (and Flats), only those are added to the WAD you have created. Check its contents to verify that it doesn't contain any map information. (select it as the wad to examine, and hit the **Contents** button: there is no MAP inside.) Now it is time to edit the map.

- Go to the menu **Edit**, select **Levels**. You invoke the Level Editing window, a windows dedicated to the composition of WAD containing one single level and nothing else. Those WADs, that are stored in the **LEVELS** subdirectory of your Working Directory, will later be used by WinTex when you will create your final WAD.

- Press on **Add** and select a WAD that contains the level (let's say **MYHOME.WAD**) in which you intend to add your textures. This level file will be copied to the **LEVELS** subdirectory of your working directory, and will appear in the list of levels, as: **MAP01 = MYHOME** (This means that when your final WAD will be build, the level contained in **LEVELS\MYHOME.WAD** will be used to create the levels called MAP01, which is mission 1. If you mean it for another mission, select another mission in the level names drop box and press the **Rename** button).

- Last, highlight the **MAP01 = MYHOME** line and press on the button corresponding to your favourite level editor, for instance, WinDEU. You will be asked to point to WinDEU.EXE is situated, as WinTex can hardly be expected to guess where you've put WinDEU.

Then, and this is the great power of WinTex, WinDEU will be called with three arguments:

- The name of the main WAD (because you could be using DOOM, HERETIC, it depends)
- The WAD **MYHOME.WAD** because it's the level file.
- The WAD **ROOMFX.WAD** because it exists since you just built it, so WinTex assumes it may contain interesting stuff (and indeed it contains new textures).

So normally your level editor is called with all the info it needs to allow seamless editing with new texture, and when you save, you save to the single-level WAD **MYHOME.WAD**. It works like a dream with WinDEU.

To test your level, just type **DOOM -file ROOMFX.WAD MYHOME.WAD**.

Of course, you could also build your Wad with texture and levels, and edit that complete wad, but since most level editors can't save properly to a WAD when there is already something in the WAD (here, the texture) you're bound to be in trouble rapidly.

Hacker's method for use with deficient level editors

The beginning of procedure is the same as before: create your patches and textures, and create a WAD **ROOMFX.WAD** that contains only Textures and Patches.

Then, in Wad To Build or Modify select anything but the WAD **ROOMFX.WAD**. Press **Change** and type DEFAULT.WAD for instance.

Select ROOMFX.WAD in the **Wad to examine** box, clear the check box called **Protect your game files** and ... wait a second: you're about to modify your main WAD. If you don't trust the way DeuTex was written, you might wish to stop here. If you have enough disk space (a bit more than the size of your **ROOMFX.WAD**) and trust that DeuTex will only make a minor modification to your main WAD, then press **Merge** (or menu **Merge**, select **Merge PWAD in main WAD**).

DeuTex will run for a little while. What happens is somewhat hard to believe: by modifying only 8 bits at the start of your WAD, DeuTex will append your **ROOMFX.WAD** to the main WAD, and will rebuild a WAD directory in such a way that DOOM, or any level editor, will believe your entries in **ROOMFX.WAD** have always been part of the main WAD.

Well, actually, nothing real new: old DMGRAPH did that...in a dirty way. What is new is that the changes are minimal, so minimal that they are in fact reversible (press on Restore, on the right of your Doom directory, and your main WAD is back to normal, as if it had never been modified, even the time stamps on the file are back to normal). Now you know what DeuTex is worth. And you know why NWT 1.02a has copied that merge feature!

Of course, the operation is only reversible if you don't modify your main WAD with another tool, that could spoil everything. Expect amazing features, but don't expect miracles from DeuTex.

So now that your Texture WAD is merged into the IWAD? you can call any level editor, be it the most carelessly written, and it should be able to edit your WAD. Be warned however than some editors create pre-calculation tables to speed their work, and you may need to re-generate those tables once your texture WAD is inside. Sorry for the mess, but I'm not responsible for the behaviour of the level editors.

