Abiathar

Documentation for v2.3

Table of Contents

Concepts

Whether you're totally new to the Keen modding scene or a veteran content creator, you should take a minute to read this section. Experienced Galaxy modders may skip the information on how Commander Keen levels are laid out. However, the majority of this section is dedicated to Abiathar-specific concepts and features.

Keen: Galaxy Levels

Many thanks to Andy Durdin for publishing "Plumbing the Depths of Keen", on which this article and the core of Abiathar is based.

Unlike the single-level Vorticons maps, Galaxy and Dreams levels are stored together in (kind of) one file. You probably have seen this; it's called `GAMEMAPS.CK#` for Galaxy and `KDREAMS.MAP` for, obviously, Dreams. Since the levels are all squished into this file, there needs to be a way to map offsets to levels. There is: it's called the map header. Usually called `MAPHEAD` when external, it is found only inside the game executables. To modify levels, we have to save modified version of both these files. Keen Dreams also has another file, `MAPDICT.KDR`, also internal, that helps decompress the maps. All internal files are patched in with CKPatch, which is bundled with Abiathar.

Galaxy levels are more complicated than those of the Vorticons trilogy. Instead of having one tile plane and one sprite plane, they have two tile planes and one extra-data plane. The first two are called **background** and **foreground** planes. Only the foreground can interact with Keen or any other sprites. The last plane, the **infoplane**, contains pseudo-tiles that turn into sprites when the map is loaded. It also contains things like the paths for the moving platforms and pointers from switches and gem holders to platforms and gates. These pseudo-tiles look strange; they are defined in the game graphics and therefore appear the same in all editors.

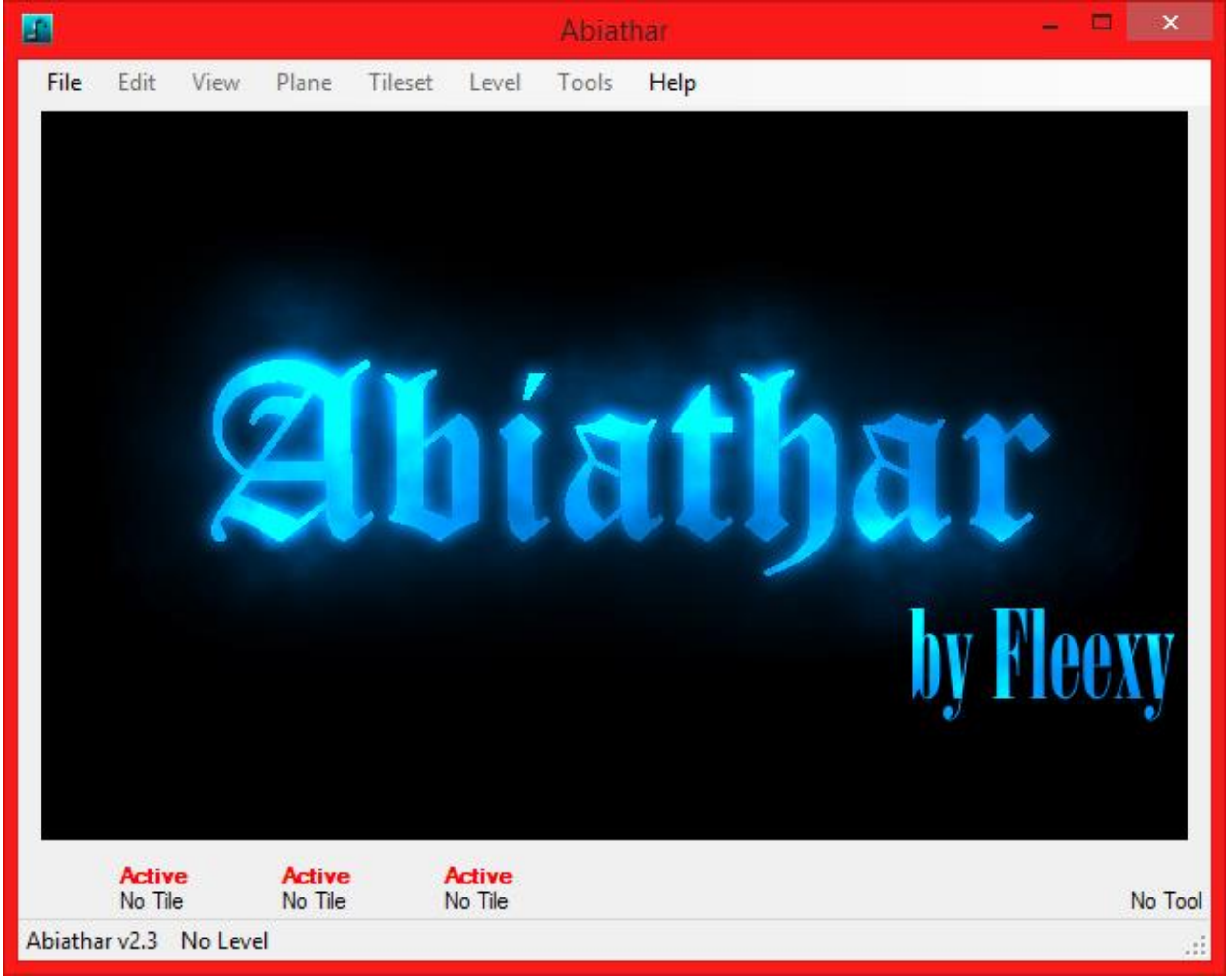
Pointers, such as those from switches to platforms, technically have tile IDs past the end of the standard infoplane. Therefore, Abiathar and other editors display them with their ID in hexadecimal (base 16 numbering system). Abiathar is also able to render the links as lines from the controller to the target. Infoplane values that are inside the normal range are, by default, given their own tileset despite being technically part of the foreground set. Abiathar in default configuration and other level editors give each plane its own tileset from which tiles can be picked and placed. Though it is possible to place the infoplane pseudo-tiles in the foreground layer as real tiles, background tiles cannot be placed in the foreground/infoplane or vice versa.

A two-tile wide border, invisible in the game, surrounds all levels. It is necessary for Keen's renderer. If Keen moves outside the normal level space into the edge space, it will count as an exit. This can be seen in the "Border Village" of Keen 4.

Setting up Abiathar

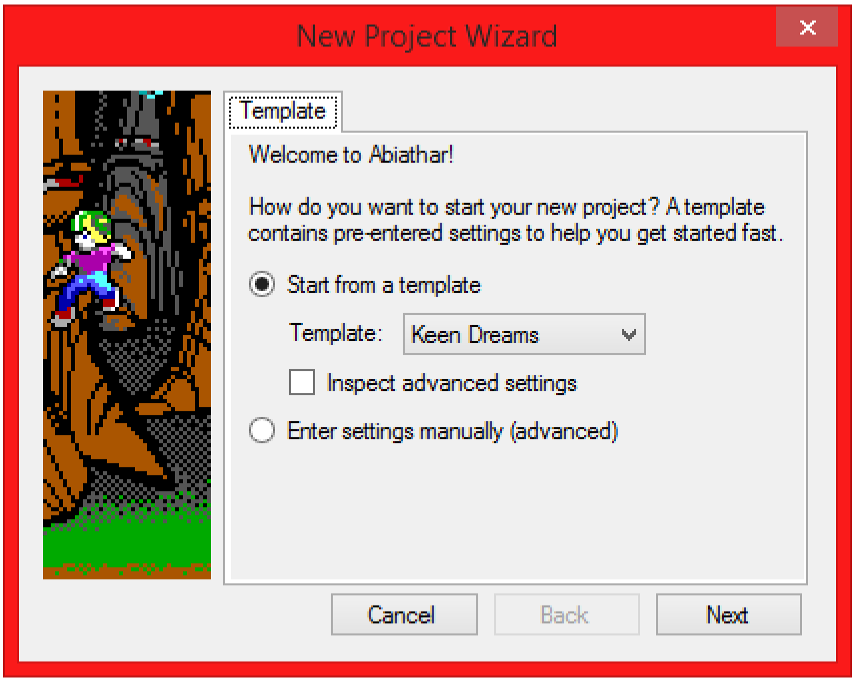
All you need to start editing levels with Abiathar is the program itself and, if you have the uncompact version, the supporting DLLs. **You don't need extracted game graphics or even the default level files to start with Abiathar!**

When you launch Abiathar, you will find a mostly blank window with menus mostly disabled.



Before allowing Abiathar to explode into a panorama of power, let's go over what we have here. Along the bottom is the **selected tile bay**. When a level set is opened, the blank spaces to the left of the "Active / No Tile" blocks will show the selected tile for the planes in the following order: background, foreground, infoplane. "No Tile" will become the ID of the tile in hexadecimal and "Active" may change to display the degree to which the plane can be modified. "No Tool" will later display the tool being used. The menu strip will not change except for becoming usable, though if you have already installed extensions, you will see an additional "Extensions" menu to the right.

To start an editing project, choose “New” under File. The New Project Wizard appears.



Welcome to Abiathar!

How do you want to start your new project? A template contains pre-entered settings to help you get started fast.

● Start from a template

Template: Keen Dreams

□ Inspect advanced settings

○ Enter settings manually (advanced)

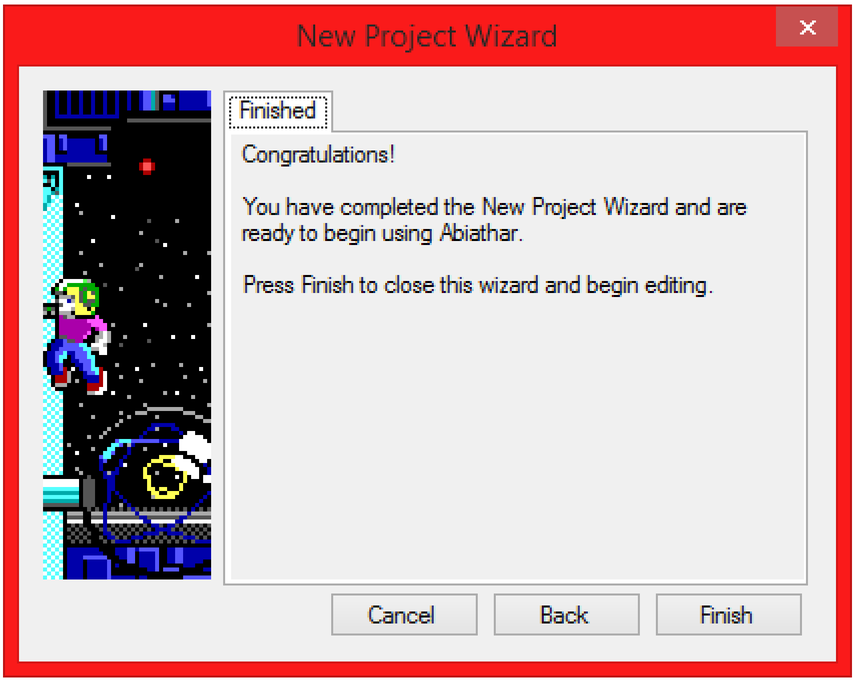
This wizard will walk you through the creation of your dependency file (project). It contains explanatory text and context-sensitive help to help you choose the appropriate options. Using a template will fill in difficult settings for you and allow you to start from original game resources, even if they are not on disk.

Important note: Loading Dreams levels outputted by The Omegamatic requires first putting the map header through FxTomDMh.exe, which can be obtained in the File Emitter, which is detailed elsewhere in this help file.

Note: The CKPatch %maphead command deliberately cuts off the tileinfo in a combined maphead/tileinfo file. Therefore, it is recommended that you have a separate tileinfo file for modding the normal episodes.

Once your resources have been selected and you have finished the wizard, press Finish to load them and begin editing. If you later choose to move to other resources, you may go through the New Project Wizard again or select Project Settings under Edit.

