WadAuthor

WadAuthor Copyright © 1995-2001
Williston Consulting

Compiled By Richard D. Clark

Table of Contents

About WadAuthor 6
What's New 11
  Overview....12
  Bug Fixes 12
  Improved Features 12
  New Features 12
  Overview....13
  Bug Fixes 13
Basic Vocabulary....14
Understanding Wad Files....15
  Header....15
  Resources 16
  Directory....16
Understanding Wad Maps....16
  Resources 16
  Geometry....18
Creating New Objects....19
  Creating a New Thing 19
Selecting Objects 20
Moving Objects....20
Editing Existing Objects....21
Deleting Objects 22
  Cascade Effect....22
  Corruption....22
Making Things Happen....23
  Sector Effects....23
  Simple Linedef Effects....23
  Complex Linedef Effects....23
Understanding Polyobjects....24
What Is A Script....24
Why Are Scripts Compiled....25
Where is the Data Stored 25
How Do I Use a Script? 25
What is the Scripting Language....26
Understanding Texture Mapping 26
  Sections....26
  Sidedef Textures - Main 27
  Sidedef Textures - Below And Above....28
  Sidedef Textures - Manual Adjustments 29
  Sidedef Textures - Common Applications....29
  Sector Textures 30
Why Does the Hall of Mirrors Happen? 30

Why Can’t I Just Draw Lines on the Map?..31
Why is There No Sound When Running a Map?..31
Reference..31
- Commands...31
  - About WadAuthor..32
  - Align Linedef Textures Command 32
  - Apply Sector Motif Command 32
  - Arrange Icons Command 32
  - Best Fit Command..32
  - Cascade Command...32
  - Check Map Command..32
  - Clear Object Zoom Command...33
  - Close Command 33
  - Context Help Command...33
  - Convert Sector To Door Command 33
  - Convert Sector To Stairs Command 33
  - Copy Command...33
  - Create Motif From Sector Command 34
  - Customize Command...34
  - Cut Command 34
  - Decrease Grid Command 34
  - Delete Command...34
  - Delete Command...34
  - Edit Raw Data Command 35
  - Exit Command..35
  - Flip Linedef(s) Command 35
  - Force Node Build Command 35
  - Full Screen Command..35
  - Go To Bookmark Command 35
  - Help Topics Command...36
  - How To Use Help Command...36
  - Increase Grid Command 36
  - Insert Object Command...36
  - Join Linedefs Command 36
  - Join Vertexes Command 37
  - Make Linedefs Parallel Command 37
  - Map Name Command 37
  - Most Recently Used File Command 37
  - Motifs Command 37
  - New Command...37
  - New Polygonal Sector Command 38
  - New Rectangular Sector Command...38
  - New Thing Command 38
  - New Window Command..38

Object Filter Command 38
Object Filter Commands 38
Object Zoom Command 39
Open Command 39
Options Command 39
Paste Command 39
Print Command 39
Print Preview Command 40
Print Setup Command 40
Properties Command 40
Refresh Command 40
Register Command 40
Rotate Sector Command 40
Run Map Command 41
Save As Command 41
Save Command 41
Scale Sector Command 41
Scripts Command 41
Scroll Commands 41
Search For Help On Command 42
Select All Command 42
Select Configuration File Command 42
Select Map Command 42
Set Bookmark Command 43
Set Linedef Length(s) Command 43
Show Floor Grid Command 43
Show Grid Command 43
Show Thing Bitmaps Command 43
Snap To Grid Command 43
Split Linedef(s) Command 44
Statistics Command 44
Tags Command 44
Thing Display Command 44
Tile Horizontally Command 44
Tile Vertically Command 44
Tip Of The Day Command 44
Undo Command 45
User Defined Tool Command 45
Window Number Command 45
Zoom In Command 45
Zoom In Max Command 45
Zoom Out Command 45
Zoom Out Max Command 46
Configuration Files 46

  - Wadgame Section....46
  - Default Sections....47
  - Other Sections....48
- Context Menus....48
  - Map Editing View....48
  - Map Object....49
- Initialization File....51
- Keyboard Interface....52
  - Global Commands....52
  - Main Frame Window....52
  - Map Editing View....52
    - Map Editing View - File Commands....52
    - Map Editing View - Tool Commands....53
    - Map Editing View - View Commands....53
    - Map Editing View - Selection Commands....54
    - Map Editing View - Edit Commands....56
- Menu Commands....57
  - File Menu....57
  - Edit Menu....57
  - View Menu....58
  - Tools Menu....58
  - Window Menu....59
  - Help Menu....59
- Motifs....59
- Mouse Interface....60
  - Map Editing View....60
    - Primary Mouse Button (Usually Left)....60
    - Secondary Mouse Button (Usually Right)....61
- Properties....61
- Using Custom Images....62
  - How To Use Custom Images....62
- Registration....62
  - Snail Mail....63
  - WadAuthor Order Form....64
  - Order Checklist....65
- Troubleshooting....66
  - The Graphics Are Wrong....66
  - The Run Map Feature Fails....66
  - Why Is There No Sound When Running A Map?....67
  - Things Are Missing....67
- Getting Technical Support....67

About WadAuthor

WadAuthor came about during the process of crafting my first wadfile for DOOM. I truly prefer the Windows environment to DOS, and it frustrated me greatly that the only editor I could really count on was a DOS application! As I learned more about the DOOM engine during the construction process, I starting thinking fondly about creating my own editor. Because several of the technical issues involved were largely unfamiliar ground to me, I approached the entire thing as one big research project. WadAuthor is the result of that process.

7/26/96 Release 1.30. If an external wadfile is used for additional images, it will now be included when running the map. Fixed a focus problem with the image browser dialog incremental searching. Improved zoom under WindowsNT to a maximum of 400%. Improved zoom under Windows95 to a maximum of 250% by limiting map size to 16000 units square. Changed the tag dialog to center the map editing view on the selected object. Added the make column option to the new rectangular sector dialog and the new polygonal sector dialog. Added the current zoom setting to the status bar. The image browser dialog now appends an asterisk to the name of an image if it is not supplied by the main wadfile. When running a map, WadAuthor will now create a response file for passing arguments if the command line length is greater than the operating system allows. Added panning on Ctrl+Shift primary click. This feature allows the user to scroll the map view by grabbing a location and dragging rather than by using the scroll bars. Added the run map dialog for customizing options immediately prior to launching the game. Added the tip of the day dialog to provide on-going help and amusement.

6/21/96 Release v1.22. Fixed duplicate thing description problem with Hexen. Changed shift-primary-click to add an object it to the selection if the user does not drag a rubberband selection. Changed primary-click on a selected object to select exclusively if a move operation is not performed. Changed the automatic tag assignment feature to select first available tag rather than use the hightest numbered tag plus one. Changed the rubberband selection to add to the existing selection rather than clearing it first. Changed the current object handling to clear automatically when the cursor is not over an object. Added code for handling long filenames correctly when running a map or executing user-defined tools. Made waiting for user-defined tool execution an option. User-defined tool replaceable parameters now include long filename macros for passing long filenames to 32-bit Windows applications. Added the How To Make A Lift help topic. Added the ability to make one linedef parallel to another, optionally aligning it at a fixed distance.

5/31/96 Release v1.21. Fixed a GPF when setting external image handling to the current document with an unsaved document. Fixed a bug that caused thing types to appear multiple times within the thing property dialog. Fixed a bug preventing the save of a read-only file under a new filename. Fixed a selection list bug causing a GPF during map reloading. Fixed image acceleration code causing a GFP during map reload. Fixed GPF during image browsing for flat images of sizes other than 64 x 64 pixels. Fixed stair motif floor and ceiling runner image list controls to browse textures rather than flats. Fixed invalid Heretic defaults for stair motifs. Fixed detection of flats within a graphics PWAD using FF_START/FF_END. Added 'X' keystroke shortcut for splitting selected linedefs. Added 'J' keystroke shortcut for joining selected objects. Changed the tag handling in property dialogs to allow selection from a list of available tags. Added the ability to set linedef lengths.

2/19/96 Release v1.20. Flipping linedefs now forces a node rebuild during the next save operation. Script import and export dialog file types now work under Windows95. The missing Floor_ForceLightning special has been added to the Hexen configuration file. The Hexen Arc of Death thing has been re-classified as a weapon. The object insertion dialog has been fixed to request a location from the user. The map checking routine has been fixed to prevent interpreting a local door as having an invalid tag number. Fixed a problem with clipboard pasting that prevented the front sides of linedefs from getting properly fixed up with the correct surrounding sector number. The center on map object dialog no longer allows an object number one larger than the maximum. When deleting a sector, WadAuthor no longer deletes two-sided linedefs still in use by another sector. A bug that caused texture alignment to fail has been fixed. Added the ability to abort running the map if the WARUN.EXE utility is not present. Improved performance when browsing thing images by the addition of a multi-threaded preload routine at startup. Under Win32s (which does not support threads), the data will still be preloaded, it will simply delay application startup. When deleting a sector, WadAuthor now goes to much greater lengths to make sure all remaining one-sided linedefs have a main texture. Changed thing drawing to display facing angle for the appropriate classes. Added incremental searching to all image name edit controls. Clicking on an unused portion of the map with the primary mouse button begins a rubberband selection. Changed insertion of rectangular sectors to allow separate specification of horizontal and vertical widths. Changed the center on map object dialog to select the number control when any of the type controls are selected to simply keyboard use. Added the ability to disable snap to grid on the fly during a drag operation by holding down the shift key. Improved drawing to make a distinction between one-sided and two-sided linedefs. By default, one-sided linedefs are drawn in a lighter color. Removed almost all restrictions on door conversion; any sector connected to at least one other sector may now be converted to a door. Changed insertion of rectangular sectors to snap upper left vertex to grid only if the snap to grid feature is enabled. Changed insertion of polygonal sectors and things to snap center to grid only if the snap to grid feature is enabled. Changed Hexen sector to door conversion to stop using up a tag number. Changed check map dialog image problem fixing to use images from the linedef to be fixed, if available, before relying upon the current sector motif. Made image caching defaults sensitive to operating environment: 75 under Windows/NT, 50 under Windows95, and 25 for all other environments. Image acceleration code has been extended to support all known modes (256 colors, 64k colors, 16M colors, and true color). Added Windows95 spin controls where appropriate to improve mouse interface. Changed polygonal sector calculations to produce normalized geometric figures. Canceling a drag and drop operation now restores the previous view settings. Optimized screen redraw performance. Added a floor and ceiling image alignment grid. To enable or disable this feature, select the Show Floor Grid option from the context menu. To change the default setting, choose Options from the Tools menu; the default setting may be changed from the Views page. Added a bookmark feature. WadAuthor now supports additional images within the current document or within an external wadfile. Added a document properties dialog. Added a feature to force a node rebuild during the next save operation. Added complete color customization. Added a full-screen view feature. Added a tag editing and assignment option to the context menu. The helpfile has been converted to a more Windows95 friendly format. WadAuthor now “remembers” the last open map for each wadfile.