To test your level, just type **DOOM -file MYHOME.WAD**.

Since your ROOMFX.WAD is already inside DOOM2.WAD, it's useless to put it after -file.

Creating your final WAD

Once you have finished the level editing business, and your level has been correctly declare in the **Level Editing** windows (by typing **Add**, and selecting it, and selecting a mission name for it) you will want to build your final WAD.

Then go back to the **Main Window** and in the **Select Type** panel press **All** and then press

the **Create** button. of course, you will first have chosen a suitable name for the **WAD to create or modify**.

Another method is to simply select **ROOMFX.WAD** as the **WAD to create or modify**, select **MYHOME.WAD** as the **Wad to examine**, and press the **Join** button.

You will then merge **MYHOME.WAD** into **ROOMFX.WAD** and then you can distribute **ROOMFX.WAD**.

This method is to be preferred when you want to distribute your work as a big WAD, but want to give the user the option to split it in one or more WAD later. I don't think it's very adapted to Texture and levels, because a texture makes little sense without it's level, but it is very adapted if you have replaced many sprites, and want to let the users be able to actually use only some of them.

Tutorial on Texture:Advanced topics

Textures for transparent walls

As is well-known, transparent textures (walk thru' walls) must be composed of only one patch to avoid the medusa effect.

Avoiding errors due to columns without patches

It is a rule in DOOM that every column must be covered by at least one patch, even if that part of the patch is entirely transparent. Failure to respect this rule will cause a game crash at boot up or only a tutti frutti effect if you're lucky. Of course, WinTex will report that kind of error when you will check your WAD.

If you are composing a non-transparent texture the tutti-frutti effect can be caused by improper alignment of a background patch. A simple example: a **128x128** texture composed of only one 128x128 patch whose upper left corner is pointed at the coordinate **1,0**. This produces a void column and the tutti-frutti effect (i.e. doom fills up the missing pixels with garbage).

The coordinate numbers in the picture window are of course helpful, but to be absolutely safe I always directly check the pointers in texture1.txt. In our example, if the texture name is **PUNKBASE** and the patch name is **PUNK** there should be an entry in **TEXTURE1.TXT**:

```
PUNKBASE 128 128  
* PUNK 0 0
```

The first line enters the name and size of the texture, the second line enters the name and top-left co-ordinates of the first (and only) patch.

To see how it works try dragging the clock-face around the clock. Highlight the different patch names and press the **Bring in Front** button.

You can also put new patches into the texture by selecting from the two menus at the far right:

- left-click on the patch name to select (or type the beginning of the name of the patch you're looking for, if it's too deep in the list),
- move the cursor to the picture window.
- left-click to place the patch into the texture.
- right-click to discard the patch.

The senseless names and apparent lack of organization in the list of doom patches is unfortunate, but there's nothing we can do about it: these are the names and order the DOOM conceptor used.

When you've finished messing up the clock hit the **Save** button (in the lower left picture window region), which saves the texture to the file, then hit the **Save File** button beside the **Textures in File** scroll box, then hit **Quit**.

A Warp Around Sky

Despite what one could expect, the sky in DOOM and HERETIC is a texture, not a Floor or

Ceiling. But it's a special texture that cannot be animated and is displayed reversed, as seen in a mirror image. When you will edit such a texture with WinTex, keep in mind that what you'll see on the sky will be reverted.

Replacing a sky texture requires re-using a Doom sky texture name. We'll reuse **SKY1**. Compose four continuous patches **256x128** pixels each (don't use cyan! but you can use some pale blue). The horizontal texture coordinate correspond to cardinal points: 0=east, 256=north, 512=west, 768=south (1024 is north again, as the sky is warp around).