Basics of Use

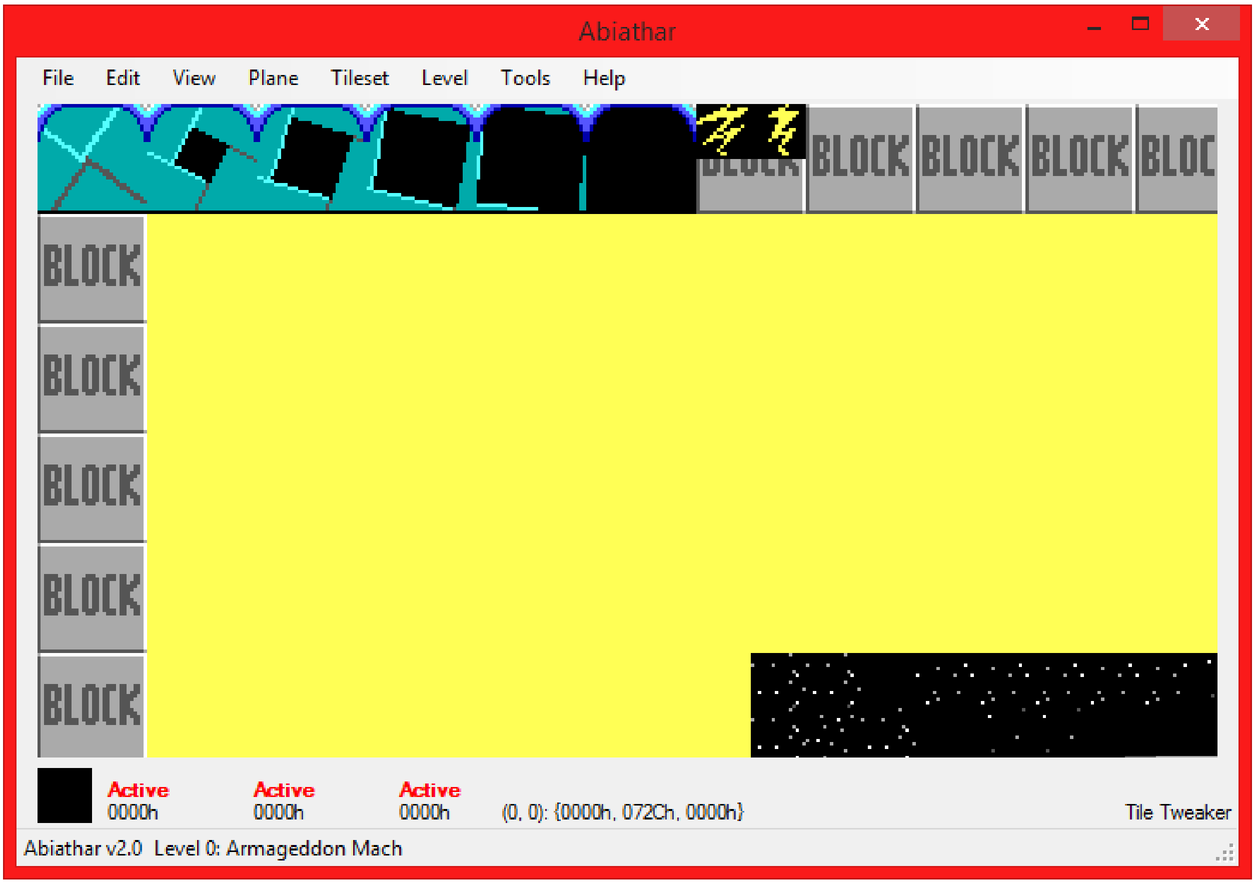


Congratulations!

You have completed the New Project Wizard and are ready to begin using Abiathar.

Press Finish to close this wizard and begin editing.

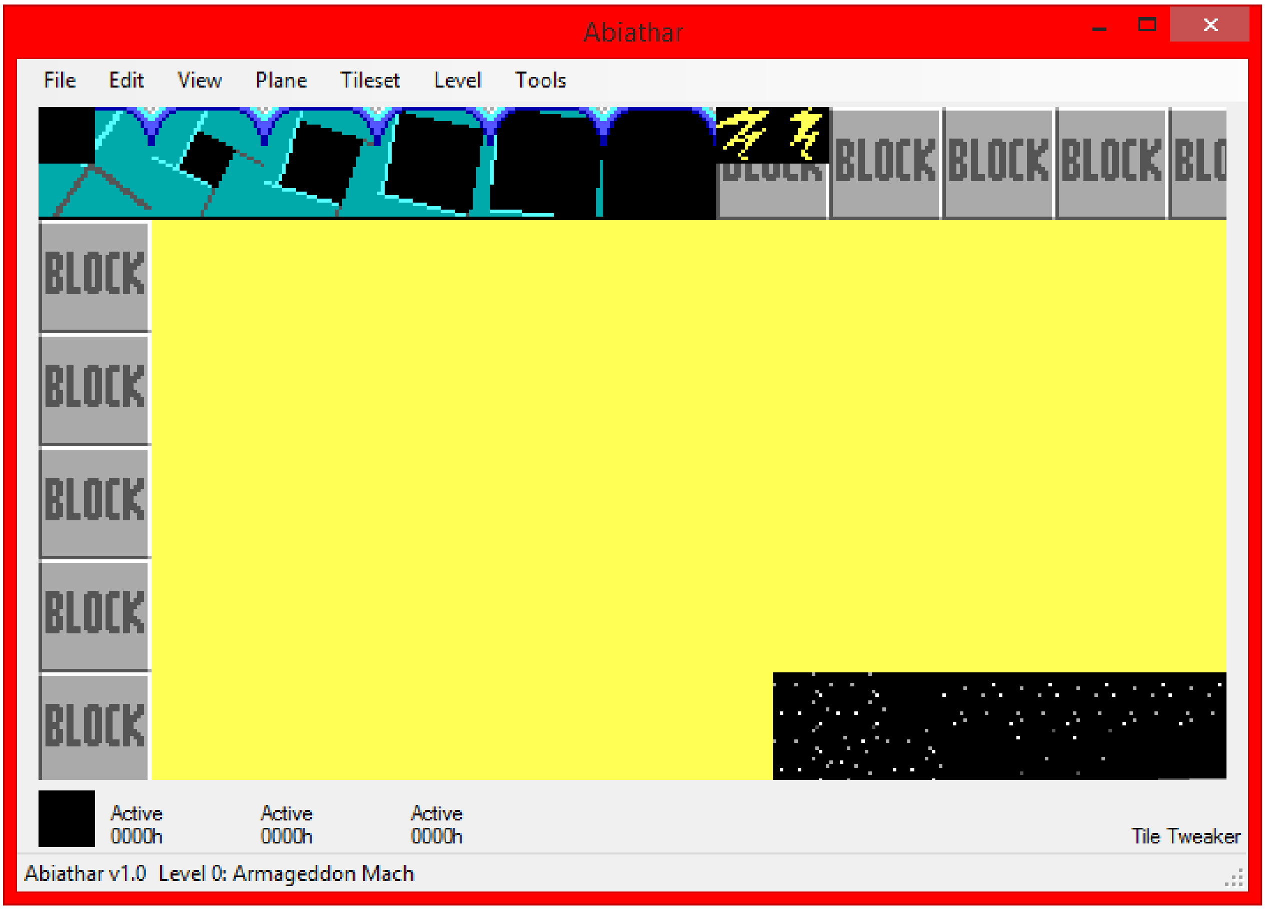
Once you finish with the New Project Wizard, Abiathar will become usable for editing. In this example, the default Keen 5 files were used.



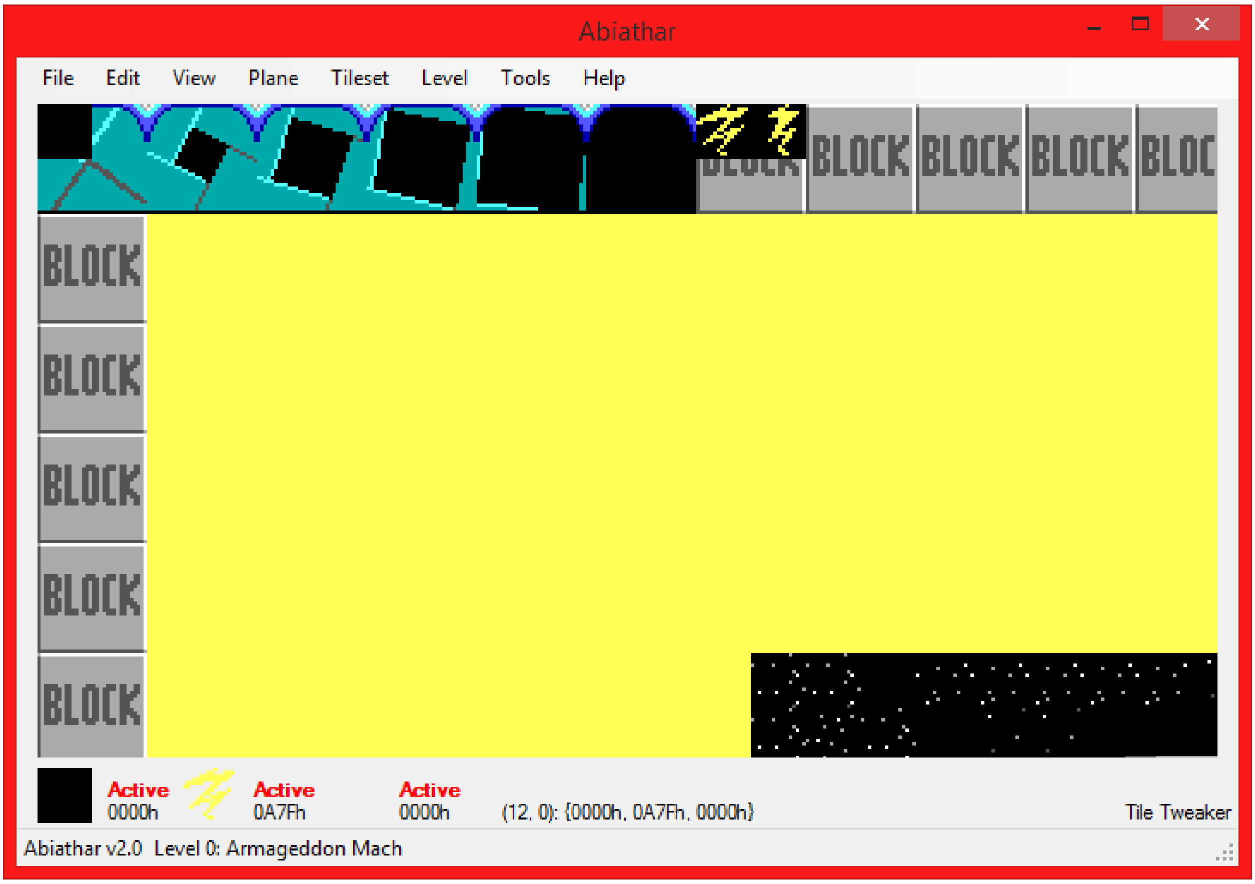
(0, 0): {0000h, 072Ch, 0000h}

Abiathar v2.0 Level 0: Armageddon Mach

As you can see, the interface has lit up and a tool has become active. This tool, the Tile Tweaker, provides the most basic functions: placing individual tiles or copying in-level tiles to the selected tile bay. More information about tools can be found in the appropriate section of this help file. This is the result of clicking in the far upper left:



The 0th tile has been set for each plane on that tile. This is the result of right-clicking on the first lightning bolt:



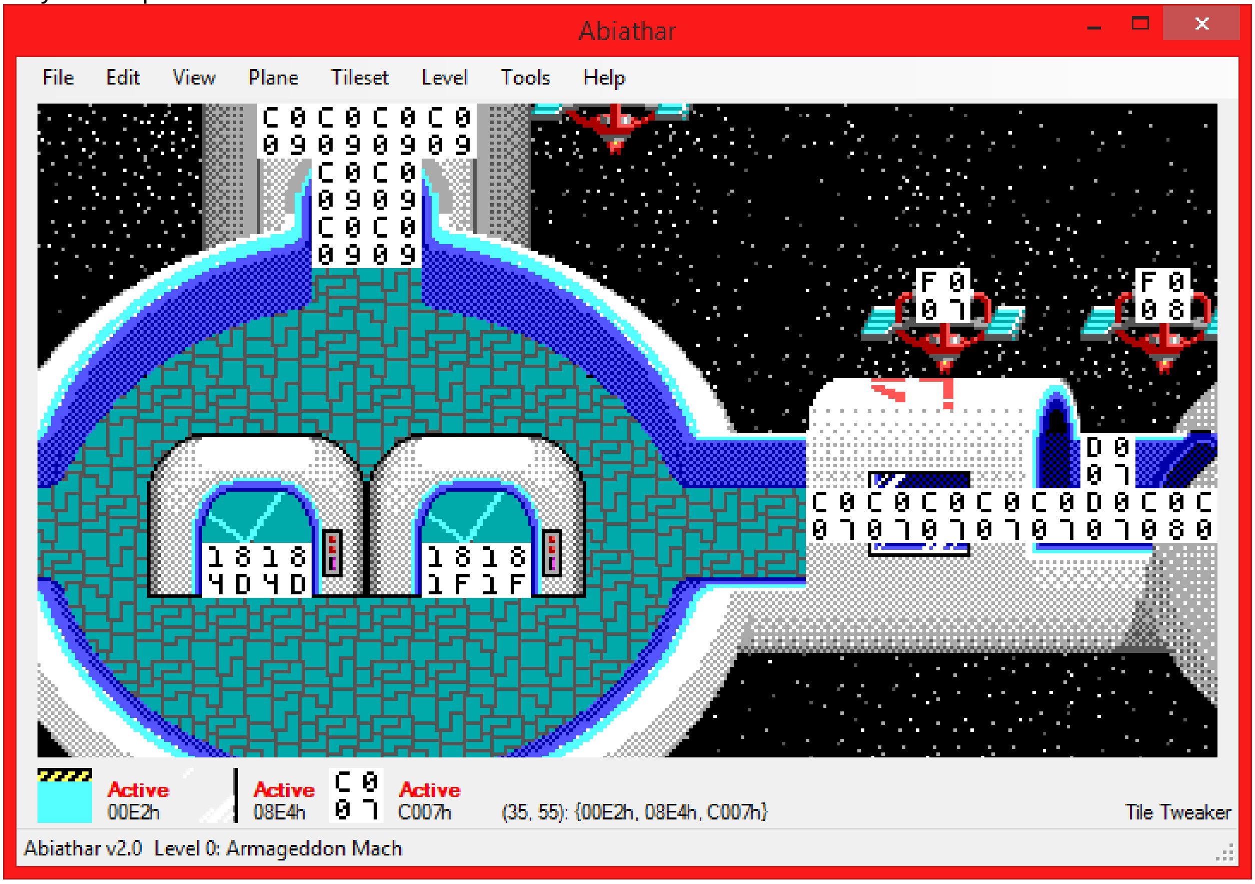
(12, 0): {0000h, 0A7Fh, 0000h}

Every plane's value at that tile has been copied to the corresponding tile bay.