12/18/95 Release v1.11.1. Fixed several bugs that went undetected during testing. Released without updating documentation as no feature changes occurred.

12/11/95 Release v1.11. Fixed the check map dialog problem list to use the correct system-wide highlighted text color setting. Fixed a GPF when trying to add a SCRIPTS resource to maps that did not previously have one. Fixed a bug that incorrectly identified maps as not being ready to run. Fixed a bug that prevented scripts from being compiled when using the 32-bit version under Win32s. Fixed a bug that prevented the script dialog buttons from appearing correctly when using the 32-bit version under Win32s. Fixed a bug that caused the grid setting to be 2 units when using the 32-bit version under Win32s. Fixed a spelling error in the helpfile about topic. Fixed a bug in the thing properties dialog that caused the thing type to be set incorrectly. Fixed a bug that prevented the available script numbers from being updated after a successful script editing session. Fixed a bug with running Hexen maps. Added manual entry of thing angle to support polyobjects. Added a helpfile topic on understanding polyobjects and expanded the understanding scripts topic. Added import and export buttons to the scripts dialog. Improved sector determination. Added full source-level script decompilation. Added an object zoom feature. Added some example Hexen wads to the product.

11/17/95 Release v1.10. Fixed several bugs in vertex and linedef joining. All known restrictions have been removed. Fixed a bug that occurred only when saving a new map failed. If the initial save failed for any reason, future attempts to save would fail. Fixed a bug when pasting a copied sub-sector into an unused portion of the map. Previously, the sub-sector's linedefs would not be correctly updated as one-sided. Fixed a bug in the 16-bit version that would cause all linedefs within a wadfile to be incorrectly described (usually as local doors or something similar). Pasting and undo operations in a wad image name edit control now properly update the associated viewer (if any). An object property editing bug has been fixed to prevent generation of an empty undo operation. Reduced map load times by block reading wadmap objects. Reduced internal node-building times slightly by internally caching more data. Added a drag cursor setting to the view options. Uncheck it if the normal drag cursor obscures the drag destination to an unacceptable degree. Added a default grid size setting to the view options. Added new keyboard accelerators '[' and ']' for increasing and decreasing the grid size. Some systems, particularly when running Windows95, would not recognize the Shift+Plus and Shift+Minus default accelerators. Added new undo and check map buttons to the standard toolbar. Decreased the minimum grid setting to two units. Improved sector determination to ignore linedefs whose front and back face the same sector. This solves identification problems with sectors that fold back upon themselves or contain “tripwire” linedefs. Changed secondary mouse button usage to be more consistent with Windows95 interface guidelines. Joining objects now re-checks the current map if the check map dialog is active. Added a fix all button to the check map dialog. When pressed, it will fix all errors capable of being fixed without input from the user. Replaced the new thing dialog with the standard thing properties dialog. Improved the thing properties dialog to allow graphical selection of the thing type. Clicking the thing preview will invoke the image browser dialog, providing an alphabetically sorted list of the available things. The tags dialog now displays the tags in ascending sorted numerical order. Added a user-defined tools section to the tools menu. Added new view and user toolbars. Added a new toolbars page to the options dialog. It allows the user to specify which toolbars should be visible. Optionally, the user may right click on any toolbar to change these settings without opening the options dialog. Changed all toolbars to support docking to the frame window and each other. Also added code to remember the toolbar layout between WadAuthor sessions. Added a dialog box for inserting a rectangular sector to allow the user to specify the side length, optionally using the current grid setting. Added raw data editing for single objects. This allows power users to directly specify the raw data that will be saved to disk. This feature should be used with care. Added limited support for Hexen, the newest wadgame from Raven/Id Software. Added support for editing and creating maps for the fourth episode of Ultimate DOOM. Clipboard support has been enhanced to support exchange of data across all game types. When transferring data from Hexen to other wadgames, some conversion may be necessary, but existing architecture will be preserved as much as possible.

9/29/95 Maintenance release v1.03. Rewrote document saving code to avoid several GPF's for maps with lots of sidedefs. Fixed bug that prevented the selected door motif from being applied during conversion. Fixed GPF when trying to join two-sided lines with an invalid back sector mapping. Fixed bug preventing drag and drop joining of two-sided linedefs. Added support for using an external node building utility. Added specific error messages for failed node building, failed application of wadfile changes, and new wadfile creation. Added the ability to set the sector mapping for linedef sidedefs. Enhanced the map statistics dialog. Added a progress dialog for save operations.

8/25/95 Maintenance release v1.02. Joining linedefs has been enhanced to allow joining of any two linedefs -- as long as the sectors involved are valid. Speeded up linedef drawing when zoomed in. Fixed duplicate vertex identification bug. Added views page to options dialog and renamed maps page to files page. When snap to grid is off, keyboard-based object insertion no longer snaps to grid. Using keyboard to set current object now updates status bar. Added multi-sector rotation and scaling with optional inclusion of things therein. Sectors will now be joined when joining vertexes. Several floating point errors in the 16-bit version have been fixed. Fixed bug with clipboard when copying a sub-sector into another sector. Fixed bug preventing maps with > 8192 objects from loading in 16-bit windows version. Add ability to rename the current map within the current wadfile. Added the Select All option to the Edit menu. Added the Tutorial, Troubleshooting, and What’s New? items to the Help menu. Improved the map checking routine to report linedefs with an invalid tag number. Using the check map dialog’s fix button to repair problems with the wadfile was not properly allowing the action to be undone, nor marking the document as changed; this bug has been fixed. Fixed a bug in the check map dialog that prevented objects numbered zero from being viewed. A double-click in the type listbox of the new thing dialog exits the dialog, placing the selected thing.

7/17/95 Maintenance release v1.01. Added a snap to grid option, added color thing printing for color printers, changed polygonal sector limits, fixed a bug that prevented the File/Save As... option from working, fixed a palette handling bug that allowed WadAuthor’s display to be corrupted when switching back from another program, fixed the VERSIONINFO resource, expanded the text for some linedef floor codes, added a check for the WARUN.EXE utility when running a map, added options dialog, and fixed a bug in the 16-bit version of the REJECT resource creation that prevented maps with more than 217 sectors from working.

6/30/95 Initial public release. Completed the keyboard interface, added thing display dialog, added texture alignment, added tags dialog, enhanced check map dialog, added printing and print preview, expanded the help file, improved current object determination for sectors, and breathed a big sigh of relief!

5/18/95 Released the second beta to some friends. Added sector scaling and rotation, added stair conversion, added door and stair motifs, added undo functionality, greatly expanded the help file, tied context-sensitive help into the app, added configuration files for DOOM][and HERETIC, added support for color depths above 256, added object filtering, and fixed quite a few bugs.

4/12/95 Released the first beta to some friends. Major functionality seems intact; lacking good documentation and some polish.

Technical Information

WadAuthor was born as a collection of humble 16-bit DOS utilities, migrated into a couple of Windows 3.x utilities, and finally became an MFC application using Microsoft Visual C++ v1.52. With the availability of Microsoft Visual C++ v2.0, WadAuthor started straddling the 32/16 bit fence as a Windows/NT native application. It has been tested in the Windows 3.1, Windows for Workgroups v3.11, Windows/NT v3.0-v4.0 (beta), Windows95 Preview, and Windows95 final release environments.

The main WadAuthor executable is compiled from 72,000+ lines of C++ code, roughly one-third of which is specific to wadfile editing. It relies on a couple hundred lines of AWK code used for generation of the wadgame configuration files and other miscellany. The help file, developed with the assistance of RoboHelp for Windows95, is compiled from around seventy separate document files and over one-hundred images.

What’s New

This topic highlights the changes in each new release of WadAuthor. The complete history can always be found in the About WadAuthor topic, but simply knowing what changed doesn’t often help one use those new features. You may click on any of the sections listed below to quickly jump to that section.

Overview

The latest release is a minor release. As such, it fixes existing bugs and adds minor features without major alterations from the user's perspective. The list of bug fixes, enhancements and new features follows.

Bug Fixes

- If an external wadfile is used for additional images, it will now be included when running the map.
- Fixed a focus problem with the image browser dialog incremental searching.

Improved Features

- Improved zoom under WindowsNT to a maximum of 400%.
- Improved zoom under Windows95 to a maximum of 250% by limiting map size to 16000 units square.
- Changed the tag dialog to center the map editing view on the selected object.
- Added the make column option to the new rectangular sector dialog and the new polygonal sector dialog.
- Added the current zoom setting to the status bar.
- The image browser dialog now appends an asterisk to the name of an image if it is not supplied by the main wadfile.
- When running a map, WadAuthor will now create a response file for passing arguments if the command line length is greater than the operating system allows.

New Features

- Added panning on Ctrl+Shift primary click. This feature allows the user to scroll the map view by grabbing a location and dragging rather than by using the scroll bars.

- Added the run map dialog for customizing options immediately prior to launching the game.

- Added the tip of the day dialog to provide on-going help and amusement.

- A special version of the WTF Productions DOOM and DOOM][Editing Guide is now included with WadAuthor.

Overview

The latest release is a minor release. As such, it fixes existing bugs and adds minor features without major alterations from the user's perspective. The list of bug fixes, enhancements and new features follows.

Bug Fixes

- If an external wadfile is used for additional images, it will now be included when running the map.

- Fixed a focus problem with the image browser dialog incremental searching.

- Improved Features

- Improved zoom under WindowsNT to a maximum of 400%.