In the **Texture window** in the **Texture** box select **SKY1** from the **texture name** drop box. Place the EAST patch at position 0,0, the NORTH at 256,0, WEST at 512,0, and SOUTH at 768,0. You will have to use the horizontal scroll bar at top of the picture box because the viewing area is only **512x128** pixels. Hit the **Save** button and the sky is done.

In TEXTURE1.TXT the entry should look like this:

```
SKY1      1024 128  
* EAST      0 0  
* NORTH     256 0  
* WEST      512 0  
* SOUTH     768 0
```

If the numbers aren't exactly these, change them. Of course, the patch names you use can be a bit more creative. If your level is part of a series in a multi-level pwad then reusing the Doom sky texture name of course means that your new sky texture will be pasted on all levels of the pwad, and when you pass from the pwad levels to the original Doom levels your sky texture comes with you.

Note, if you're under Heretic, the procedure is the same, but the patches must be bitmaps of size **256x245**. However, the texture will still be declared as **1024x128**.

Animated Textures

New animated textures require re-using the Doom animated texture names, eg. **BFALL1**, **SFALL1**, **FIREBLU1**. Our example will use the **FIREBLU1>2** sequence (the 'cold fire' texture).

We will use original Doom patches: for the background **WALL40_1**, and for the animation the eight patches in the **VILEA1D1>VILEA8D2** sequence.

Go to the **Texture editing** window, click inside the **Patches in Main Wad** box and type **WALL40**

and that should bring you WALL40_1. In the **Texture** box use the Texture name drop box to select FIREBLU1 (press the arrow, a list drops down, find FIREBLU1 which is reported as an **Animation Start**). Press on New, and leave the texture size at the default 128x128.

Click WALL40_1 in the **Patches in Main Wad** box and center it on the texture at (0,0).

Click again in the **Patches in Main Wad** box and type VILEA1, you should get the **VILEA1D1** sprite, that will appear on your texture. Yes, sprites can be used on Texture, this is known since TRINITY.WAD was out.

Put that sprite at approximately (48,28). Select '?????' as the texture name and click New, type in the texture name EXAMPLE1 in the place of the ?????, select once again WALL40_1, place it as usual, select VILEA2D8, center it (32,28), repeat the procedure for EXAMPLE2...and so on until you've done with texture EXAMPLE6 using VILEA7D3.

For the last patch with VILEA8D2 use the name FIREBLU2 that you will find in the drop box, with the indication **Animation End**.. Save it all out and build your wad. Now when you use the texture FIREBLU1 you'll be using your own animation of the archvile forever circling on the wooden wall.

The entry in D:\WINTEX\ROOM\TEXTURES\TEXTURE1.TXT should look something like this:

```
FIREBLU1 128 128
* WALL40_1 0 0
* VILEA1D1 48 28
EXAMPLE1 128 128
* WALL40_1 0 0
* VILEA2D8 32 28
EXAMPLE2 128 128
* WALL40_1 0 0
* VILEA3D7 24 28
EXAMPLE3 128 128
* WALL40_1 0 0
* VILEA4D6 34 35
EXAMPLE4 128 128
* WALL40_1 0 0
* VILEA5D5 50 31
EXAMPLE5 128 128
* WALL40_1 0 0
* VILEA6D4 34 33
EXAMPLE6 128 128
* WALL40_1 0 0
* VILEA7D3 24 34
FIREBLU2 128 128
* WALL40_1 0 0
* VILEA8D2 32 28
```

Now you have the idea, which is to sandwich a sequence between an Animation Start and an Animation End, in that order. Note that it must be the same animation (FIREBLU here), and the Start must be placed before the End. Otherwise, you crash DOOM. Simple.

For more detailed information about animated walls read the **Unofficial DOOM Specs** by **Matt Fell**.

Tutorial on Floor and Ceiling Editing

A natural flow-on from custom Texture is custom Floors and Ceilings, also known as Flats. These can be done with the **Flat Editing** windows, **Patches Editing**.