Very important is the ability to move throughout the level. Abiathar has two ways of doing so: the keyboard and the scroll wheel.

- Keyboard
  - Arrow keys move the viewer 1 tile
  - Holding Control changes the offset to 5
  - Holding Shift multiplies the offset by 3
  - Pressing Backspace moves back to the upper left
- Scroll wheel
  - Each notch of the wheel moves 2 tiles vertically
  - Holding Control makes it move horizontally

Abiathar by default does not place a limit on scrolling; it is possible to move out of the level, though tiles may not be placed.

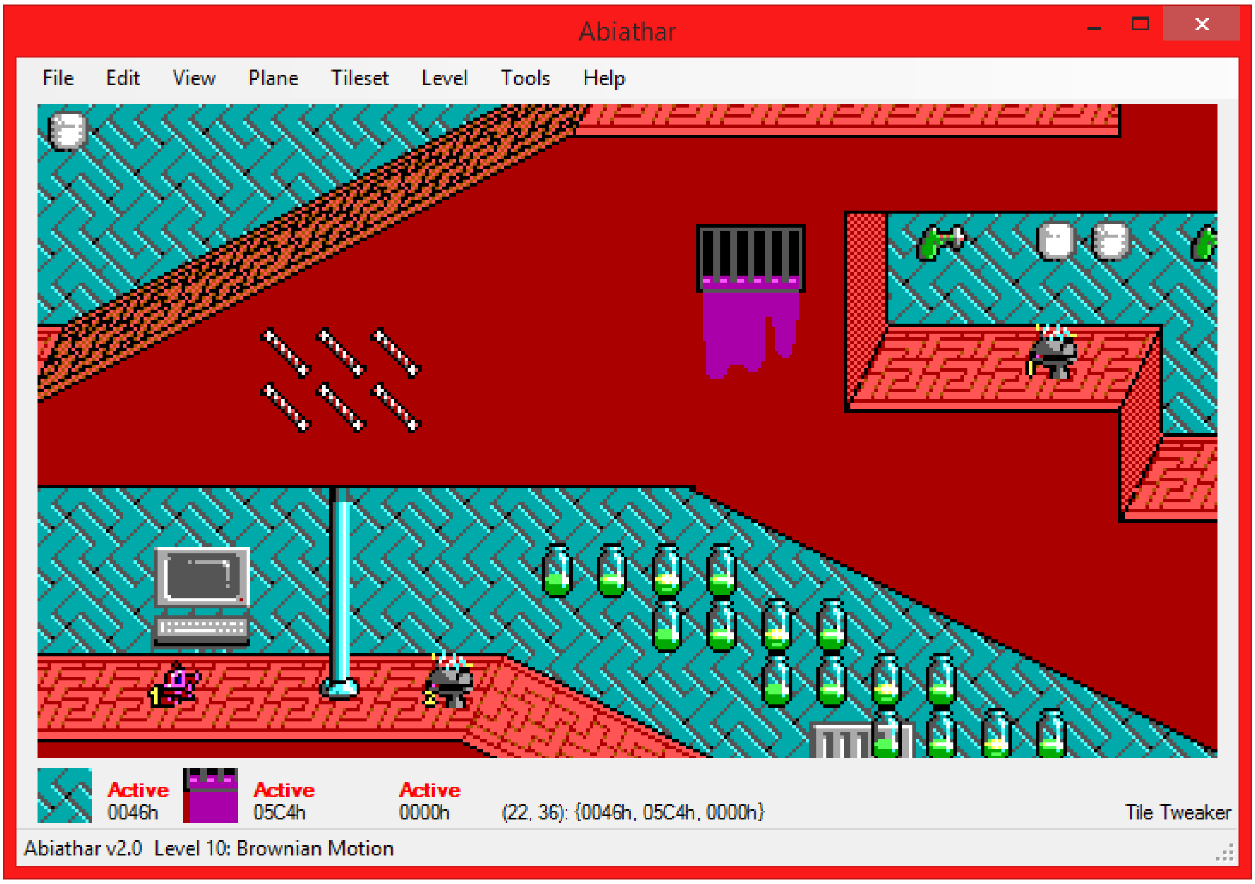


(35, 55): {00E2h, 08E4h, C007h}

Abiathar v2.0 Level 0: Armageddon Mach

Infoplane values outside of the known range are rendered as their numeric ID.

To view and modify a different level, select it from the Level menu, which also contains level operations.

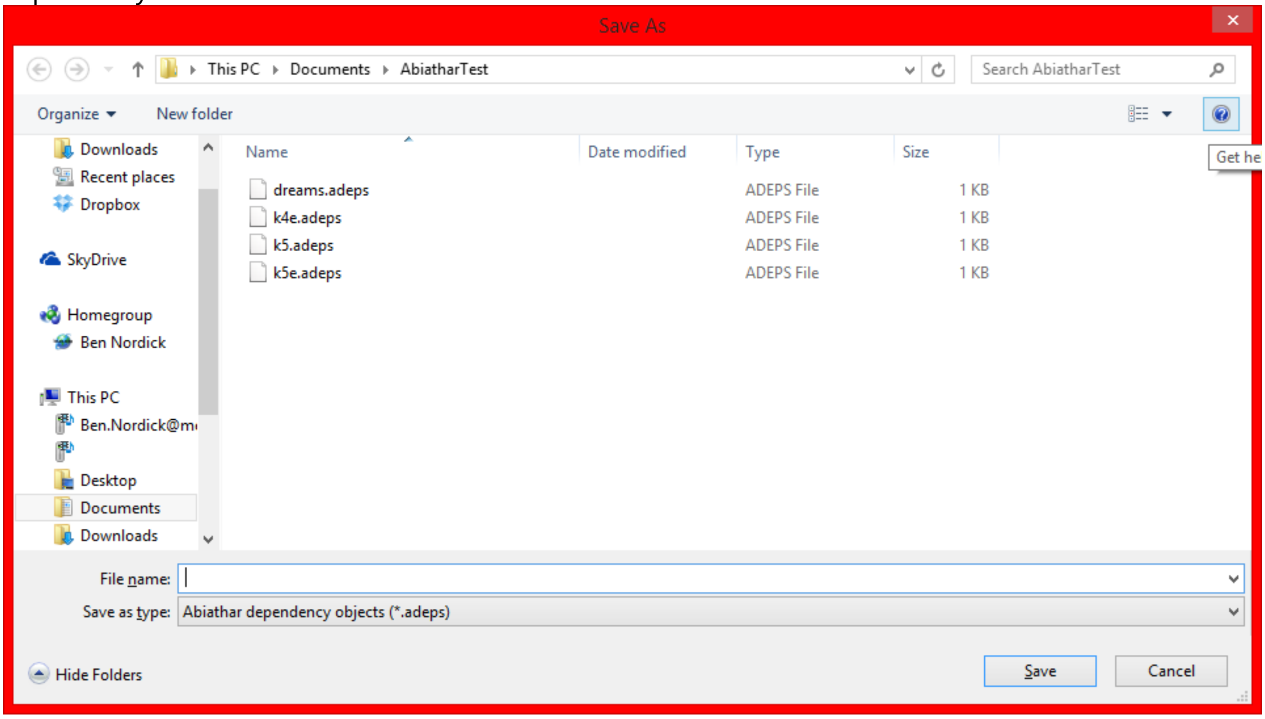


Active 0046h Active 05C4h Active 0000h (22, 36): {0046h, 05C4h, 0000h}

Abiathar v2.0 Level 10: Brownian Motion

Files & Storage

When you have made enough progress after running the Dependency Collector to make your level set worthy of a few kilobytes of disk storage, use “Save” under File to - obviously - save it. The first time you do this, you will be presented with exactly one dialog box, asking where you would like to place the dependency file.

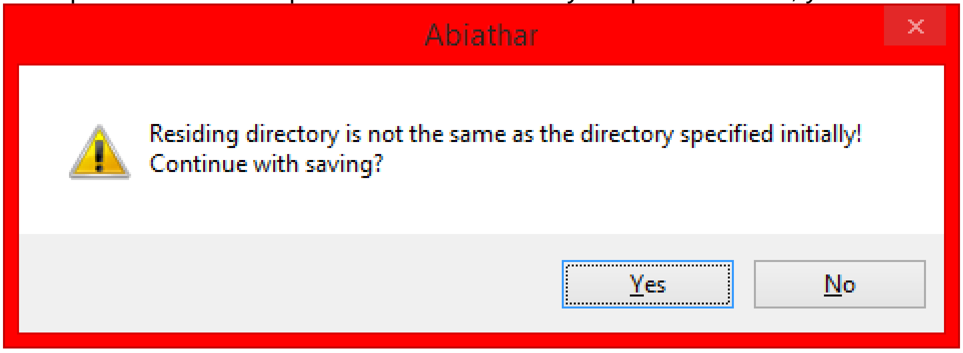


Name	Date modified	Type	Size
dreams.adeps		ADEPS File	1 KB
k4e.adeps		ADEPS File	1 KB
k5.adeps		ADEPS File	1 KB
k5e.adeps		ADEPS File	1 KB

File name:

Save as type: Abiathar dependency objects (*.adeps)

When you type a name and click Save, your level set's settings will be written to an .adeps file. As seen in the above image, these files are very small. That's because they only contain references to the real resources files, which are automatically saved as the names you specified in the New Project Wizard. If you attempt to save the .adeps outside of the folder you specified there, you will receive a warning:



Residing directory is not the same as the directory specified initially!
Continue with saving?

Since Abiathar collects its resources ("dependencies") from simple file names, it needs to be able to access them given only their name and the path of the dependency file. If you click No, no files will be created or saved; you may click Save again and choose the default location, as suggested by Abiathar. If you choose Yes, Abiathar will place all new files in the directory of the dependency file, so it may lose access to unmodified resources, such as graphics.

Even if you make a mistake when saving the first .adeps, it is always possible to run the New Project Wizard again and place the resources and dependency object where you intended.

After you have created an .adeps, you can use the Open command to resume editing later. The Open dialog will only accept .adeps files.

Note: Abiathar-created resource files (but not dependency files) can be opened with other editors. Exception: Keen Dreams map headers might not be recognized by The Omegamatic due to its use of an incomplete format specification.

Note: Graphics are not modified after saving with Abiathar.