- Improved zoom under Windows95 to a maximum of 250% by limiting map size to 16000 units square.

- Changed the tag dialog to center the map editing view on the selected object.

- Added the make column option to the new rectangular sector dialog and the new polygonal sector dialog.

- Added the current zoom setting to the status bar.

- The image browser dialog now appends an asterisk to the name of an image if it is not supplied by the main wadfile.

- When running a map, WadAuthor will now create a response file for passing arguments if the command line length is greater than the operating system allows.
- New Features
- Added panning on Ctrl+Shift primary click. This feature allows the user to scroll the map view by grabbing a location and dragging rather than by using the scroll bars.
- Added the run map dialog for customizing options immediately prior to launching the game.
- Added the tip of the day dialog to provide on-going help and amusement.
- A special version of the WTF Productions DOOM and DOOM][Editing Guide is now included with WadAuthor.

Basic Vocabulary

It is important to establish a minimum working vocabulary to use WadAuthor effectively. The terms you need to understand before proceeding are summarized in the following table. Consult the glossary for a more comprehensive list.

Term Definition

Linedef: A linedef defines a wall within a map. A linedef has a starting vertex, an ending vertex, and various other information.

Map: A map is stored within a wadfile as a well-defined list of resources. Each map corresponds to a single “level” or “mission” when played.

Node building When a map is saved, quite a bit of information must be generated before the map can be used in a game. This information is allows the game engine to quickly choose which walls need to be drawn for the player, avoiding time intensive computations at run-time. This process is referred to as node building.

Sector: A sector defines a room within a map. It contains lighting conditions, floor and ceiling heights, and other information.

Sidedef: A sidedef defines how a wall appears from a given side. Each linedef may have one or two sidedefs associated with it based on whether the player can ever see the back side.

Thing: A thing defines a single object within a map. Most things interact with the player in some way, although some are strictly scenery. A few examples are weapons, ammunition, and enemies.

Vertex: A vertex defines a single point within a map via Cartesian coordinates. For those who don't remember geometry, a Cartesian coordinate is a simple pair of orthogonal values usually denoted by (x,y).

Wadfile: A data file intended for use with several games from id software. A wadfile consists of a header, resources, and a resource directory.

Wadgame: Wadgame is a term I have coined to refer to any game utilizing a wadfile. A few examples are DOOM, DOOM][, and HERETIC.

Understanding Wad Files

A wadfile is simply a data file that adheres to the same format as the original DOOM.WAD file released with DOOM. A wadfile must have a header, a number of resources, and a directory. Presumably, wadfiles were given their name because they provide a simple way to store "wads of stuff." The header format, resource format, and directory format are described in the sections that follow. Non-technical types may wish to skip to the next topic; all you really need to know to use WadAuthor effectively is that maps are stored within wadfiles.

Header

At the beginning of every wadfile, a twelve-byte header structure contains a four-byte signature. The signatures of which I am aware are listed in the following table.

Signature	**Description**
IWAD	Possibly short for "internal wadfile", this denotes a main wadfile designed for use with a wadgame. The aforementioned DOOM.WAD is an example of an IWADwadfile.
PWAD	Possibly short for "patch wadfile", this denotes an add-on wadfile. All new wadfiles created by WadAuthor are initialized with this

	signature
TWAD	Short for template wadfile, this signature was introduced by DoomCAD, another wad editing application for Windows. Following the four-byte signature is a four-byte number which provides the number of directory entries in the wadfile directory. The remaining four bytes is a number which provides the offset, from the beginning of the wadfile, at which the directory exists.

Resources

The resources within a wadfile exist in many formats, depending upon their purpose. For example, map data, image data, sound data, and textual data are all stored differently. Because of this, I will offer no further explanation here. For those interested in learning more about the various resources, I would suggest that you visit various internet sites or other on-line services for more detail.

Directory

A wadfile directory consists of a number of directory entries; the exact number, as explained above, is given by the wadfile header. Each directory entry is a sixteen-byte structure. The first four bytes is a number providing the offset, from the beginning of the wadfile, at which the resource exists. The second four bytes is a number providing the size, in bytes, of the resource. The remaining eight bytes provides a textual name for the resource.

Understanding Wad Maps

A single wadmap corresponds to a single mission or level within a given wadgame. Wadmaps are stored within a given wadgame's main wadfile until needed at runtime. The required wadmap resources and geometry are explained in the sections that follow.

Resources

A wadmap is defined by a fixed series of wadfile resources. Their names, listed in the order in which they appear in the wadfile directory, and a brief description of their purpose are given in the following table.

Name	**Description**
Mapname	The map naming conventions vary between different wadgames. For example, DOOM uses the EnMn convention replacing the n’s with the episode and mission numbers, while DOOM][uses the MAPnn convention, replacing the n’s with the two-digit mission number.
THINGS	The things resource contains the thing data for the map.
LINEDEFS	The linedefs resource contains the linedef data for the map.
SIDEDEFS	The sidedefs resource contains the sidedef data for the map.
VERTEXES	The vertexes resource contains the vertex data for the map.
SEGS	The segs resource is used to define ssectors.
SSECTORS	The ssectors resource divides all of the sectors into convex polygons.
NODES	The nodes resource contains branches in a binary space partition (BSP) that partition the map. It is used to determine which walls are in front of others.
SECTORS	The sectors resource contains the sector data for the map.
REJECT	The reject resource contains a two-dimensional matrix that determines whether enemies in a given sector can detect or attack players in another sector.
BLOCKMAP	The blockmap resource contains a series of tables that simplify collision detection between walls and moving objects.
BEHAVIOR	The behavior resource is only present in maps created for Hexen. It contains compiled script data for the map.
SCRIPTS	The scripts resource is only present in maps created for Hexen and is optional. WadAuthor uses this

	resource to maintain the script code for a given map. This resource is a proposed standard accepted by some wad editors.

WadAuthor maintains the things, linedefs, sidedefs, vertexes, and sectors resources for a given map in memory. It is only when the map is saved that the segs, ssectors, nodes, reject, and blockmap resources are generated on disk. Because the node building process can be quite lengthy for large maps, WadAuthor only does it when necessary.

You can use this information to your advantage by approaching map construction in two separate phases: creating and modifying. If you complete your vertex-related editing chores during the creating phase, saving your map during the modifying phase will be much quicker. In general, any time you add a new vertex or delete or move an existing vertex, your next save will incur the time hit for a complete node rebuild.

Geometry

You remember geometry, right? Well, don’t worry, WadAuthor takes care of virtually all of the geometry involved in creating new maps. It is important, however, that you understand how the wadmap resources relate to each other; otherwise, you risk creating a map that cannot be played due to one or more architectural errors. The following list provides a geometric overview of the relevant map object types.

- Vertexes define locations and are essentially points in the Cartesian plane.

- Linedefs are strung between vertexes and are essentially line segments.

- Sidedefs define how the front and back of a linedef appears from the player’s perspective. They also determine which sectors the front and back of a linedef face. They have no geometric equivalent, true, but they do provide the crucial link between linedefs and sectors.

- Sectors define some of the attributes of a “room” and are essentially polygons.

These simple relationships have some very important implications for budding map designers. Some of the most important points to grasp are listed below.

- Deleting a vertex will delete any linedefs that reference it.
- A linedef can be one-sided or two-sided.
- A one-sided linedef is essentially a brick wall; it cannot normally be crossed and only has a front side. A one-sided linedef that has its back side facing the sector with which it is associated is going to produce the hall of mirrors effect at runtime.
- A two-sided linedef can normally be crossed. Because the sectors faced by its front and back sides can have different floor and ceiling heights, great care must be taken in specifying the below, main, and above images to avoid the hall of mirrors effect at runtime.
- Deleting a linedef can leave a sector open. All sectors must be closed polygons for the node building process to work.
- Deleting a sector affects its associated linedefs and vertexes. One-sided linedefs are deleted, two-sided linedefs become one-sided, and any vertexes no longer in use are deleted.
- WadAuthor is doing an awful lot of work for you behind the scenes!

Creating New Objects

WadAuthor allows you to create two basic types of objects: sectors and things. Sectors may be created anywhere within a map, while things must be created within an existing sector. Sectors created in an unused portion of the map have one-sided linedefs, separating them from the rest of the map. Sectors created within other sectors have two-sided linedefs, acting as sub-sectors through which the player can freely travel. Step by step instructions follow.

Creating a New Thing

- Right-click within an existing sector where the thing should be placed.
- Select the New Thing.. option from the context menu.
- Enter the desired parameters in the resulting dialog box and press the OK button to complete the new thing, or press the Cancel button to abort.

- Creating A New Rectangular Sector
- Right-click anywhere on the map roughly where the sector should be centered.
- Select the New Rectangular Sector option from the context menu.
- Enter the desired parameters in the resulting dialog box and press the OK button to complete the new sector, or press the Cancel button to abort.
- The vertexes of the new sector will be snapped to grid.
- Creating A New Polygonal Sector
- Right-click anywhere on the map roughly where the sector should be centered.
- Select the New Polygonal Sector... option from the context menu.
- Enter the desired parameters in the resulting dialog box and press the OK button to complete the new sector, or press the Cancel button to abort.
- The vertexes of the new sector will not be snapped to grid. This step is omitted to preserve the most accurate shape for the given number of sides and radius.

Selecting Objects

WadAuthor provides several ways to select objects. Most of them require setting the current object and pressing a key or clicking with the mouse. For more information, consult the related topics.

When a sector is selected, all of the linedefs associated with it are also selected. This is mainly to provide some kind of visual feedback to the user, although it is also useful for various editing operations.

Moving Objects

WadAuthor supports the drag and drop interface found in many Windows programs. To move an object or group of objects, select them, then click and hold the primary mouse button on one of the selected objects. Move the mouse and use the helpful wire-frame display WadAuthor supplies to position the objects at the correct final location. Release the mouse button to complete the move operation, or press the escape button at any time to abort the procedure.

When the snap to grid feature is enabled, WadAuthor allows the user to temporarily disable it during a drag operation by holding down the shift key.