Introduction to Flat editing

First become familiar with the flats in DOOM by clicking in the dark grey box that list the flats contained in the main WAD. Note that you can type a name in that box, and the patch with that name will be highlighted.

Note also that sometime, when you select a flat that is related to an animation, it's name will appear in the Flat identification drop box, together with the indication of whether it is an animation start or an animation end (or, under HERETIC, if it makes water splash)

Now let's take a look at the way WinTex stores Flats:

In the **Main Window**, select the file **ROOM.WAD** provided with WinTex as the **Wad To Build or Modify**, select a new **Working Directory** and press Extract.

Once DeuTex as finished running, in menu **Edit**, select **Flats**, and the **Flats Editing windows** appear.

There you will find the flat **BTHFLR** in the list:

BTHFLR = **BTHFLR**

Double click on it to edit it if you want.

The first name means the entry will appear in the WAD as **BTHFLR**, and the second entry, after '=', means that the file that is used for the **BTHFLR** entry is in your **Working Directory** as **FLATS\BTHFLR.BMP**.

If that second entry was, say, BTHCEIL, the file would be **FLATS\BTHCEIL.BMP**, but the entry would still be included in the WAD as BTHFLR. This is to let you use your own names for files.

Of course, with Flats, you will only use that feature when redefining animations, since flats can have any name that doesn't conflict with other names.

Declaring your Flat to WinTex

You will add your flats one by one: First select a name for your flat in the flat identification drop box, by typing letters in place of the '?' signs, or by selecting an already existing name. For instance type **WOODEN**.

Then pressing **Add** and pointing to the relevant BMP file, for instance **C:\OAK.BMP**.

This tells WinTex that when building your Wad it will use that **OAK.BMP** file to create the flat called **WOODEN**. Of course, **OAK.BMP** should be a **64x64** bitmap.

See the [Picture Tricks](#) before you start composing a flat.

Now that your flat is declared to WinTex, **click** on it to display it, **double click** to edit it.

Creating your PWAD with Flat

Press **Save** to take your modifications into account, Press **Quit** and in the **Main Window** select the flats in the **Entry Selection** box, select a WAD **D:\DOOM2\WAD\ROOMFX.WAD** as **Wad to build or modify** and press **Create**.

You should obtain a **ROOMFX.WAD** that will contain only one entry, a Flat named **WOODEN**, but in fact there will be three entries, as can be seen by selecting ROOMFX.WAD and pressing **Contents**. You get:

```
FF_START
WOODEN
FF_END
```

In fact **FF_START** and **FF_END** are special markers, that make no sense to DOOM, but that WinTex uses to be sure that the entries between them are flats.

Completing your Flat PWAD by the list of all Flats

As it is, your WAD is not usable. Before you try to use it, press on Add Flats. This will complete it with the list of all flats in DOOM as you can see by pressing **Contents**. Note also that when all flats are there, the F_START marker is used instead of FF_START. This F_START is recognised by DOOM and it means...all flats are there.

DOOM really needs all the flats to be in the same WAD. It's a misfeature of the game, that was not voluntary, but could not be corrected.

However, as flat completion takes time (and waste a lot of space on your disk), for development purpose it's better to rely on Merge: Select your PWAD as the **WAD to examine**, and press Merge.

Distributing and Installing your WAD on other computers

So completing flats is what you will do when you will let other people install your WAD on their system, once it's finished. Of course, you will use DeuSF for that, not WinTex or DeuTex. DeuSF is much smaller, and can be freely distributed.

To spare you some boring work, WinTex is able to generate the batch file that will call **DeuSF** to do the job you did by pressing Add Flats. Select your WAD as the WAD to build or modify (well, it should be selected already, shouldn't it?) and in menu Build select Make Install Batch.

Reply no to 'use exe patch' because you don't have to install a Dehacked or HHE patch. Reply that you don't have sprites, but you have flats, and presto WinTex will put the right batch file, called **INSTALL.BAT**, in your Working directory.