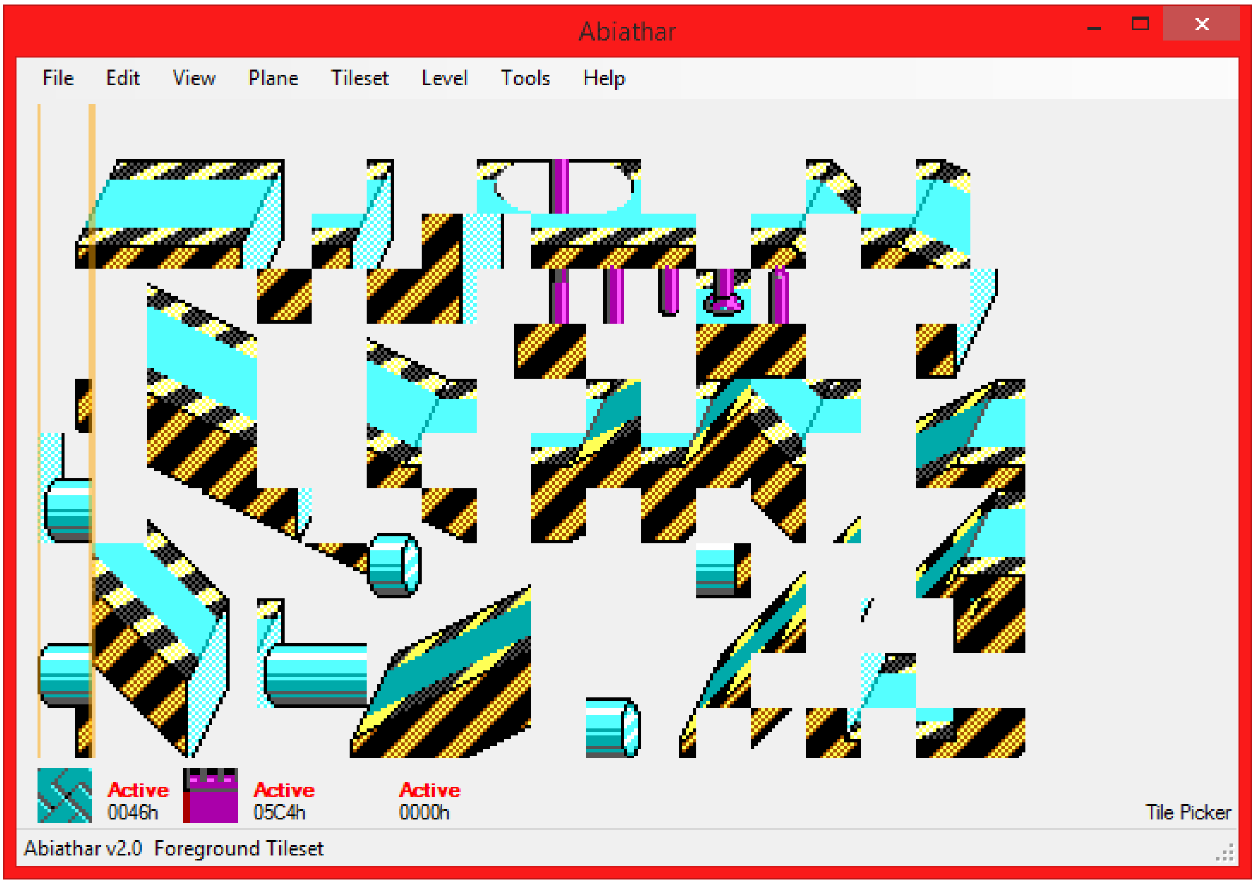
Common Tasks

This section guides you through the most frequently used features in Abiathar. Unusual tasks and esoteric shortcuts are documented in the next major section. Other tools and ways to modify the levels can be found in the Tools section.

Tile Palette

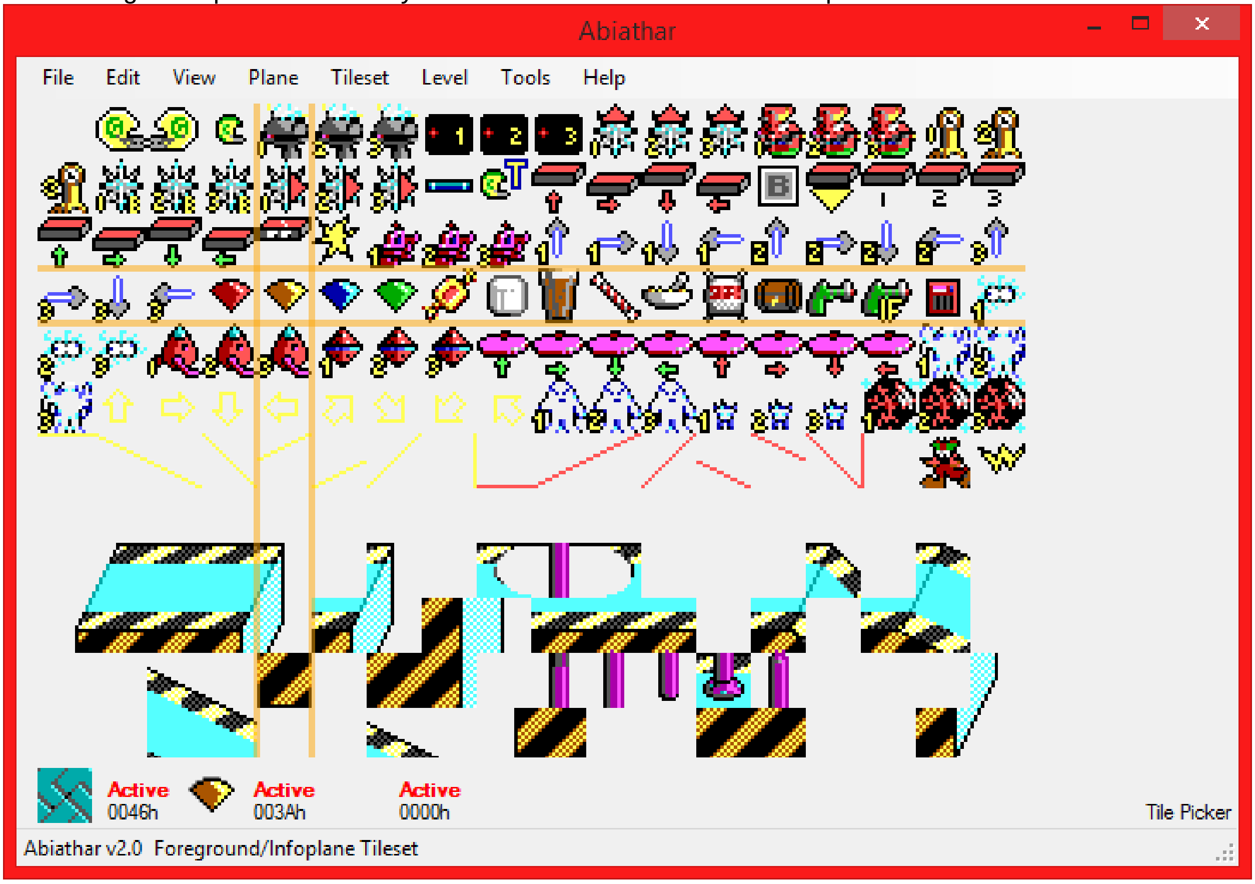
One of the most important parts of a level editor is its tile palette. It gives you, the modder, access to your most important tools.

In Abiathar, the tile palette can be called either with the Tileset menu or by pressing the appropriate number key after pressing Space, which switches to the tile palette view. 1 calls the background set, 2 calls the foreground set, and 3 calls the set of infoplane pseudo-tiles. Simply click a tile to copy it to the respective bay. When finished, press Space to return to your level. If you do not want to press Space to access the palettes, you may at any time press 7 to open the background palette, 8 for the foreground, or 9 for the infoplane.



Abiathar v2.0 Foreground Tileset

A downfall of previous Galaxy editors is that they always split the foreground and infoplane, making it impossible to squeeze those last few rows out of the tileset. To help you add the most content to your mod, Abiathar provides an amazing feature: Assimilate Infoplane. Upon activation, AI will add the infoplane tiles to the foreground palette. You may select tiles as normal from the infoplane tileset.



Abiathar v2.0 Foreground/Infoplane Tileset

Plane States

Editing with all planes active all the time would be very difficult. To help you see and change only what you want to, Abiathar has several systems to control the status of the planes. Each of the three planes can hold one of three states:

- Active: visible and editable (default)
- Locked: only visible
- Hidden: not visible or editable

These can be controlled in several ways. The Plane menu provides fine-adjustment tools and commands to modify the state of multiple or all the planes at once. However, those can be cumbersome when switching planes frequently, so Abiathar also has keyboard shortcuts.

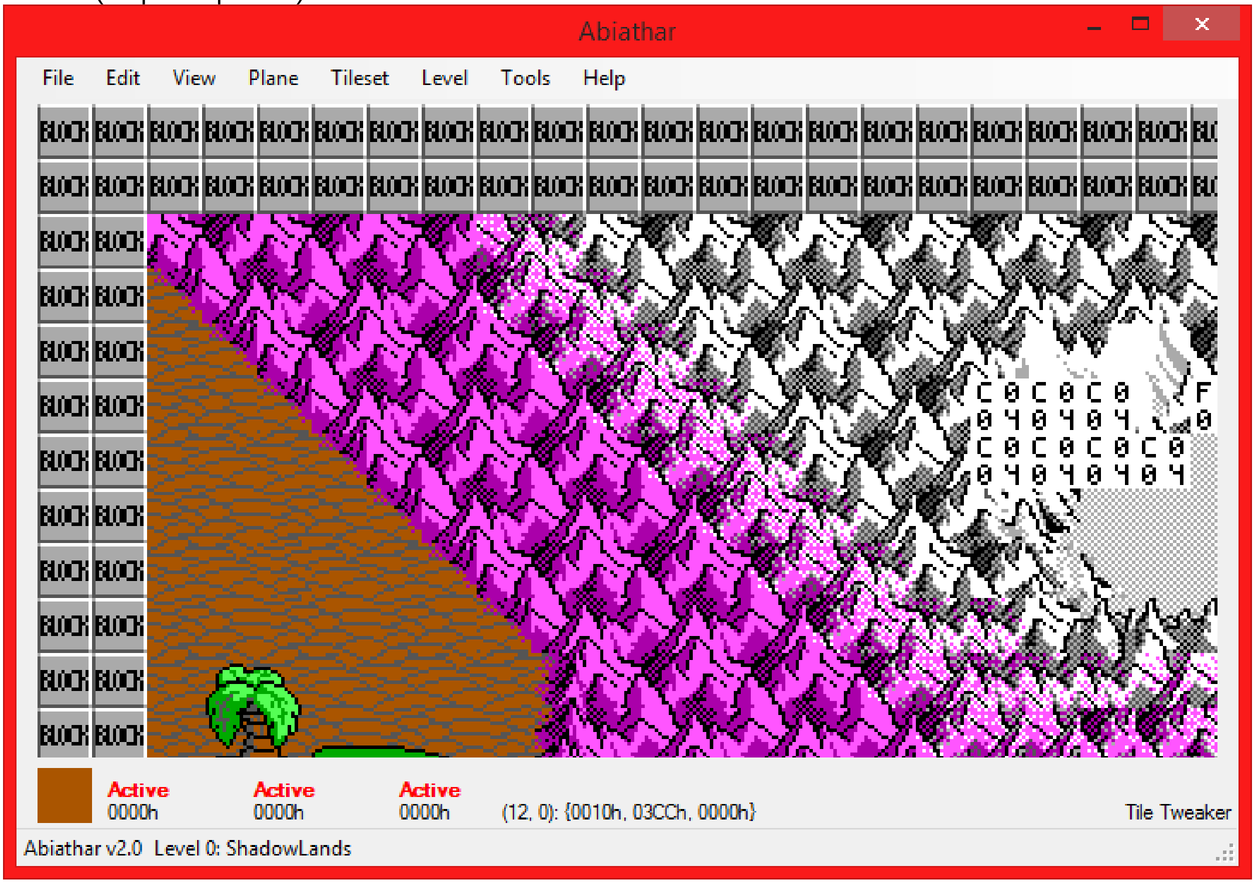
While viewing a level, the 1-3 keys control editibility for the respective plane, while the 4-6 keys control visibility. For example, the 2 key toggles the foreground plane between Active and Locked, while the 4 key toggles it between Active and Hidden.

The labels in the selected tile bay that display the plane states can also change them. Right-click one to make it less active (active will become locked, locked will become hidden) or left-click to make it more active (hidden will become locked, locked will become active).