Editing Existing Objects

Editing existing objects is a two step process of selection and manipulation. Whenever possible, WadAuthor supports direct manipulation of an unlimited number of objects of varying types. The most frequently useful editing operations are available through the context menu. Once the desired objects are selected, you may manipulate them. The following list describes some of the most useful operations made available by the context menu. For more information, consult the related topics.

- Flipping a linedef can reverse only the starting and ending vertexes or reverse the sidedef data as well. Avoid this feature unless you know what you're doing!

- Splitting a linedef (or linedefs) inserts a new vertex at the center of the line. This allows the user to quickly reshape a given sector as desired.

- Joining two linedefs is a fast means for connecting two sectors; select the first linedef, select the linedef to which it should be joined, and select the join option from the context menu. The first line will "move" to join the second.

- Joining two vertexes is a fast means for getting rid of a linedef or fixing a problem with an un-closed sector. Select the first vertex, select the second vertex to which it should be joined, and select the join option from the context menu. The first vertex will "move" to join the second.

- Creating a door manually is difficult at best; using WadAuthor is much simpler. Select any single four-sided sector that connects two other sectors and select the door conversion option from the context menu.

- Scaling or rotating a sector usually implies moving each associated vertex manually. Select any single sector, and select the desired option from the context menu.

- Creating a staircase is a lot of work; using WadAuthor is much simpler. Select any single four-sided sector and select the stair conversion option from the context menu.

- Editing object properties is probably the most common operation. Select any number of objects and select the Properties... option from the context menu.

Deleting Objects

Deleting objects deserves its own help topic because of the relationships between the different object types. Deleting the wrong objects can render a map unplayable, and require you to delete entire regions of the map to fix the problem. So, it is important that you understand the ramifications of deleting something. The following sections provide more information.

Cascade Effect

Deleting a single object can cause many other objects to be deleted, and I call this the cascade effect. For example, a linedef requires two vertexes, so deleting a vertex will necessitate deleting any linedefs that are attached to it. Sectors are made up of linedefs (actually they're referenced by sidedefs which are referenced by linedefs), so if deleting a vertex results in the deletion of any remaining linedefs in a given sector, the sector will also be deleted.

WadAuthor properly cascades a delete operation for you. For any objects that you delete, WadAuthor will make sure that any orphaned objects are similarly removed.

Corruption

It is all too easy a thing to corrupt a map. The most common source of problems is from deleting something that needs to be there. For example, a novice might see the linedef between two connecting sectors and figure it shouldn't really be there — after all, the player can't walk through walls, right? If you suspect that a delete operation may have corrupted your map, use the map checking tool before saving the map to find out for sure. If you have open sectors or other errors of that nature, you should probably undo the delete operation.

Making Things Happen

The wadgame environment is a dynamic one; that is, architectural features can change over time independent of or in response to a variety of conditions. In every case, however, achieving a desired effect is a matter of configuring linedefs and sectors. The following sections group these dynamic features into three categories: sector effects, simple linedef effects, and complex linedef effects.

Sector Effects

Depending on the wadgame, different sector types can be used to achieve a several effects. Flickering lights, floors that damage the player, and ceilings that drop or rise are common across all wadgames, while only the most recent allow the map designer to express ideas like moving water or wind. Sector-based effects are achieved by selecting one of the non-normal sector types from the sector properties dialog.

Simple Linedef Effects

Simple linedef effects are those which do not require the use of tag numbers. Examples of these include horizontally scrolling wall images and local doors. To achieve a simple linedef effect, select the desired linedef type from the linedef properties dialog — that's it. Granted, this isn't much of an explanation, but then these are supposed to be simple...

Complex Linedef Effects

Complex linedef effects are those which require the use of tag numbers. Examples of these include remote doors, lifts, et. al. Achieving a complex linedef effect is a matter of choosing the linedefs that trigger it, choosing the sectors it should affect, setting the tag numbers to match, and enjoying the results. For a step by step example on creating a remote door, check the related topics.

The Hexen wadgame adds a new realm of possibilities for wadfile design. By avoiding the use of a fixed set of codes for linedef actions in favor of a system of specials and scripts, there is little the wadfile designer cannot achieve. This comes, of course, at the expense of simplicity.

Understanding Polyobjects

Polyobjects were introduced with the release of Hexen. They allow the wad designer to create effects that are simply not possible with other wadgames due to the self-imposed limitations of the game engines. A polyobject is a complex architectural feature whose location is resolved at map load time. In English this means that a polyobject is a polygonal object defined at one location on the map that appears in a different location at run time.

There are several important points to making a polyobject work. Without following them the map may refuse to load, or the game may crash when the polyobject is triggered. The important points are listed below.

- You must select a number between 1 and 255 to uniquely identify the polyobject to be created. This number is used during the construction.
- You must define the polyobject. This is done so that the wadgame knows which linedefs to manipulate as part of the object. The simplest method to accomplish this is to let WadAuthor create the polyobject as a polygonal sector of the desired number of sides, then apply the Polyobj_StartLine special to one of the linedefs — supplying the polyobject number selected in the first step. It is possible to manually specify the linedefs, but it’s much easier this way.
- You must provide an anchor for the polyobject. This supplies the wadgame with the point around which the polyobject will rotate, slide, or whatever. This is done by placing a polyobject anchor thing at the desired position near the polyobject and setting the thing’s angle to the polyobject number selected earlier.
- You must provide a destination on the map to which the anchor will be translated at map load time. This supplies the wadgame with the final destination for the polyobject anchor. This is done by placing a start spot (crushing or non-crushing) at the location which the polyobject anchor should occupy at runtime and setting the thing’s angle to the polyobject number selected earlier.

What Is A Script

An Action Code Script, or ACS for short, is a series of instructions to be carried out. Originally the script is defined in terms of a C-like programming language. Prior to run time, however, the script code must be compiled into pcode form and stored with the map.

The term pcode is a kind of shorthand for pseudo-code. It refers to instructions that are executed via an interpreter. Pcode is often used where a fully compiled set of instructions is undesirable, but fully interpreted speed is unacceptable. Visual Basic programs, for example, are actually compiled as a bunch of pcode instructions of a different sort.

Why Are Scripts Compiled

he scripts are compiled because Id/Raven Software decided to do it! All kidding aside, I suspect the scripts must be compiled for sake of performance and simplicity in the runtime environment. As mentioned earlier, the script data must be compiled prior to running the map, or else the script will not execute correctly.

Where is the Data Stored

The compiled pcode is stored in a new BEHAVIOR resource for each map. WadAuthor stores the script code in another resource after the BEHAVIOR resource named SCRIPTS. This is a convention proposed by editor authors to allow wadfile designers to share their code and ideas more easily.

When WadAuthor opens a map for which no SCRIPTS resource exists, it tries to decompile the existing BEHAVIOR resource. The decompilation, if successful, will provide source code similar to the code the wadfile designer wrote. WadAuthor cannot determine original variable names or provide the comments, but the essential code will be preserved.

If the **BEHAVIOR** resource contains no scripts, a default file is read when script editing is desired. Compiling and saving edited script data will overwrite the original scripts for such a map, so caution should be exercised. A map can quickly be rendered unplayable by destroying some of the scripts required at runtime.

How Do I Use a Script?

Each script has a unique identification number that may be used in conjunction with the ACS_ special codes. For example, to trigger script number one when the player crosses a given linedef, the linedef type should be set to ACS_Execute, the arguments should be set to the script number, map number, and script arguments (if any), and the linedef activation should be set to Player crosses.

What is the Scripting Language

Programmers familiar with the C programming language will immediately recognize the scripting language as a variant of C. It supports many of the standard C language constructs including pre-processing, looping, variables, etc. For more information on the scripting language syntax, consult the official Hexen specifications included with WadAuthor.

Understanding Texture Mapping

Understanding wadgame texture mapping is, perhaps, the most difficult concept involved in creating good maps. More specifically, understanding how to achieve the desired texture alignment is somewhat complicated. The following sections explain wadgame texture mapping and the factors that affect it. For purposes of illustration, I have created a default texture, 128 units square, shown to the left. I will use this texture in all my examples. Because the following explanation is fairly lengthy, you may click on any of the sections listed below to quickly jump to that section.

Sections

Sidedef Textures - Basics
Sidedef Textures - Main

Sidedef Textures - Below And Above
Sidedef Textures - Manual Adjustments

Sidedef Textures - Common Applications
Sector Textures

Sidedef Textures - Basics

Every sidedef, for purposes of texture rendering, is broken into three distinct areas: the area below the sector’s floor, the area between floor and ceiling, and the area above the ceiling. Each area may have a different texture name assigned to it. If no texture is needed, a hyphen is used for the null texture. The texture names cannot be left blank — this will cause an error at runtime.

WadAuthor refers to the textures used for these areas as the Below, Main, and Above textures.

Wadgames render each area by tiling its texture horizontally and vertically until the entire area is covered. The texture will be truncated in either direction if the area dimensions are not an even multiple of the dimensions of the texture.

Textures whose vertical dimension is anything other than 64 or 128, stair runners for example, do not always tile well vertically. They sometimes result in small pink or multi-colored “cracks” appearing between the tiling. This is due to a bug in the rendering engine.

Sidedef Textures - Main

The vertical axis is handled differently for the three areas. Let’s begin with the main area, since it is the most commonly used. By default, the main texture is mapped from top to bottom. This can cause problems if the overall sector height, the difference between floor and ceiling, is not equivalent to the texture’s vertical size. The following illustrates two possible cases.

Here the sector is taller than the texture. Notice how the texture is tiled. If the texture had some distinctive feature at its bottom end that was designed to meet the floor, we’d have a real problem.

Here the sector is shorter than the texture Notice how the texture is truncated. In this case, a distinctive feature at the bottom end of the texture would be truncated.

As you can see, potentially undesirable results occur in both cases. Fortunately we have a way to modify the rendering behavior. Linedefs have two attributes that change the manner in which textures are vertically rendered. The lower unpegged attribute applies to the main and below textures, and the upper unpegged attribute applies to the above texture. The lower unpegged attribute will cause the main and below textures to be reverse rendered, from bottom to top. The following table illustrates the previous cases with the lower unpegged attribute set.

Notice how setting the lower unpegged attribute has made the bottom of the texture meet the floor. This can be particulary important when working with high ceilings.