Warning: Distribute your WAD only when it contains the flats you added (press Restore to revert to your original WAD). **DO NOT distribute pwads with all the regular flats included**, this is illegal and expensive in download time and money.

Tutorial on Sprite Editing

(Author: Avery Andrew)

Our tutorial project will be to modify an `obstacle' to be placed in the room wad. So set the **Working directory** to the one you've set up for roomfx in the flats tutorial, and in menu **Edit** select **Sprites** from the main menu.

The Sprites Editing window will then appear, which is the most complicated we have seen yet, due to the inherent complexity of the sprites.

It will include a big blue window towards the top, as well as a large white panel underneath, which should be empty, as well as lots of other stuff. The blue box is for displaying the graphics, and the white box is a list of the sprites that your project is redefining. There should also be a gray listbox, with **CHGGAO** at the top. This is the list of sprites defined in the **Main WAD**.

To see the graphic of a sprite, highlight and click on it, and the picture will appear in the blue window. Scroll through the list, browsing among the pictures.

There is some method amid the chaos of the names. The first four letters of the name, say CHGG, indicate what kind thing the sprite is a graphic of, in this case, a chaingun. There is sometimes some evident rationale for these names, other times, not. The pink demons, for example, are **SARG** (maybe there is evidence about the prehistory of the game hidden in these names ...).

Next comes a single letter for the animation Phase. Many static objects have only one animation phase, A. The chaingun has two, A and B, the archvile (VILE), has 29 (using punctuation marks for the last three, in the WinTex interface).

Then there is the animation Viewpoints number between 0 and 8 for the viewpoint. 0 means the same graphic is used from all viewpoints. But if any from 1 thru 8 are provided, they all must be, and will override 0, if it is also present. Finally, it is optional to have another number and letter, with the same meaning, except that the sprite is displayed in reverse, so you don't need to include graphics that are mirror images of each other.

To define a graphic for a sprite, go to the drop-box at the very bottom of the screen, and scan it to see if the one you want is there (most of the enemies and a few other things are, but most of the sprites are not). If you can't find what you want, click on **?????????? - any sprite**, so that this appears in the box's window. Then type into the window the full name of the sprite you want to replace, which in our case is the `technical column', **ELECAO** (be sure to end with two spaces, to erase the last two question-marks).

Now click on the `new' button hanging off the white panel. A file-selection dialog appears, which wants you to find a **BMP** image to use for the sprite. Anything will do, say, the scrawl you made in the textures tute, or the clockface in the **PATCHES** subirectory of the **ROOMFX** directory.

Pick a bitmap, and okay the dialog. A line will appear in white panel such as:

```
ELECAO 0 0 = CLOCK
```

ELECAO is the name of the sprite we're working on, the two numbers are offsets, to be discussed in a moment, and **CLOCK** is the file-name for the graphic.

The offsets indicate how the sprite is positioned. There seem to be various schemes, depending on what kind of sprite it is. For monsters, players and items and obstacles (lamps, trees), the x offset indicates what column occupies the (x,y) position of the sprite in the map, while the y offset indicates what row occupies the floor (so the x offsets tend to be approximately half the width, and the y offsets not too far off the height of the graphics). For projectiles + explosions, the offsets seem to identify the pixel located at the center of the

missile/event. For player-weapons, noone knows exactly what the offsets mean, so you'll have to experiment to meddle with these.

For quick setting of the offsets, there are four buttons, **walking**, **creature**, **missile**, **explosion**, **weapon**, **large**, and **weapon**, **small**. Hit **walking**. Crosshairs will appear in the display, and you can then use the slidebars to adjust the position of the point that will occupy the ground-level position of the sprite we're making (since it's an obstacle). When it looks the way you want it (do you want your clockface floating in above the ground, or sitting on the ground??), hit **Apply**, and the new offsets will appear in the white panel. Now hit `save' and `done', and you will be returned to the main window.