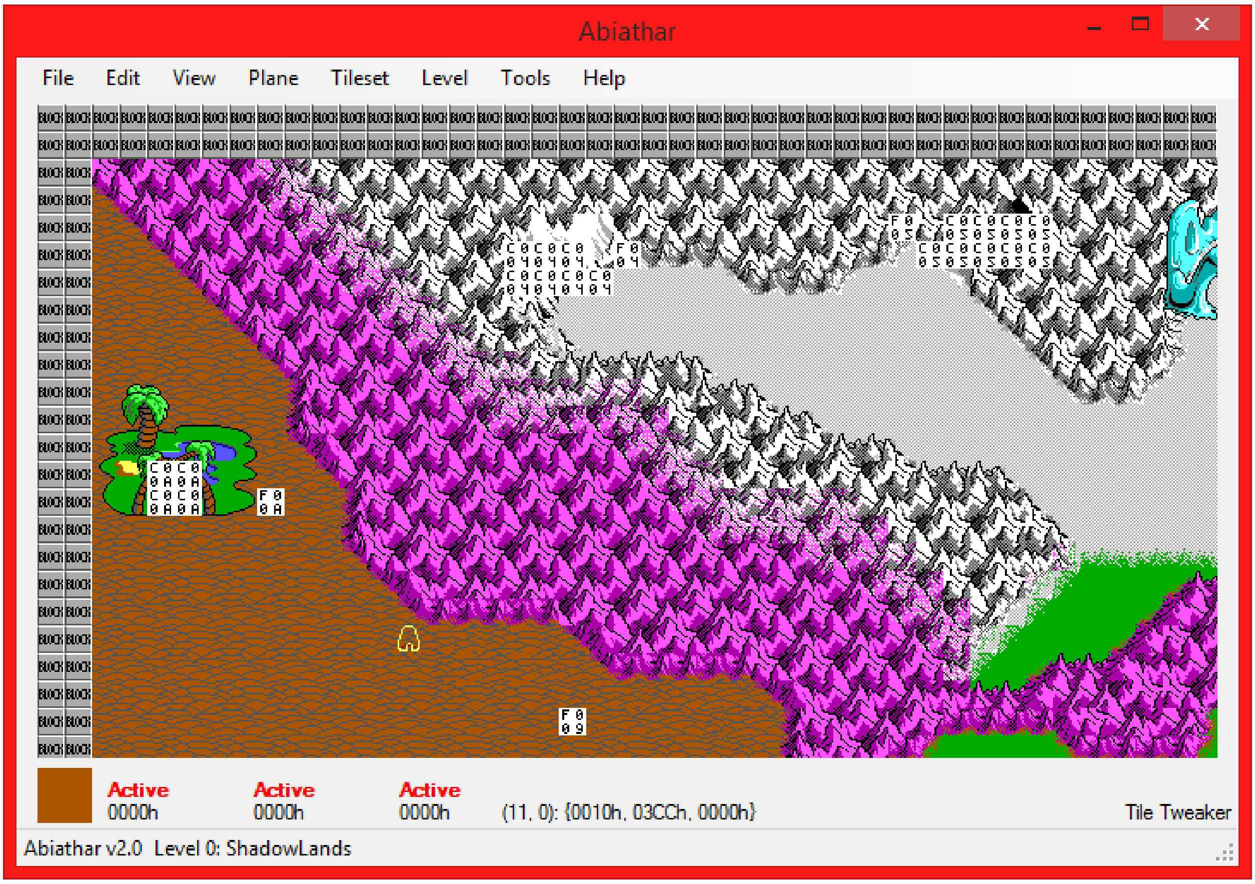
Zoom

Abiathar starts at twice the zoom level of previous editors. The zoom controls can be found under the View menu or by using the plus and minus numpad keys.

Default (32 pixels per tile):



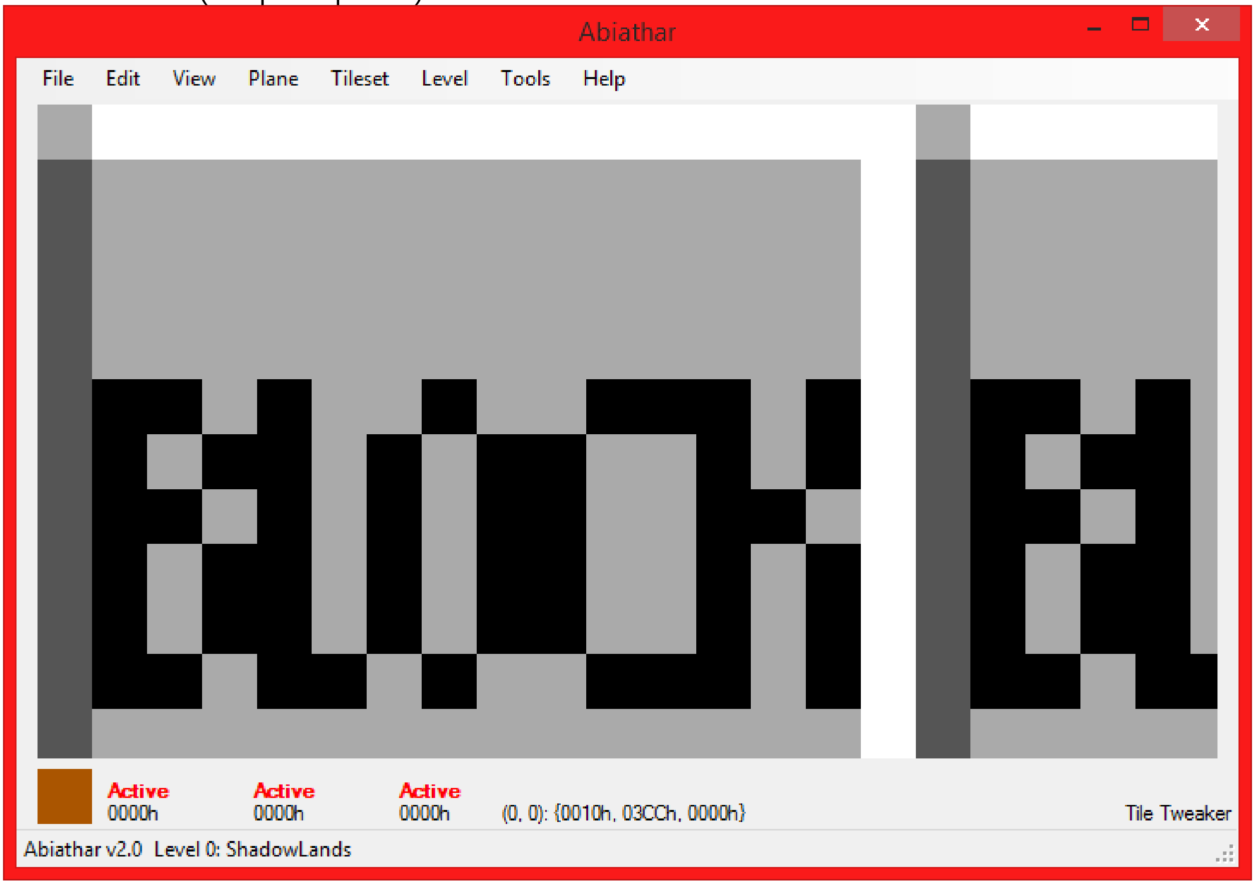
The Omegamatic's default (out 1 notch, 16 pixels per tile):



Active 0000h Active 0000h Active 0000h (11, 0): {0010h, 03CCh, 0000h} Tile Tweaker

Abiathar v2.0 Level 0: ShadowLands

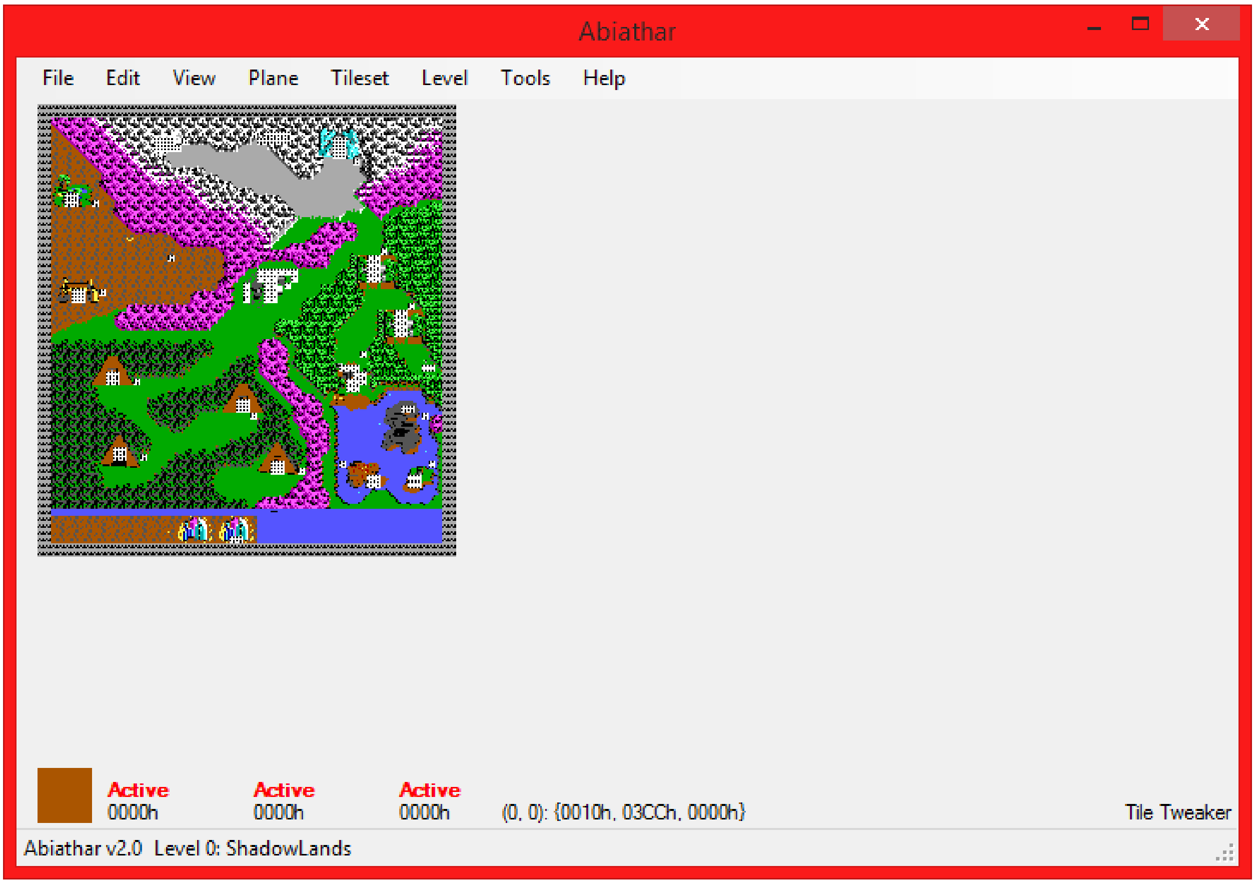
Maximum zoom (512 pixels per tile):



Active 0000h Active 0000h Active 0000h (0, 0): {0010h, 03CCh, 0000h} Tile Tweaker

Abiathar v2.0 Level 0: ShadowLands

Minimum zoom (4 pixels per tile):


Abiathar
File Edit View Plane Tileset Level Tools Help
Active
0000h
Active
0000h
Active
0000h
(0, 0): {0010h, 03CCh, 0000h}
Tile Tweaker
Abiathar v2.0 Level 0: ShadowLands

Zoom order in pixels per tile, 32 is default: `{4, 8, 16, 32, 48, 64, 96, 128, 256, 512}`

Undo & Redo

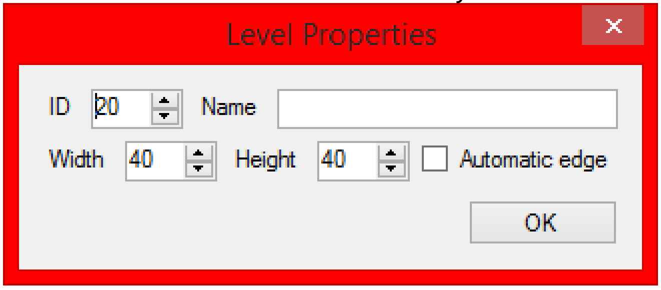
To err is human. Though it might not be possible to stop oneself from making a mistake, Abiathar can help you correct it without having to do all the hard work again. Like in most applications, Undo and Redo can be found under the Edit menu. It may be easier to press Z instead, which works equally well. The undo/redo action stack is preserved until the file is saved or closed.

Important note: Some extensions may not support undo/redo operations. Contact your extensions' publishers for questions as to their undo stack safety.

Level Operations

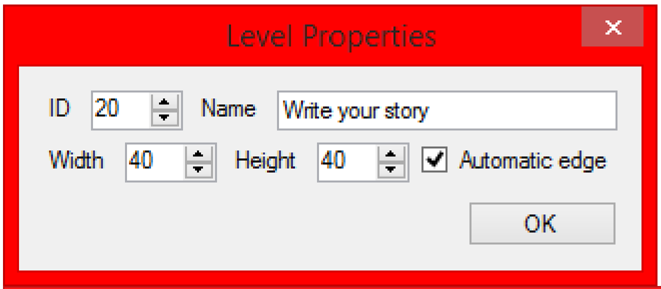
Abiathar contains never-before seen level commands to make management of your mod that much easier. These can be accessed from the Level menu, above the levels list.