In this case, the aforementioned distinctive feature would still appear connected to the floor. Our player may be crouching, but at least the walls look good...

As you can see, the texture is now being mapped from bottom to top. Setting the lower unpegged attribute can quickly solve most main texture alignment problems between sectors with the same floor height but different ceiling heights.

Setting the main texture to something other than the null texture can provide an interesting effect with a two-sided linedef. By setting one side of a linedef's main texture to null and the other side to something else, you effectively create a wall that isn't. It appears to be solid from one side, but is transparent from the other. Imagine the player's surprise when something big and nasty comes tromping through the wall after him!

Sidedef Textures - Below And Above

The issue is further complicated by the below and above textures. A linedef may be two-sided, having different sectors on either side. If the floor and ceiling heights of the first sector match that of the second, there's no problem. When they differ, the below and above textures become important. The below texture will be required when looking into a sector whose floor height is greater than that in which the player is standing. The above texture will be required when looking into a sector whose ceiling height is less than that in which the player is standing.

If the ceiling texture of both sectors is F_SKY1, the upper textures between them will always be invisible. This is, apparently, a special case to allow map designers to include outdoor scenes.

By default, the below texture is mapped from top to bottom, while the above texture is mapped from bottom to top. If this seems fairly innocent, consider the illustration to the left, a simple example of a window. A window is usually a small sector whose floor and ceiling are higher and lower than the surrounding sectors. In this event, the textures will be improperly aligned with the surrounding non-windowed walls.

As shown to the left, by setting the upper and lower unpegged attributes, we can fix this problem. The above texture will be rendered top to bottom, while the below texture will be rendered bottom to top.

Just when you thought the pain was over... I am sorry to say there is a single remaining complication with vertical texture alignment. In the above example, the sector in which the player is standing is the same height as the texture. If this is not the case, the textures will still be aligned, but perhaps not as one might suspect. If the sector in which the player is standing is shorter than the texture height, the bottom of the textures will be clipped as shown in the following illustration.

This probably seems confusing; after all, I just explained how lower unpegged renders from bottom to top, right? Here's the catch: rendering from bottom to top does not necessarily start at the bottom of the given texture. It selects its starting point within the texture based on the distance from floor to ceiling. It may seem confusing, but it's all done to allow nice alignment of sidedef textures.

Now for the easy part. In each area, the texture mapping proceeds along the horizontal axis from the left edge of the wall to the right, as seen from the player's perspective. This is not usually a problem except in a few cases. For example, a staircase usually consists of many small sectors with consistently varying heights connected together. If the sides of the staircase are visible, the horizontal textures will probably need to be aligned. The following illustration demonstrates a rising staircase using the example texture for the below texture. Notice the horizontal and vertical alignment problems.

The vertical alignment can be fixed by applying the appropriate unpegged attribute. The horizontal alignment problem provides a good lead in for the next section...

Sidedef Textures - Manual Adjustments

There are some situations in which it is necessary to make manual adjustments to the texture alignment. The previously mentioned horizontal alignment problem with a rising staircase, for example, requires an offset value for the textures to be tiled properly. The offsets determine how far into the texture rendering starts, and they may be positive or negative. Values larger than the size of the texture will wrap. WadAuthor provides editing of these offsets via the linedef properties dialog.

Calculating the horizontal offset is usually pretty simple. Calculating the vertical offset requires more work. The three factors to consider are the ceiling height of the current sector, the ceiling height of the adjacent sector, and the height of the texture. The desired offset will almost always be the difference between two of these values.

Sidedef Textures - Common Applications

Having provided such a wealth of information, it’s time to get to the good stuff: applying the knowledge. The following list details the most common situations in which texture alignment can be a problem.

Doors are usually implemented as a sector where the ceiling height is the same as the floor height. When the door is shut, the player sees the above texture. When the door opens, the ceiling of the door sector rises. None of the unpegged attributes are set; this lets the door image rise with the ceiling since the above texture is based at the bottom edge of the rising ceiling. The sidedefs making up the inside tracks of the door are usually one-sided and thus need only a main texture. Because main textures are rendered from the ceiling down, the inside tracks will rise as the door opens. This is usually not desirable. By setting the lower unpegged attribute for the inside tracks, they will be based on the non-changing floor and will remain stationary.

Secret doors add a little complexity if the door is to blend in with the surrounding wall. When the door is closed, the player normally sees the above texture, from floor to ceiling. Since the upper texture is rendered from the bottom up, the bottom of the door, where it meets the floor, will show the bottom of the texture. If the room has a non-standard ceiling height, the adjacent sidedef will not be vertically aligned with the closed door. The simplest solution is to set the lower unpegged attribute for all adjacent sidedefs, making them render bottom up as well.

Switches get shifted towards the floor if the sidedef to which they are applied is shorter than the switch texture. To place the switch at a better vertical position, set the lower unpegged attribute. The sidedef will render bottom up, and the switch will always be at a constant height.

Sector Textures

The floor and ceiling images are all 64 units square. Unlike sidedef textures, their alignment is very simple; they are aligned with an underlying 64 unit grid. Because of this, floor and ceiling textures will always be tiled seamlessly within any given sector. The only problem arises with images that contain special markings, teleporter pads for example. The sector in which these textures appear must be exactly aligned to a 64 unit grid to insure proper alignment.

Why Does the Hall of Mirrors Happen?

As briefly explained in the glossary, this refers to the effect caused by a sidedef without a texture when played within a wadgame. I do not know exactly why it happens (you'd probably have to ask the father of the gaming engine), but reproducing it is, unfortunately, rather easy.

Usually, when you see the hall of mirrors effect, you have a sidedef being viewed from above its ceiling or below its floor without a corresponding texture. The various wadgame engines seem to handle this by not handling it, resulting in the aptly-named hall of mirrors effect.

Why Can't I Just Draw Lines on the Map?

Truthfully, you could if I had taken a different direction in constructing WadAuthor. My favorite wad editor (prior to WadAuthor of course — shameless plug, eh?) allowed the user to create sectors by drawing the linedefs. I got tired of clicking all over the place just to draw a single sector, and I decided that my editor would work differently. I found myself thinking in terms of rooms and the connections between them most of the time; that dictated how WadAuthor works.

There is, actually, another good reason as well: sectors are referenced by sidedefs through linedefs. Maintaining the proper internal data structures in the program was greatly simplified by allowing WadAuthor to do the sector construction. I'm hoping that most people will prefer WadAuthor's methods just as I do.

Why is There No Sound When Running a Map?

The WadAuthor configuration files disable the sound by default to avoid problems. If you are running WadAuthor under Windows95, however, it is quite possible that your hardware is capable of providing sound. To test this, edit the wadgame configuration file with notepad, or some other text editor, and remove the -nosound option from the Run entry in the Wadgame section.

Be sure to make a backup of the original configuration file before editing it.

Reference

Commands

About WadAuthor

The about WadAuthor command invokes the about dialog.

Align Linedef Textures Command

The align linedef textures command adjusts the texture offsets to insure that the images will blend seamlessly from the player's perspective.

Apply Sector Motif Command

The apply sector motif command displays the apply motif dialog, allowing the user to select a motif to be applied to the selected sectors.

Arrange Icons Command

The arrange icons command re-arranges any minimized windows within the client area of the main window.

Best Fit Command

The best fit command centers the map within the map editing view and adjusts the scale so that the entire map can be viewed at once.

Keyboard: Home

Cascade Command

The cascade command re-arranges the open windows in a cascading fashion.

Check Map Command

The check map command examines the currently active map for errors and, if any are found, displays check map dialog to aid in fixing them.

Keyboard: Ctrl+K

Clear Object Zoom Command

The clear object zoom command restores the map editing view scale to its original settings prior to the first use of the object zoom command.

Keyboard: Shift+Z

Close Command

The close command closes the currently active map.

Keyboard: Ctrl+F4

Context Help Command

The context help command changes the cursor to the context help cursor. Help will be provided for the next menu item, button, dialog box, or other control upon which the user clicks.

Keyboard: Shift+F1

Convert Sector To Door Command

The convert sector to door command changes the selected sector into a door as specified by the current door motif.

Convert Sector To Stairs Command

The convert sector to stairs command displays the convert sector to stairs dialog, allowing the user to change the selected sector into a staircase as specified by the current stair motif.

Copy Command

The copy command places a copy of the current selection on the clipboard.

Keyboard: Ctrl+C

Create Motif From Sector Command

The create motif from sector command creates a new motif based on the selected sector. To invoke the command, select the sector, then click the secondary mouse button on it to invoke the context menu and choose the Create Motif From Sector option.

Customize Command

The customize command displays the user tools dialog to allow the user to customize the list of user tools.

Cut Command

The cut command removes the current selection and places it on the clipboard.

Keyboard: Ctrl+X

Decrease Grid Command

The increase grid command decreases the map editing view grid spacing.

Keyboard: Shift+Minus
[

Delete Command

The delete command removes the contents of the current selection.

Keyboard: Delete

Delete Command

The delete command removes the contents of the current selection.

Keyboard: Delete

Edit Raw Data Command

The edit raw data command invokes one of the raw editing dialogs to allow the user to access the actual data saved to disk for a given map object. This feature should be used with caution.

Exit Command

The exit command closes the WadAuthor application.

Keyboard: Alt+F4

Flip Linedef(s) Command

The flip linedefs command displays the flip linedefs dialog, allowing the user to specify how the front and back sides of the linedef should be exchanged.

Force Node Build Command

The force node rebuild command will cause a node building operation to be performed the next time the map is saved — whether it is actually needed or not.

Full Screen Command

The full screen command toggles the state of the full screen display mode. Full screen mode hides the main window caption, menu, toolbar docking frames, and status bar in order to allow open views to occupy the entire screen. When using full screen mode, the special full screen floating toolbar, shown in the illustration below , will be displayed. Clicking the button will toggle full screen mode.

Keyboard: Ctrl+U

Go To Bookmark Command

The go to bookmark commands restore the view position and zoom settings from the given bookmark number.

Keyboard: Ctrl+1 through Ctrl+9

Help Topics Command

The contents command displays the help topics for the WadAuthor help file.

Keyboard: F1

When invoked from the keyboard, this will display help for the current context, if any, before defaulting to the contents topic.