Now rebuild **ROOMFX.WAD** and merge it with the iwad, and fire up your editor. When you edit things, and select the `technical column' (standardly the tall thing with wiring, & pipes in it), instead you'll get the clockface. Stick it in a wad, & play the wad. When you're done, return to WinTex and **Restore** the **Main WAD**.

See the rest of the Help for further info on sprite construction.

The picture tricks: Help on making graphics

(Authors: Olivier Montanuy and Avery Andrews, with parts from Kevin McGrail)

Those are general tricks to use when creating Graphics for DOOM or HERETIC:

Reduction of picture colors (quantisation)

Most of the time, when you work on pictures meant for DOOM, you will work on RGB pictures (24 bit mode). But unfortunately DOOM only supports 256 colors (8 bit color), and even worse: only the 256 colors of a special DOOM palette. It's the same for HERETIC, but the palette is different.

So you must fit the colors in your pictures down to that special palette.

Of course, the first step is to always use colors that are close to the DOOM or HERETIC palette.

Here is how to fit your picture to the DOOM palette, using the cool Shareware PaintShop Pro 2.0.

- Load a reference picture file, extracted from the main WAD.
 - Select menu **Colors | Save Palette** and save the palette to a file.
 - Load the file you want to convert.
 - Select Menu **Colors | Load Palette**, browse to find the palette, select it.
 - In the 'apply palette to image using' box, possibly select Error diffusion. Select Ok.
- Then save your file as **BMP - OS/2 or Windows Bitmap, Windows RGB Encoded**

DOOM will now display your Graphic pretty much as you expect. Experiment with this first, as your program might not do what you expect it to. For example ZSoft's PC Paintbrush will load the DOOM Graphic correctly and allow you to save the palette. However, if you take a 24 bit image and load the palette, nothing will happen. You can let PC Paintbrush convert the image to 256 colors (which it will do poorly), and then apply the palette. The results will be grotesque, because it doesn't quantize when applying a palette. A DOOM palette saved in this program would be useful only for Graphics you want to hand-draw from scratch. The moral is, if you don't know what you're doing, experiment first before using a Graphic you don't want to lose.

Note that DeuTex supports **24-bit** bitmap (millions of colors), but since color matching is slow as hell, it is not supported in WinTex. 16-colors bitmap are not supported either, to spare you the ridicule of feeding a 16-color picture in DOOM. Only 256-colors is supported.

Smoothing the pictures

The DOOM engine use a pervert technique when scaling down a picture (for instance, a monster seen from very far away): it suppresses lines and columns. Do that, and at best, you end-up with a terrible looking pictures. At works, you are a victim of aliasing, the same phenomenon that happens when a sound is under-sampled: high frequencies and low frequencies mix up and the result can be painful to the eye (or the ear).

There is only one defense against aliasing: cutting the high frequencies, which, put in graphic terms, means smoothing your picture.

After you created a picture of a monster or a wall patch or a floor, you'd better apply one of those smooth or blur filters found in all graphic package. Be warned that they may only work with 24 bit RGB picture, and thus you will need to quantise your color after smoothing. Paint Shop Pro can do all that: convert your picture to 24bit, smooth it, load the DOOM palette and you've got a valid bitmap again, but much more suitable for DOOM this time.

Avoiding regular patterns

You will want to avoid all repeating or regular patterns in your wall patches or your floors. The reason is always the same: aliasing. A repeating pattern, especially if the pattern is short, corresponds to rather high frequencies, and when it's scaled down the resulting mess is terrible.

Good example of that mess are the yellow and black stripes in Aliens-TC, and the original floor of Alien-TC.

To avoid trouble, you will never use grids or meshes, unless they have a very irregular pattern. A grid of step 64 pixel will generate little aliasing, except at very far distance. The rule is: the further your pattern shall be seen from, the least regular is shall be.