Adding a level is made easy by Level>Add: no more looking for the next free level ID. If you change the suggested ID to one that is in-use, Abiathar will warn you before overwriting. The "Automatic edge" check box will cause Abiathar to automatically fill in the boundary.


Level Properties

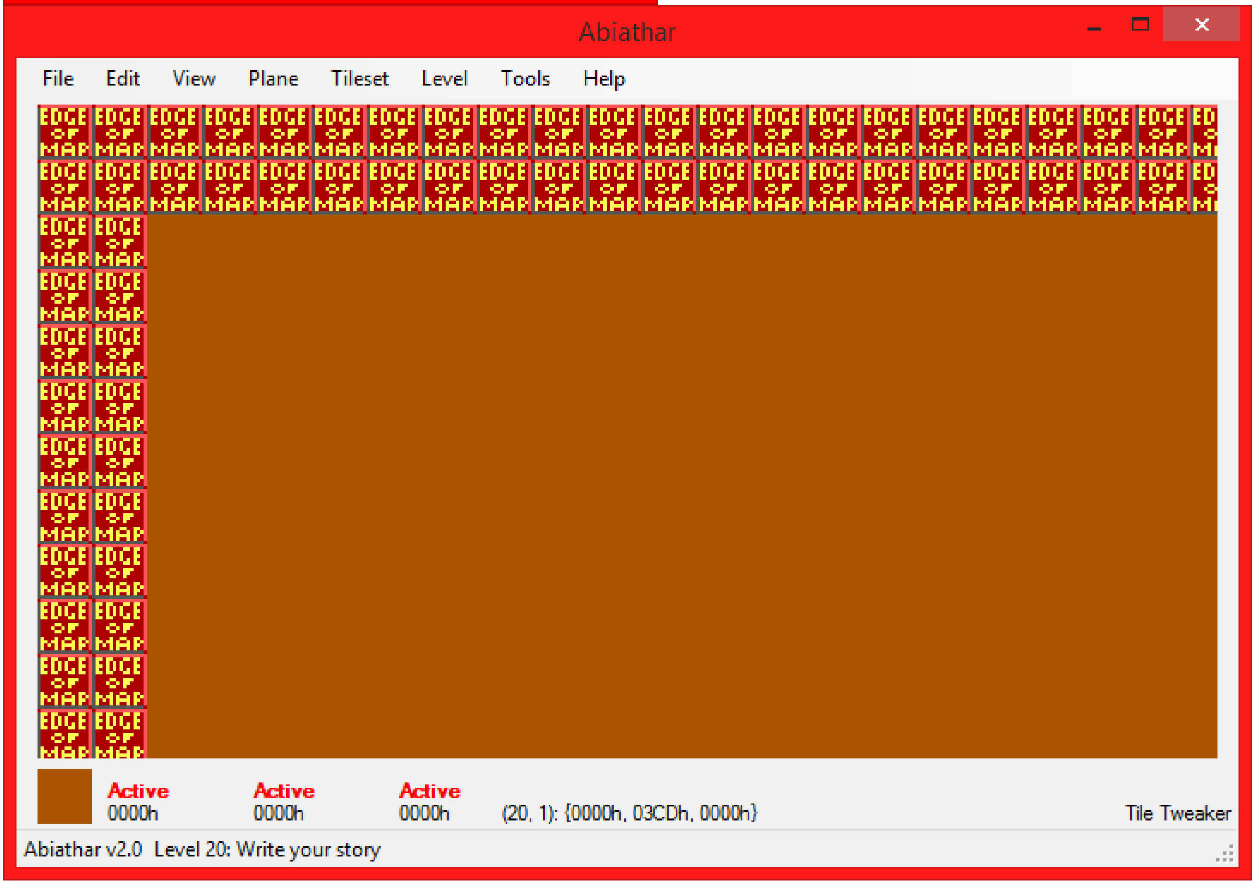
ID 20 Name

Width 40 Height 40 □ Automatic edge



ID 20 Name Write your story

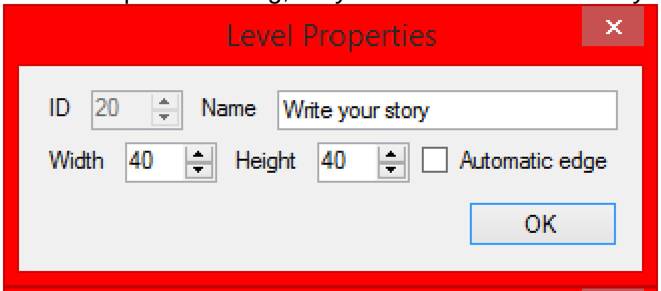
Width 40 Height 40 ☑ Automatic edge



Active 0000h Active 0000h Active 0000h (20, 1): {0000h, 03CDh, 0000h} Tile Tweaker

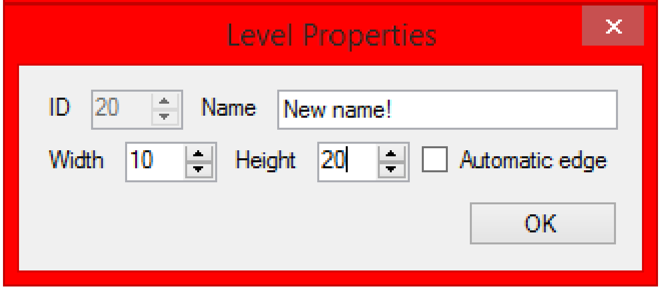
Abiathar v2.0 Level 20: Write your story

Even after a level is created, its dimensions can be changed or an automatic border added using the Level>Properties dialog, very similar to that created by adding a level:



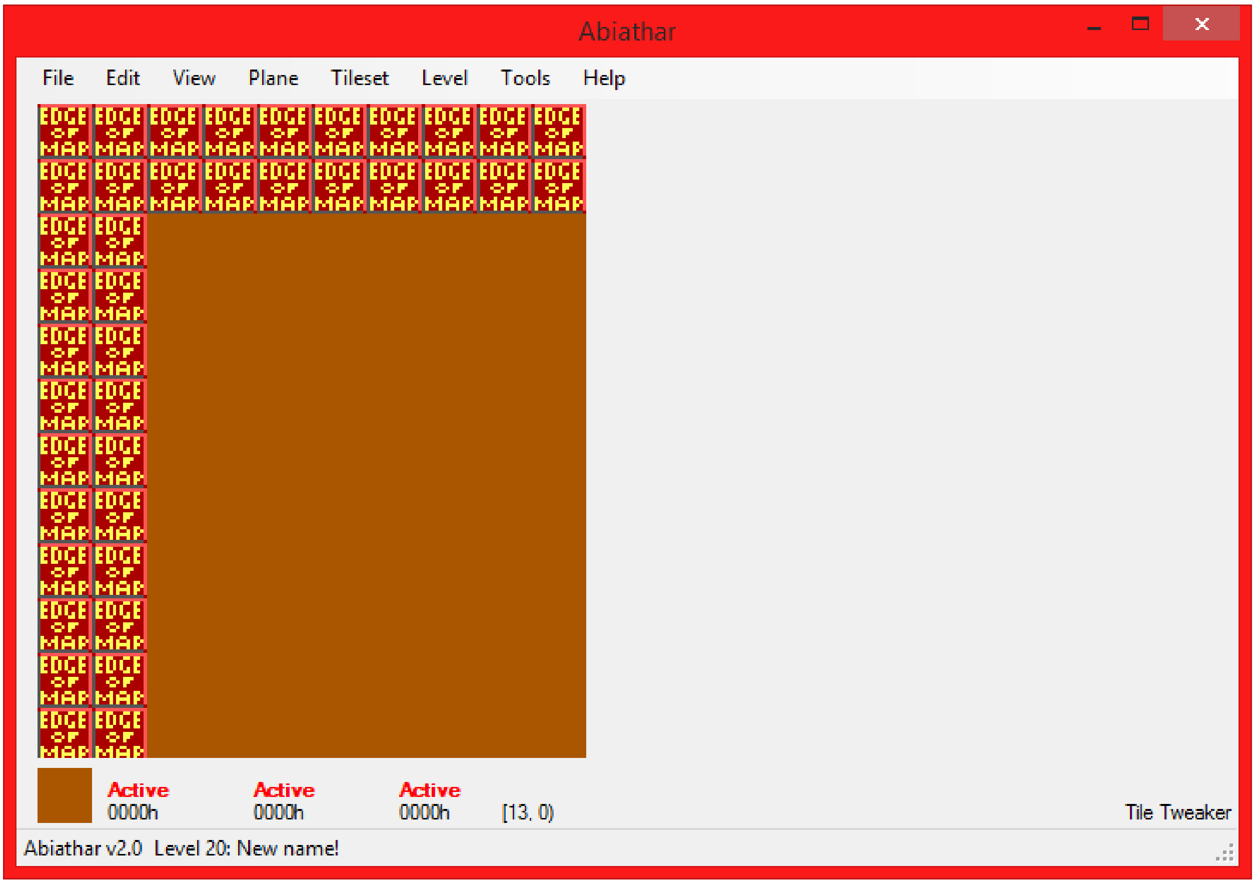
ID 20 Name Write your story

Width 40 Height 40 □ Automatic edge



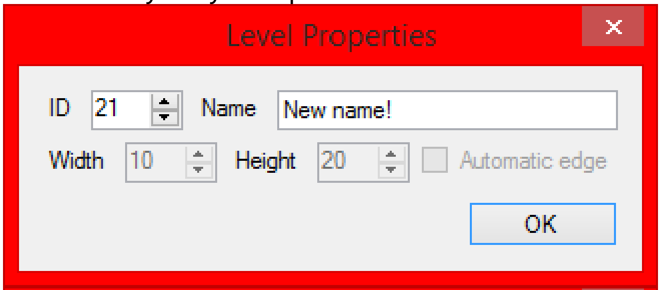
ID 20 Name New name!

Width 10 Height 20 □ Automatic edge

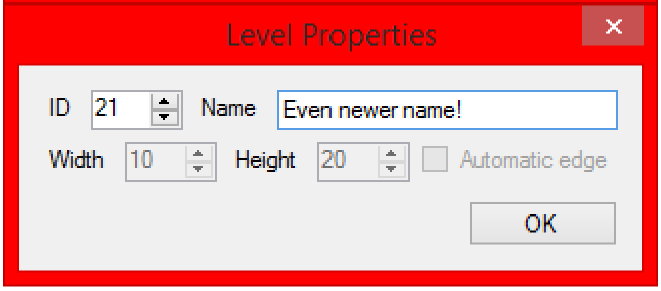


Abiathar v2.0 Level 20: New name!

It is also very easy to duplicate levels into a new slot with Level>Copy:

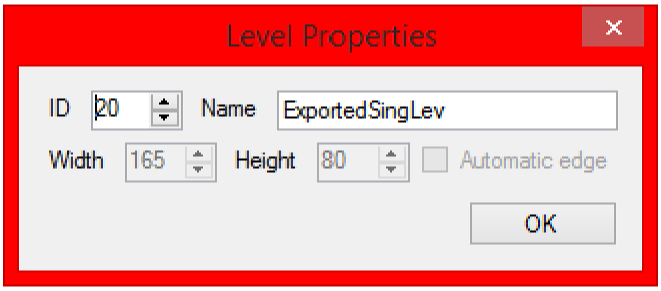


ID: 21
Name: New name!
Width: 10
Height: 20
□ Automatic edge



ID: 21
Name: Even newer name!
Width: 10
Height: 20
□ Automatic edge

Finally, Abiathar can create single-level files for easy assimilation into a group project. The selected level will be exported as an .aslev containing all information about the level, which can then be inserted anywhere in the level set.