How To Use Help Command

The how to use help command displays the Windows help on using the help file viewer.

Increase Grid Command

The increase grid command increases the map editing view grid spacing.

Keyboard: Shift+Plus
]

Insert Object Command

The insert object command displays the insert object dialog allowing the user to select a type of object to be inserted into the map.

Keyboard: Insert

Join Linedefs Command

The join linedefs command joins the first selected linedef to the second. This command can be used to connect two sectors so that the player may pass between them.

Keyboard: J

Join Vertexes Command

The join vertexes command joins the first selected vertex to the second. This command can be used to reduce the number of sides in a sector or closing open sectors.

Keyboard: J

Make Linedefs Parallel Command

The make linedefs parallel command displays the make linedefs parallel dialog, allowing the user to specify options to be used in performing the command.

Map Name Command

The map name command invokes the map name dialog box to allow the user to change the name of the current map.

Most Recently Used File Command

The most recently used file commands open the wadfiles listed on the File menu.

Motifs Command

The motifs command displays the motifs dialog for the current wadgame configuration file.

Keyboard: Ctrl+M

New Command

The new command creates a new map and a new window in which to edit the map.

Keyboard: Ctrl+N

New Polygonal Sector Command

The new polygonal sector command displays the new polygonal sector dialog and inserts the specified sector into the map.

New Rectangular Sector Command

The new rectangular sector command displays the new rectangular sector dialog and inserts the specified sector into the map.

New Thing Command

The new thing command displays the thing properties dialog and inserts the specified thing into the map.

New Window Command

The new window command opens another window into the currently active map. All editing operations are synchronized across multiple windows into a single map.

Object Filter Command

The object filter command displays the map editing view to specify the type of objects to edit.

Keyboard: Ctrl+F

Object Filter Commands

An object filter command sets the object filter to a specified object type.

Keyboard

Key	Effect
L	Sets the object filter to linedefs.
N	Sets the object filter to none.
S	Sets the object filter to sectors.

T	Sets the object filter to things.
V	Sets the object filter to vertexes.

Object Zoom Command

The object zoom command configures the map editing view scale to best fit the current object.

Keyboard: Z

Open Command

The open command displays the file open common dialog to allow the user to open a map from an existing wadfile.

Keyboard: Ctrl+O

Options Command

The options command displays options dialog to allow the user to modify WadAuthor configuration settings.

Paste Command

The paste command inserts the contents of the clipboard into the current map at a user-specified position. WadAuthor will allow you to specify the position by using the arrow keys to move the insertion point, or by clicking the primary mouse button at the desired location. Pressing the escape key will abort the operation.

The paste location determines which sector will be used as the parent for any linedefs requiring fixup during the paste operation.

Keyboard: Ctrl+V

Print Command

The print command displays print dialog to print the currently active map.

Keyboard: Ctrl+P

Print Preview Command

The print preview command displays a preview of the currently active map as it would appear when output to the current printer.

Print Setup Command

The print setup command displays the print setup dialog to configure the current printer.

Properties Command

The properties command edits the properties of the current selection. If there is no current selection, it displays the document properties instead.

Keyboard: Alt+Enter

Refresh Command

The refresh command causes the currently active map to perform a complete redraw.

Keyboard: F5

Register Command

The register command displays the register dialog which allows the user to authorize the software license.

Rotate Sector Command

The rotate sector command displays the rotate sector dialog, allowing the user to specify the angle by which the selected sector should be rotated.

Run Map Command

The run map command runs the currently active map using the run command specified in the current wadgame configuration file.

Keyboard: Ctrl+R

Save As Command

The save as command displays file save common dialog and saves the currently active map under a new filename.

Save Command

The save command saves the currently active map. If the map has not previously been saved, file save common dialog will be displayed to allow the user to supply a filename.

Keyboard: Ctrl+S

Scale Sector Command

The scale sector command displays the scale sector dialog, allowing the user to specify the percentage by which the selected sector should be scaled.

Scripts Command

The scripts command invokes the scripts dialog box to allow the user to change the script code for the current map.

Scroll Commands

A page scrolling command moves half the current map distance in the specified direction. A line scrolling command moves one-twentieth the current map distance in the specified direction.

Keyboard

Key	Effect
Up Arrow	Scrolls the view up by one line.
Down Arrow	Scrolls the view down by one line.
Left Arrow	Scrolls the view left by one line.
Right Arrow	Scrolls the view right by one line.
Ctrl+Up Arrow	Scrolls the view up by one page.
Ctrl+Down Arrow	Scrolls the view down by one page.
Ctrl+Left Arrow	Scrolls the view left by one page.
Ctrl+Right Arrow	Scrolls the view right by one page.
Page Up	Scrolls the view up by one page.
Page Down	Scrolls the view down by one page.

The current map distance is derived from the current zoom setting. As the zoom increases, scroll commands change position by smaller amounts; as the zoom decreases, scroll commands change position by larger amounts.

Search For Help On Command

The search for help on command allows the user to search the indexed list of available topics in the WadAuthor help file.

Select All Command

The select all command selects all objects in the current map as defined by the current object filter and thing display options. For example, if the object filter is set to linedef, all linedefs within the current map will be selected.

Select Configuration File Command

The select configuration file command displays the select configuration file dialog to allow the user select a different wadgame configuration file.

Select Map Command

The select map command displays the select map dialog to choose which map to edit within a multi-map wadfile.

Set Bookmark Command

The set bookmark commands record the view position and zoom settings for the given bookmark number.

Keyboard: Ctrl+Shift+1 through Ctrl+Shift+9

Set Linedef Length(s) Command

The set linedef length command displays the set linedef length(s) dialog, allowing the user to specify a length for the selected linedefs.

Show Floor Grid Command

The show floor grid command shows or hides the floor image alignment grid. This grid setting may be used to properly place sectors whose floor or ceiling images must be tiled correctly.

Show Grid Command

The show grid command shows or hides the grid. The current grid setting will be used for all alignment operations; it simply will not appear within the map editing view.

Show Thing Bitmaps Command

The show thing bitmaps command enables or disables the use of thing bitmaps at high zoom settings. Enabling thing bitmap display provides a clearer picture of exactly what objects are within a given area, but disabling it improves performance and minimizes graphics resource usage.

Snap To Grid Command

The snap to grid command determines whether or not WadAuthor will align objects with the current grid at the conclusion of a successful drag and drop move operation.

Split Linedef(s) Command

The split linedefs command divides all selected linedefs in half, inserting a new vertex at each center point.

Keyboard: X

Statistics Command

The statistics command displays the statistics dialog which provides some useful information about the current map.

Tags Command

The tags command displays the tags dialog for the current map. This allows the user to add, modify, or remove tags.

Keyboard: Ctrl+T

Thing Display Command

The thing display command displays the thing display dialog to specify which attribute bits a thing should have to be displayed.

Tile Horizontally Command

The tile horizontally command re-arranges the open windows in a tiled fashion in the horizontal direction.

Tile Vertically Command

The tile vertically command re-arranges the open windows in a tiled fashion in the vertical direction.

Tip Of The Day Command

The tip of the day command displays the tip of the day dialog which provides useful information.

Undo Command

The undo command reverses the effects of the last operation. WadAuthor features multiple levels of undo, limited only by system memory.

When a map is saved, the undo history is emptied.

Keyboard: Ctrl+Z
Alt+Backspace

User Defined Tool Command

The user-defined tool commands execute the user-defined command line and arguments.

Window Number Command

The window number command selects from the list of open windows on the Window menu.

Zoom In Command

The zoom in command increases the map editing view viewing scale.

Keyboard: Plus

Zoom In Max Command

The zoom in max command increases the map editing view viewing scale to the allowed maximum.

Keyboard: Ctrl+Plus

Zoom Out Command

The zoom out command decreases the map editing view scale.

Keyboard: Minus

Zoom Out Max Command

The zoom out max command decreases the map editing view scale to the allowed minimum.

Keyboard: Ctrl+Minus

Configuration Files

WadAuthor configuration files (WCF files) contain information specific to a given wadgame. WadAuthor uses them to locate the wadgame's main wadfile, provide the naming convention for new maps, supply a textual name of the game to the user, generate default motifs, and obtain data for all the different types of things, linedefs, and sectors available in the given wadgame. The format is somewhat similar to Windows initialization files (INI files).

The configuration file data is loaded into memory during startup. If you edit the file, you must either exit the application and restart or read use the Select Configuration File command on the File menu to use your changes.

Wadgame Section

A valid WCF file must contain a wadgame section at the top of the file. The following example is taken from the DOOM.WCF file that ships with the product.

```
[WadGame]
Name=DOOM

IWAD=DOOM.WAD
NewMap=E1M1

Directory=C:\DOOM
Run=$_Comspec /c doom.exe -file $_Wadfile -warp $_Wadmap -nosound
```

The Name entry provides a textual name for the wadgame, the IWAD entry provides WadAuthor with the filename of the wadgame's main wadfile, the NewMap entry provides the naming convention for new maps, and the Directory entry provides the location where the wadgame is installed. If any entry (except

Run) in this section is invalid, the wadgame configuration dialog will be invoked to allow the user to supply correct information.

The Run entry cannot easily be validated; fortunately, it rarely requires alteration. The Run entry is used to launch the current wadgame to run the current map. The $_Wadfile, $_Wadmap, and $_Comspec macros are replaced at run time with the fully qualified pathname of the current map, the command-line parameter appropriate to the map name, and the command shell processor. You may wish to change the Run entry if a special PIF file is required, or if you wish to perform batch-file processing when running a map.

The -nosound parameter is present by default to avoid crashing when running wadgames under Windows 3.1, Windows for Workgroups 3.11, or Windows/NT. Most users of Windows95 should be able to safely remove this parameter, allowing the wadgame to supply sound effects and music during play.

The $_Wadmap parameter is formed by eliminating all non-numeric characters from the map name, discarding leading and trailing spaces. For example, E2M7 is transformed into 2 7 at run time. If a wadgame that violates this convention is released, you will need to obtain an updated copy of the software or modify the Run entry manually.