ID 20 Name ExportedSingLev

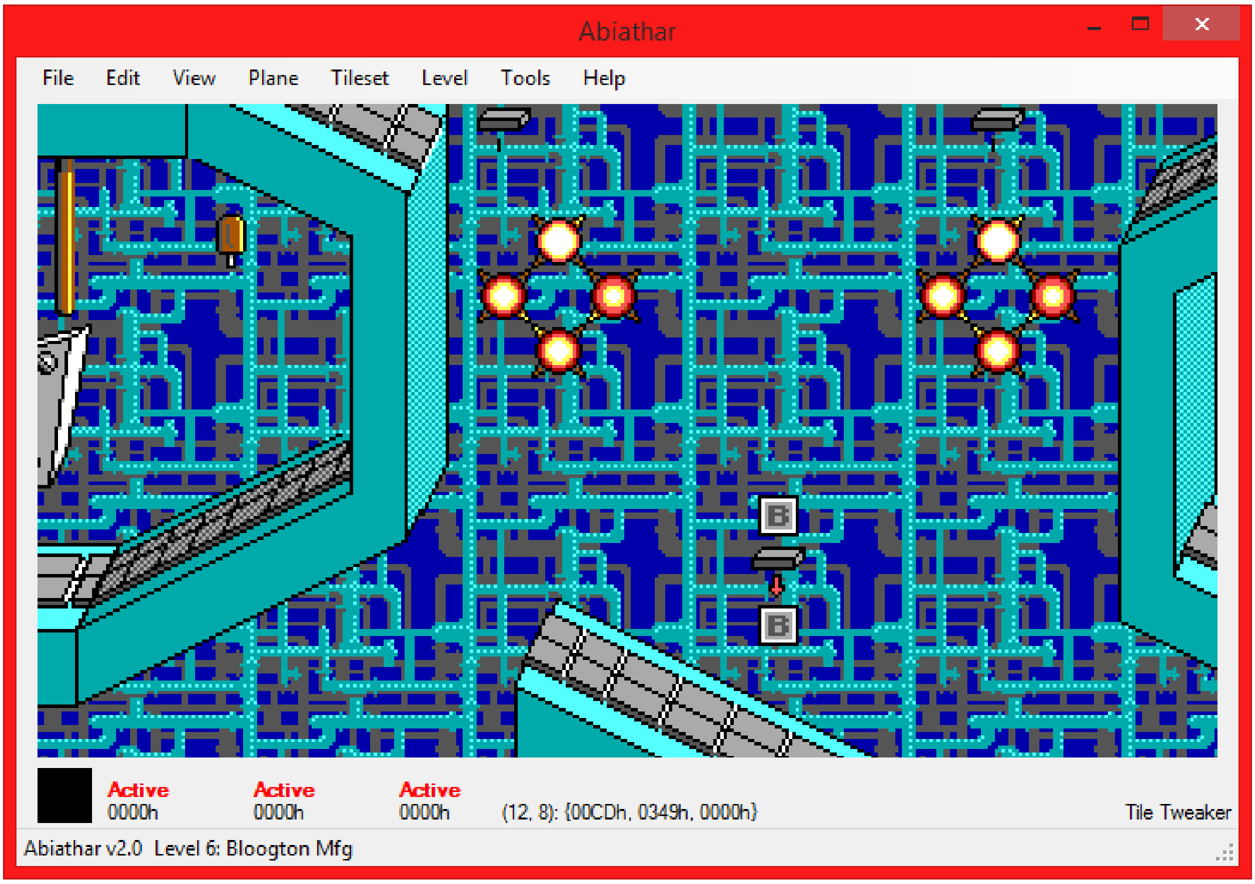
Width 165 Height 80 □ Automatic edge

Some Dependency Collector templates also enable the modification of the internal four-character level signature. If the file allows this, a “Signature” field will appear in the lower-left of the level property window. Notably, NetKeen uses this field as the “Author” entry of a map.

Overlays

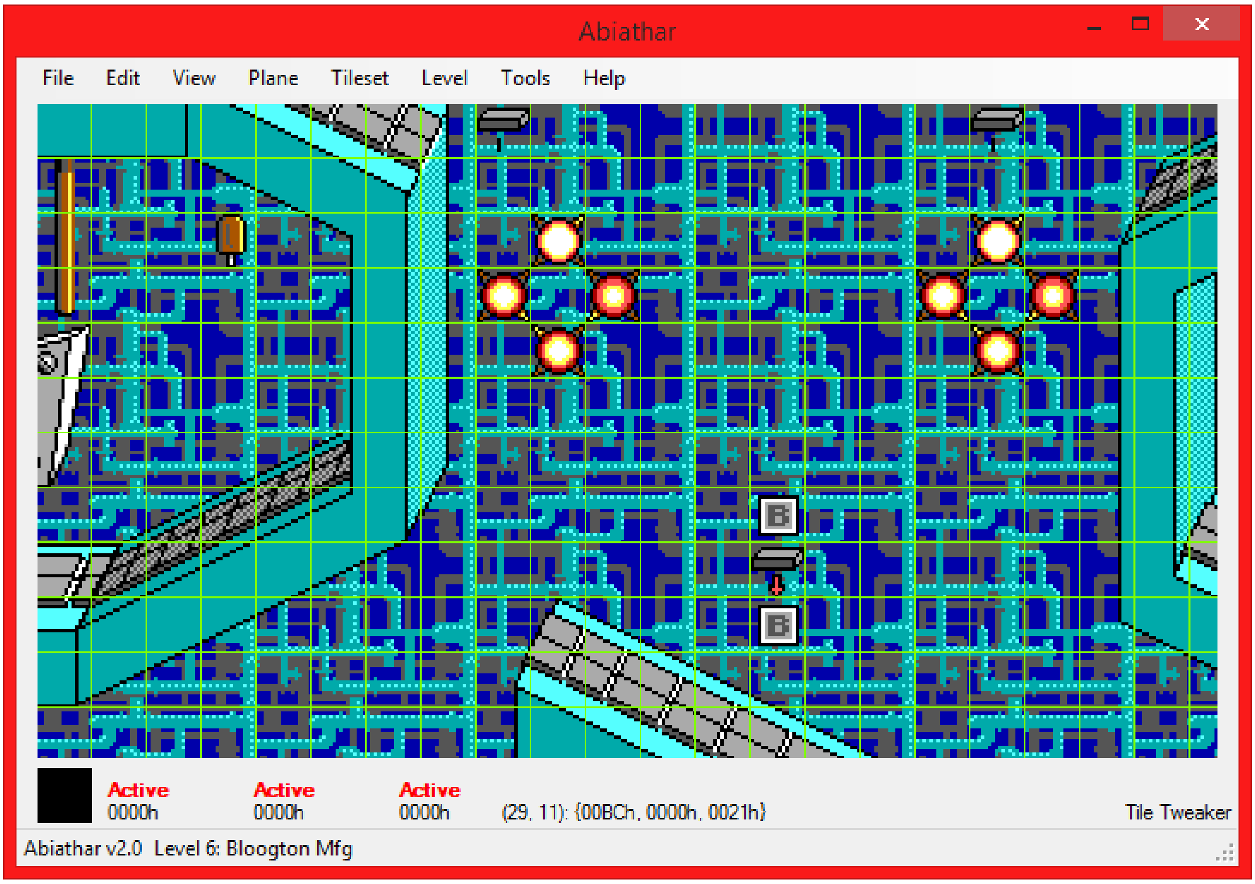
Abiathar features three overlay pseudo-planes to help you find what you need to, all under the View menu. In this article, a screenshot has been taken before and after activating the overlay.

The Grid displays, as one might expect, a grid separating the tiles. It is automatically adjusted for changes in zoom level.

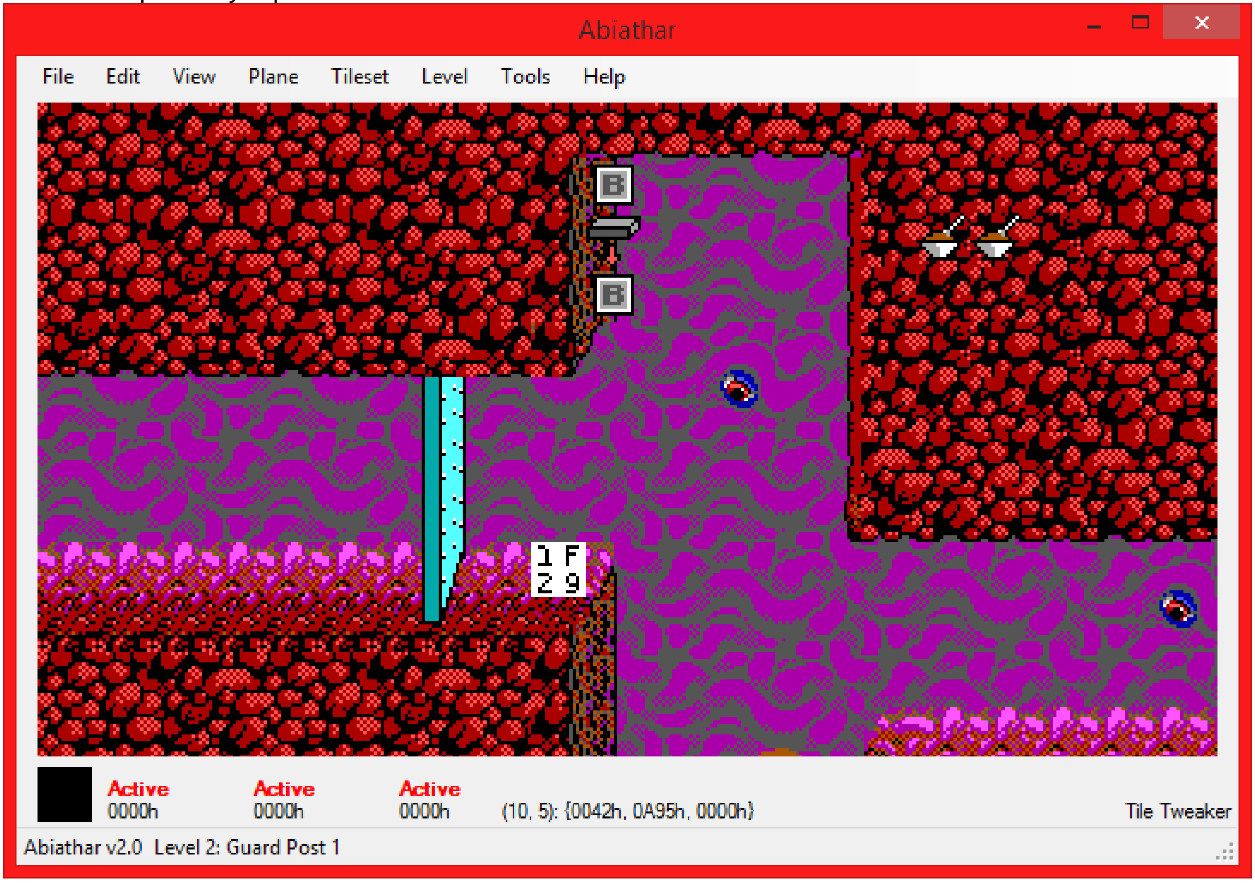


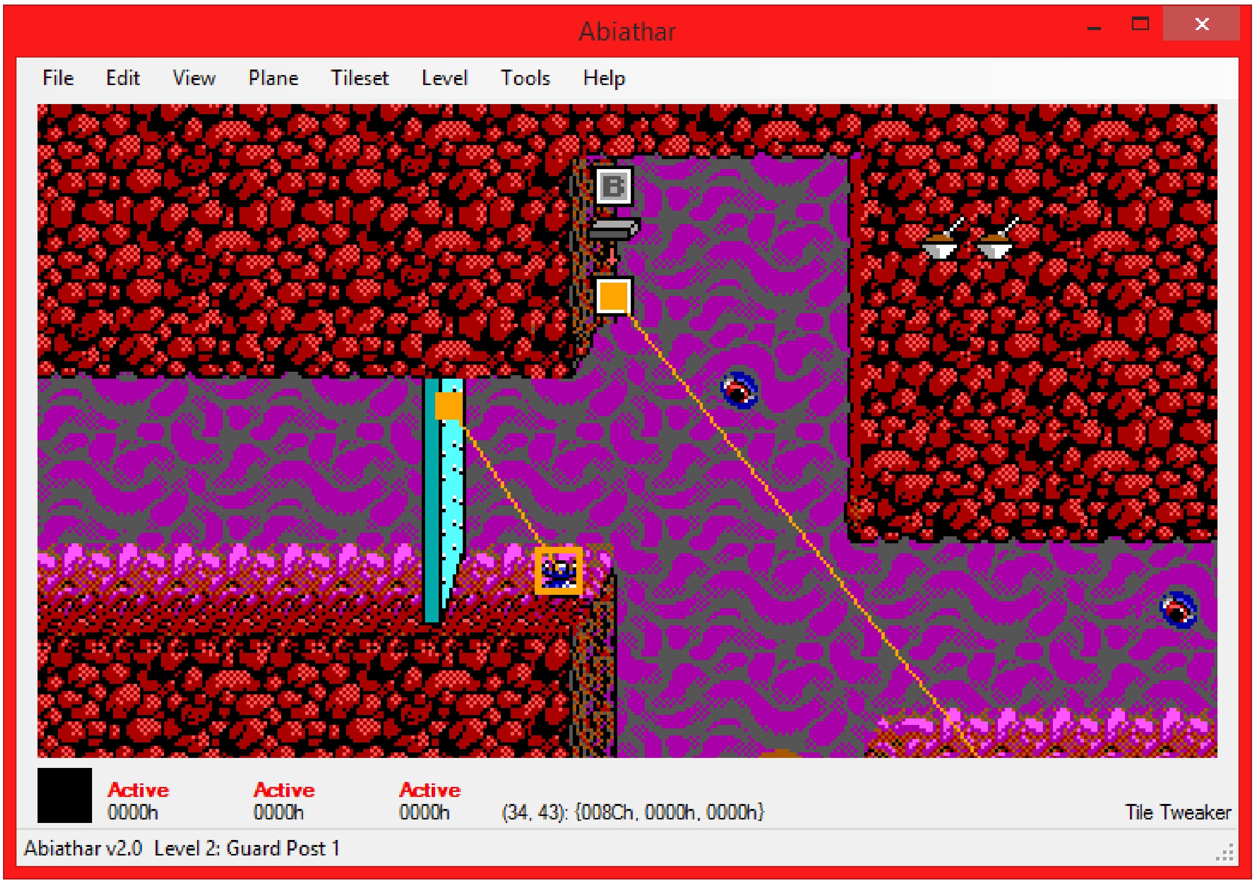
Active 0000h Active 0000h Active 0000h (12, 8): {00CDh, 0349h, 0000h}