Default Sections

A valid WCF file must also contain some default sections. These are used to generate default sector, door, and stair motifs in the event that a motif file cannot be located at startup. The following examples are taken from the DOOM.WCF file that ships with the product.

```
[Default.Sector]
Above=STARTAN2

Main=STARTAN2
Below=STARTAN2

Ceiling=CEIL3_5
Floor=FLOOR4_8

CeilingHeight=128
FloorHeight=0

Lighting=160
```

```
[Default.Door]
Base=FLAT1

Door=BIGDOOR2
Track=DOORTRAK

Type=1


[Default.Stair]
FloorRunner=STEP1

CeilingRunner=STEP1
Stairwell=METAL

FloorInc=8
CeilingInc=0

LightingInc=0
```

Other Sections

The data for all the different types of things, linedefs, and sectors available in the given wadgame follows these sections. The format of the information is documented within the WCF file and is subject to change in future releases. You may edit this data to suit your preferences, but I strongly urge you to create a backup copy of the original WCF file first and proceed with caution.

Context Menus

Context menus, short for context-sensitive menus, are menus whose options are limited to those that are appropriate within the current context. To display an object's context menu, select the object and press Shift+F10, or click the object with the secondary mouse button. The following sections describe the context menu options that will appear when a given object is clicked.

Map Editing View

Menu Item	**Description**

New Rectangular Sector	Inserts a new square sector the size of a single grid, centered at the menu position. The sector is snapped to grid.
New Polygonal Sector	Displays the new polygonal sector dialog, allowing the user to insert a new sector of a specified radius and number of sides. The sector is centered at the menu position, but not snapped to grid.
New Thing	Displays the new thing dialog, allowing the user to insert a new thing at the menu position.
Zoom In	Increases the map viewing scale.
Zoom Out	Decreases the map viewing scale.
Best Fit	Alters zoom and position to fit the map to the available display area.
Increase Grid	Increases the grid spacing.
Decrease Grid	Decreases the grid spacing.
Show Grid	Shows or hides the grid.
Show Floor Grid	Shows or hides the floor and ceiling image alignment grid.
Show Thing Bitmaps	Enables or disables the use of thing bitmaps at high zoom settings.
Snap To Grid	Enables or disables alignment of objects with the grid.
Map Name	Allows editing of the map name.
Scripts	Allows editing of map script code.
Properties	Allows editing of document properties.

Map Object

Menu Item	**Description**
New Rectangular Sector	Inserts a new square sector the size of a single grid, centered at the menu position. The sector is snapped to grid.
New Polygonal Sector	Displays the new polygonal sector dialog, allowing the user to insert a new sector of a specified radius and number of sides. The sector is centered at the menu position, but not snapped to grid.
New Thing	Displays the new thing dialog, allowing

	the user to insert a new thing at the menu position.
Join Vertexes	This item is only available when exactly two vertexes are selected. Joins the first vertex selected to the second.
Join Linedefs	This item is only available when exactly two one-sided linedefs are selected. Joins the first linedef selected to the second, connecting the two sectors.
Split Linedef(s)	Splits all selected linedefs in half by inserting a new vertex at each center.
Flip Linedef(s)	Displays the flip linedefs dialog, allowing the user to reverse the selected linedef direction and, optionally, sidedef data.
Set Linedef Length(s)	Displays the set linedef length(s) dialog, allowing the user to specify the length and anchor points for the selected linedefs.
Make Linedefs Parallel	Displays the make linedefs parallel dialog, allowing the user to specify the anchor and distance for the selected linedefs.
Align Linedef Textures	Displays the align textures dialog, allowing the user to specify which side to align and which texture to use for sizing information.
Apply Sector Motif	This item is only available when at least one sector is selected. Displays the apply motif dialog and applies the motif selected to the sectors.
Rotate Sector	This item is only available if a single sector is selected. Displays the rotate sector dialog and rotates the sector by the specified angle.
Scale Sector	This item is only available if a single sector is selected. Displays the scale sector dialog and scales the sector by the specified percentage.
Create Motif From Sector	This item is only available if a single sector is selected. It creates a new motif based on the properties of the sector.
Convert Sector To Stairs	This item is only available if a single

	four-sided sector is selected. Displays the convert sector to stairs dialog, allowing the user to select some options and convert the sector to a staircase.
Convert Sector To Door	This item is only available if a single that connects to at least one other sector is selected. Displays the apply motif dialog, allowing the user to select a door motif and convert the sector to a door.
Edit Tags	This item is only available if a single object capable of being tagged is selected. It allows the user to display the objects sharing the given tag.
Edit Raw Data	This item is only available if a single object is selected. It allows the user to edit the actual data saved to disk for a given map object.
Properties	This item is only available if at least one object other than a vertex is selected. Displays the appropriate property editing dialog.

Initialization File

WadAuthor makes use of a standard Windows initialization file (INI file) to retain most of its settings. The file exists in the same directory as WadAuthor and may be safely deleted at any time — it will be recreated the next time WadAuthor is run. The following topics explain each section of the INI file and the various entries found therein.

[Files] Section
[Caching.Images] Section

[General] Section
[Recent File List] Section

[Views] Section
[Window.X] and [Dialog.X] Sections

Keyboard Interface

Although WadAuthor is, perhaps, at its best with a mouse or other pointing device, I have always been most efficient when working from the keyboard. The following sections describe the keyboard interfaces available within WadAuthor.

Global Commands

The following commands are available from anywhere within the application.

Key	Command	Description
F1	Context Help	Displays the help file for the current context.
Shift+F1	Context Help	Enables mouse-driven context-sensitive help.

Main Frame Window

The main frame window is the main application window. It is the window that contains all the open documents and provides the toolbar and status bar. The following commands are available when working with the main frame window.

Key	Command	Description
Ctrl+N	New	Creates a new document.
Ctrl+O	Open	Opens an existing document.
Alt+F4	Exit	Exits the application.

Map Editing View

Map editing views provide the primary WadAuthor functionality. They display wadmaps and allow the user to manipulate the objects within.

Map Editing View - File Commands

Key	Command	Description
Ctrl+N	New	Creates a new document.

Ctrl+O	Open	Opens an existing document.
Ctrl+F4	Close	Closes the current document.
Ctrl+S	Save	Saves the current document.
Ctrl+P	Print	Prints the current document.

Map Editing View - Tool Commands

Key	Command	Description
Ctrl+M	Motifs	Displays the motifs dialog.
Ctrl+T	Tags	Displays the tags dialog.
Ctrl+K	Check Map	Checks the current map for errors.
Ctrl+R	Run Map	Runs the current map.

Map Editing View - View Commands

Key	Command	Description
Plus	Zoom In	Increases the viewing scale.
Minus	Zoom Out	Decreases the viewing scale.
Home	Best Fit	Alters zoom and position to fit the map to the available display area.
Shift+Plus or]	Increase Grid	Increases the grid spacing.
Shift+Minus or [	Decrease Grid	Decreases the grid spacing.
Ctrl+Plus	Zoom In Max	Zooms in to the maximum allowed scale.
Ctrl+Minus	Zoom Out Max	Zooms out to the minimum allowed scale.
Up Arrow	Scroll Up	Scrolls the view up by one line.

Down Arrow	Scroll Down	Scrolls the view down by one line.
Left Arrow	Scroll Left	Scrolls the view left by one line.
Right Arrow	Scroll Right	Scrolls the view right by one line.
Ctrl+Up Arrow	Page Up	Scrolls the view up by one page.
Ctrl+Down Arrow	Page Down	Scrolls the view down by one page.
Ctrl+Left Arrow	Page Left	Scrolls the view left by one page.
Ctrl+Right Arrow	Page Right	Scrolls the view right by one page.
Page Up	Page Up	Scrolls the view up by one page.
Page Down	Page Down	Scrolls the view down by one page.
F5	Refresh	Redraws the current map.
Z	Object Zoom	Zooms the map to best fit the current object.
Shift+Z	Clear Object Zoom	Restores the original map settings prior to the object zoom command.
Ctrl+U	Full Screen	Toggles the state of full screen display.

Map Editing View - Selection Commands

Key	Command	Description
Tab	Next Object	Sets the current object to the next object.
Shift+Tab	Previous Object	Sets the current object to the previous object.
Ctrl+F	Object Filter	Sets the object selection filter.
Ctrl+M	Select Map	Selects a map from a multi-map wadfile.
F2	Center On Object	Centers the current map display on a specified object.

Shift+F10	Context Menu	Displays the context menu for the selection.
T	Filter Things	Sets the object filter to things.
V	Object Filter Commands	Sets the object filter to vertexes.
L	Object Filter Commands	Sets the object filter to linedefs.
S	Filter Sectors	Sets the object filter to sectors.
N	Filter None	Sets the object filter to none.

The following thing display commands are only valid for wadgames other than Hexen.

1, 2	Thing Display	Toggles the level 1/2 attribute for thing display.
3	Thing Display	Toggles the level 3 attribute for thing display.
4, 5	Thing Display	Toggles the level 4/5 attribute for thing display.
D	Thing Display	Toggles the deaf attribute for thing display.
M	Thing Display	Toggles the multi-player attribute for thing display.

The following thing display commands are only valid for Hexen.

1, 2	Thing Display	Toggles the level 1/2 attribute for thing display.
3	Thing Display	Toggles the level 3 attribute for thing display.
4, 5	Thing Display	Toggles the level 4/5 attribute for thing display.
D	Thing Display	Toggles the deaf

		attribute for thing display.
R	Thing Display	Toggles the dormant attribute for thing display
F	Thing Display	Toggles the fighter attribute for thing display
C	Thing Display	Toggles the cleric attribute for thing display
M	Thing Display	Toggles the mage attribute for thing display.
I	Thing Display	Toggles the single-play attribute for thing display
O	Thing Display	Toggles the cooperative-play attribute for thing display
H	Thing Display	Toggles the deathmatch-play attribute for thing display

Map Editing View - Edit Commands

Key	Command	Description
Delete	Delete	Removes the contents of the selection.
Insert	Insert Object	Inserts an object into the map.
Alt+Enter	Properties	Edits properties of the selection.
Ctrl+C	Copy	Copies the contents of the selection to the clipboard.
Ctrl+V	Paste	Pastes the contents of the clipboard into the current document.
Ctrl+X	Cut	Removes the contents of the selection to the clipboard.
Ctrl+Z	Undo	Reverses the effects of the last editing operation.
J	Join Vertexes or Join Linedefs	Joins selected vertexes or linedefs.

X	Split Linedefs	Splits selected linedefs.

Menu Commands

File Menu

Clicking a top level menu expands or contracts the list of menu items; clicking a menu item displays the help topic for the item.

File Menu

New
Open

Select Configuration File
Close

Save
Save As

Print
Print Preview

Print Setup
Most Recently Used Files

Exit

Edit Menu

Clicking a top level menu expands or contracts the list of menu items; clicking a menu item displays the help topic for the item.

Edit Menu

Undo

Cut
Copy

Paste
Delete

Select All

Force Node Build
Properties

View Menu

Full Screen

Zoom In
Zoom Out

Best Fit

Go To Bookmark

Set Bookmark
Increase Grid

Decrease Grid
Object Filter

Thing Display
Select Map

Statistics

Tools Menu

Customize

Options
Motifs

Tags
Check Map

Run Map

Window Menu

New Window

Cascade
Tile Horizontally

Tile Vertically
Arrange Icons

Refresh
Window Numbers

Help Menu

Help Topics
Search For Help On

How To Use Help

Register
About WadAuthor

Motifs

Motifs allow the map designer to more easily maintain a consistent look and feel when creating new architecture. When new sectors are created, the current motif is applied to all appropriate sidedefs. Motifs may also be applied to existing sectors if the user has a decorative change of heart.

Because motifs are directly tied to a specific wadgame, the available motifs will depend upon the current configuration file. WadAuthor looks for motif files (MTF files) in the directory from which it runs when reading data from the current configuration file. It uses the base name of the wadgame's main wadfile to determine which motif file to use. For example, if the wadgame's main wadfile is DOOM.WAD, WadAuthor will look for DOOM.MTF. If WadAuthor cannot find a motif file for the current wadgame, it will create a default motif file.

To edit the list of available motifs or to select the current motif, choose the Motifs... option from the Tools menu. To manually apply a motif, select the sectors to which the motif should be applied, and choose the Apply Motif... option from the context menu.

Mouse Interface

WadAuthor is, perhaps, at its best when used with a mouse or other pointing device. I find the mouse to be the most direct means for some operations. The following sections describe the mouse interfaces available within WadAuthor.

Map Editing View

Map editing views provide the primary WadAuthor functionality. They display wadmaps and allow the user to manipulate the objects within.

Primary Mouse Button (Usually Left)

Action	Effect
Click	Clicking an object will do one of two things: if selected, a drag and drop operation will begin; otherwise, the object will become selected. Clicking an empty area of the map will clear the selection.
Ctrl+Click	Holding down the control key while clicking an object will toggle its selection state; clicking an empty area of the map has no effect.
Alt+Click	Holding down the alt key while clicking allows for very precise selection. If objects are too close together for the software to distinguish, use this method. Whenever more than one object is “near enough,” a dialog box will allow the user to pick the specific object of interest.
Shift+Click	Holding down the shift key begins a rubberband selection. This allows the

	user to select a range of objects in a single operation. If the user does not drag a rubberband selection, the object clicked will be added to the selection.
Double click	A double click will clear the selection, select the object clicked (if any), and display the properties dialog for the given object.
Ctrl+Double click	Holding down the control key while double-clicking will center the map editing view at the given location.

Secondary Mouse Button (Usually Right)

Action	Effect
Click	Clicking an object will do one of two things: if selected, the context menu for all selected objects will be displayed; otherwise, the object will become selected and the context menu for the specific object will be displayed. Clicking an empty area of the map will display the context menu for the map itself.
Alt+Click	As with the primary mouse button, holding down the alt key while clicking allows for very precise selection. Refer to the primary mouse button Alt+Click explanation for more detail.

Properties

The term properties refers to the innate attributes or qualities of an object. For example, a sector has a type, a floor image and height, a ceiling image and height, a lighting level, and a tag number. Property editing refers to the process of changing object properties. WadAuthor provides very powerful property editing through various dialog boxes. To edit an object’s properties, select the object and choose Properties... from the context menu, choose Properties... from the Edit menu, or press Alt+Enter to display the appropriate dialog box.

Using Custom Images

WadAuthor supports the use of additional images not present in the main wadgame wadfile. When a wadfile is opened, WadAuthor will default to using any images contained within the wadfile adding to or overriding the images contained in the main wadgame wadfile. If the custom images for a given wadfile are contained in a separate wadfile, for whatever reason, you must tell WadAuthor which wadfile to use for the additional images. You may configure how WadAuthor uses custom images via the document properties dialog.

The images applied to linedef sidedefs (usually called textures) are actually composed of a number of smaller images (usually called wall patches) which are referenced by number. This allowed the original game designers to avoid wasting space within the main wadfile.

If additional textures are supplied without a corresponding list of panels, WadAuthor will provide a warning message. If the panels present in the main wadgame wadfile are sufficient to create each texture, no problems will occur. If, however, the new textures require new panels, WadAuthor will not be able to display the images. If the warning message appears, it is strongly suggested that you include the panels in the wadfile containing the additional textures.

How To Use Custom Images

1 Open the wadfile for which custom images are needed.
2 Choose the Properties option from the Edit menu to display the document properties dialog.
3 Change to the Images page if necessary.
4 Select the desired image handling option. If using images in an external wadfile, be sure to specify a valid file name.

Registration

WadAuthor is not free. If you use it, you must pay for it. I believe in the shareware system because the “try before you buy” approach is a great way to make sure you’re getting something you want. In allowing you the free use of this program for a time, I am extending you a degree of trust; please do not violate this trust. Remember, only you can make shareware work.

You may use WadAuthor free of charge for 15 days. If, at the end of this time period, you decide that WadAuthor is not for you, please stop using it and remove it from your system — better yet, pass it along to a friend (or enemy if you really don't like it). If you decide to keep WadAuthor, you must register through one of the means listed at the bottom of this topic.

In exchange for registration, you will receive technical support, upgrades for the life of the product, access to handy wadfile utilities, and the author's gratitude — not to mention that pleasant lack of guilt that comes with a clear conscience! By registering WadAuthor, you will also get the chance to shape its future through your comments and complaints. Upon receipt of payment, you will be sent the information required to authorize the software license.

Snail Mail

To register WadAuthor by snail mail, please send a check or money order, payable to Williston Consulting, in the amount of $20 (U.S.) to the following address.

Williston Consulting
7732 24th. St.
Westminster, CA 92683

Along with your payment, you must enclose a message including the product name (WadAuthor), your name, your company (if any), address, e-mail address (if any), and any comments you care to add. A pre-prepared order form is available by clicking on this topic's title bitmap or by clicking below.

Registration payments from countries outside the United States of America must be capable of being cashed by a U.S. bank. I will make every attempt to cash whatever I receive, but if I cannot do so, I will be forced to return your order. Alternatively, you may contact me to make any special arrangements for wire transfer of funds. I do not wish to discourage potential foreign users in any way; I simply have not found an alternative.

WadAuthor Order Form

Name: ____

Company: ____

Postal Address:

____

____

____

E-mail Address:

____

Comments: ____

____

____

Copies: ____

Order Checklist

- Make sure that your name and address are clearly legible.
- Make sure you have indicated the number of copies desired.
- Make sure that your payment to Williston Consulting is enclosed.
- Make sure sufficient postage is affixed to the envelope.
- Make sure the address matches the address shown below.

Williston Consulting
7732 24th. St.
Westminster, CA 92683

Troubleshooting

I really hope you're not reading this. There cannot possibly be any bugs in my code, right? (*sigh*) Oh well, back to reality. The following section contains solutions for common problems, and instructions for getting technical support.

The Graphics Are Wrong

If the wadfile images are not displayed or are displayed incorrectly, it is probably due to an incompatibility between your video card and the image acceleration code WadAuthor employs. Try switching to a different color depth to see if the problem goes away. If the problem does not go away, please consult the related topics for getting technical support.

The Run Map Feature Fails

If the run map feature fails, it is probably due to one of the following reasons.

The run command is invalid. Open the current wadgame configuration file in notepad (or some other text editor), carefully examine the Run setting in the WadGame section, and fix any errors before trying again.

You do not have enough memory to launch the wadgame. Make more memory available or exit Windows before running your map.

There is some kind of problem with the map you're trying to run. Select Check Map... from the Tools menu and fix any problems that WadAuthor finds before trying again.

The WARUN.EXE utility does not exist in the same directory from which WadAuthor is running. WadAuthor uses this utility to launch the wadgame, and it must be present in the same directory as the WadAuthor executable.

The run map feature was very difficult to support in a consistent manner across Windows, Windows95, and Windows/NT. It is possible that it will fail, even when none of the reasons given above apply. If you simply cannot get it to work, consult the related topics for getting technical support.

Why Is There No Sound When Running A Map?

The WadAuthor configuration files disable the sound by default to avoid problems. If you are running WadAuthor under Windows95, however, it is quite possible that your hardware is capable of providing sound. To test this, edit the wadgame configuration file with notepad, or some other text editor, and remove the -nosound option from the Run entry in the Wadgame section.

Be sure to make a backup of the original configuration file before editing it.

Things Are Missing

One of the most common problems for new wadfile designers is that of missing things. A missing thing, in this context, is a thing that is displayed by WadAuthor, but not present during actual game play.

The most common reason for this is that the thing attributes are set in such a manner that they prevent the thing from appearing. For example, a thing with the multi attribute will not appear in a single player game.

Getting Technical Support

If you are a **registered** user of WadAuthor, or if you are having problems getting the shareware version installed and working, please contact me through one of the means listed below.

Electronic Mail	wconsult@ix.netcom.com
Snail Mail	Williston Consulting329 N. Orange Ave.Fullerton CA 92833

Make sure to include the following information in your message.

- The product with which you are having the problem.
- The version of Windows that you are using.
- If a message is displayed, please include the full text including the caption of the dialog box.
- If the problem is reproducible, please include whatever steps are necessary to recreate the problem.

As a conscientious author, I will make every attempt to fix any bugs reported to me. It is very difficult, however, to fix bugs that I cannot reproduce. So, the more information you can provide, the greater chance I have of finding and squashing the bug.