Abiathar v2.0 Level 6: Bloogton Mfg

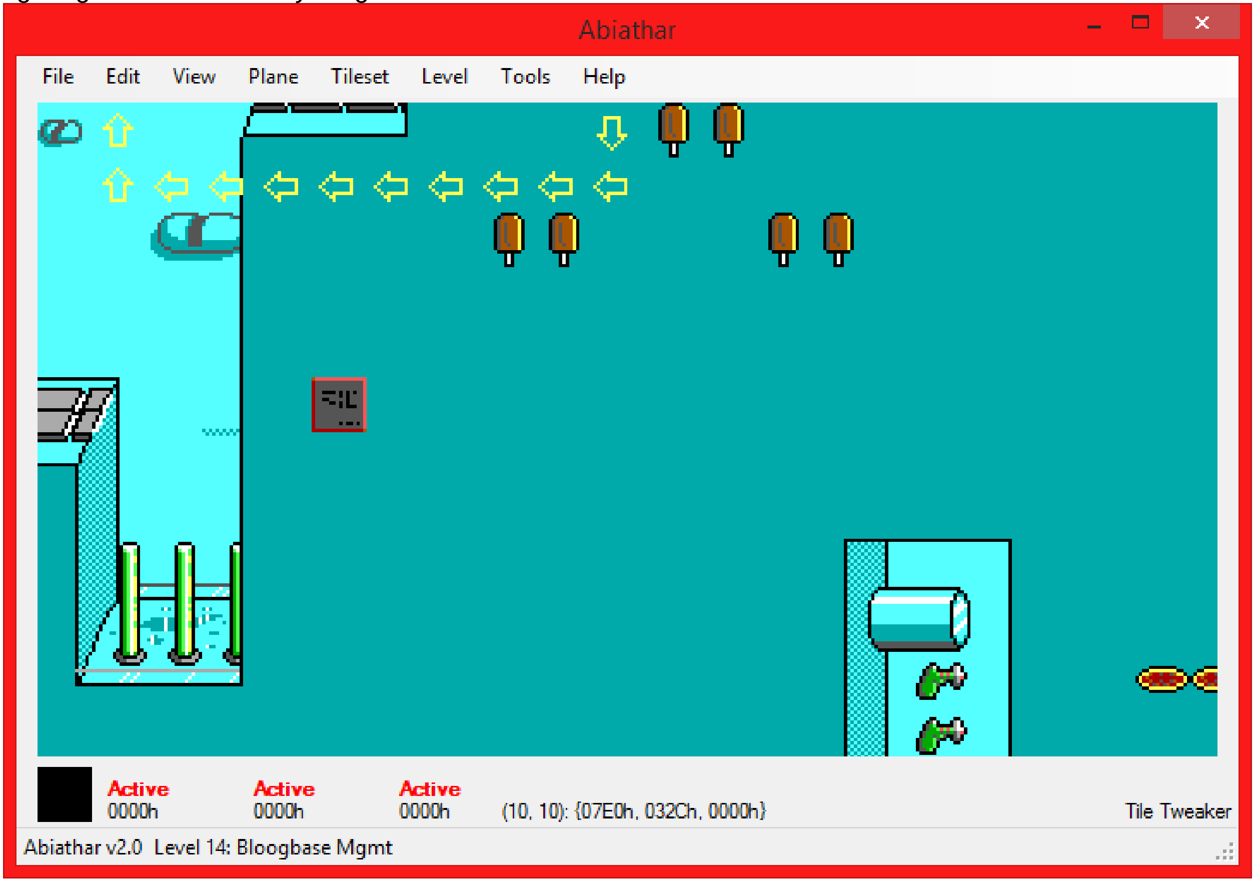


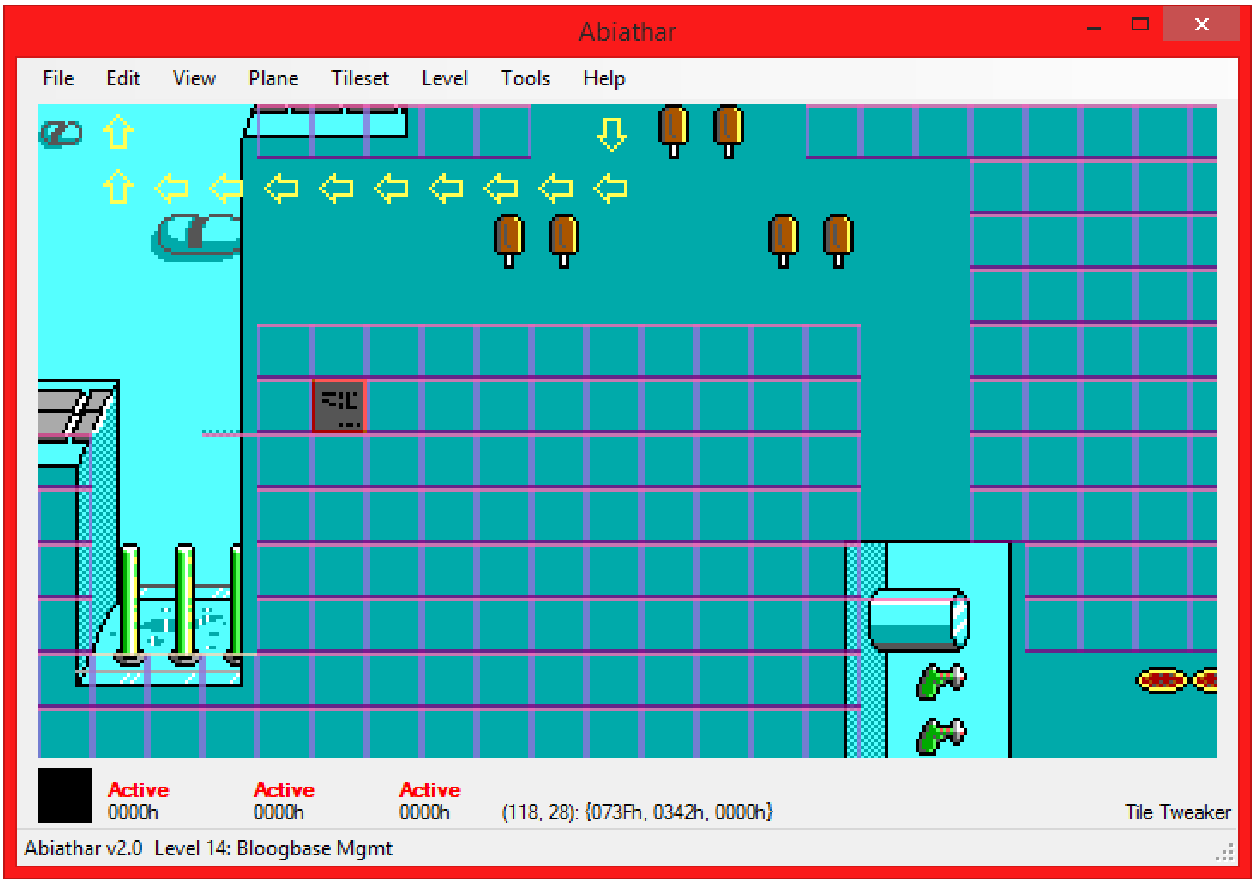
The Links overlay processes unknown infoplane values and renders graphical links instead of infoplane values that probably represent links.




Abiathar
File Edit View Plane Tileset Level Tools Help
Active
0000h
Active
0000h
Active
0000h
(34, 43): {008Ch, 0000h, 0000h}
Tile Tweaker
Abiathar v2.0 Level 2: Guard Post 1

Never before seen in a Keen level editor, Abiathar can understand and display foreground tile properties. Even in the selected tile bay and tile palette, you can see the blocking on a tile and not have to waste time figuring out later than all your ground is made out of secret area tiles.


Abiathar
File Edit View Plane Tileset Level Tools Help
Active
0000h
Active
0000h
Active
0000h
(10, 10): {07E0h, 032Ch, 0000h}
Tile Tweaker
Abiathar v2.0 Level 14: Bloogbase Mgmt



Active 0000h
Active 0000h
Active 0000h
(118, 28): {073Fh, 0342h, 0000h}

Abiathar v2.0 Level 14: Bloogbase Mgmt

Lone Editing

When doing fine detail work on a level, it is often necessary to work with only one plane at once. Lone Editing, an option found under Tileset, helps you do exactly this. When it is enabled, opening a tile palette causes that plane to become active and all others that were previously active to become locked. This mode is probably most helpful in Simultaneous Tileset mode.

Playing the Levels

Once some work has been done on a level set, it is advisable to test them in-game. This, and the final publication of the mod, require the use of CKPatch to patch the maphead (and, if they exist, mapdict and tileinfo) back into the game to read the modified gamemaps file. Abiathar makes this exceedingly simple.

If you have other custom patches that are not automatically generated by Abiathar, transfer them to the Patches store found under Edit. Remove from the custom patches all the things that Abiathar will handle for you (which are listed above the text field) and click OK. These will be integrated into the generated patch file.

Choose Generate Patches under File. If the episode template you selected supports the patch generator (as all the templates for the original Keen games do), Abiathar will inform you of the files to be created or overwritten. Once you click OK, the appropriate patch, batch, and patcher program files will be created in the dependency file’s directory. Make sure you run the “play.bat” launcher file instead of the original Keen executable.

iD Software used a self-extracting compression format called LZEXE on the program to make it smaller, thereby fitting it on a floppy disk. Before CKPatch can work, however, the Keen executable must be decompressed. Abiathar contains UNLZEXE, a program to reverse the compression. Choose “Un-LZ EXE” under File and open the Keen executable. If it was LZ-compressed, Abiathar will decompress it; if not, you will be informed.

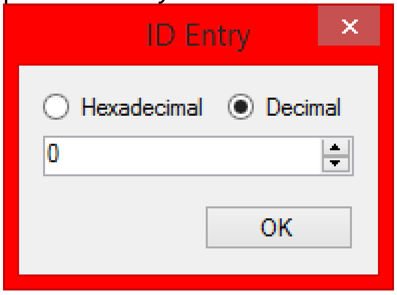
With those steps done, you are ready to play your levels in Keen! Congratulations, modder!

Advanced Use

Over the course of Abiathar's development, there have been so many tiny time-saving tricks added to the code base that they can't all be remembered or documented. This section contains the most interesting and important shortcuts to help advanced users get around faster. It also documents some of the less frequently used features.

Tile ID Entry

Right-clicking any of the selected tile bay images will produce a dialog that allows you to type the tile ID to put in that bay.

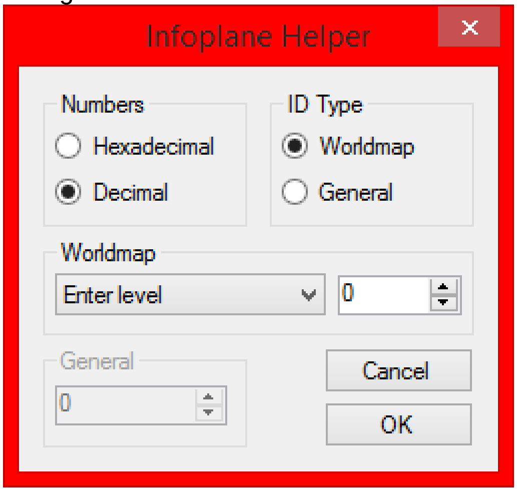


○ Hexadecimal ● Decimal

0

Entering world map infoplane values (and possibly all values) is easier to do in Hexadecimal mode, which displays the number like the selected tile captions do.

Right-clicking the infoplane's image will produce a more complex dialog that helps you create special world map values without having to memorize anything. It can also be set to act like the other planes' ID entry dialog.



Numbers: ○ Hexadecimal ● Decimal

ID Type: ● Worldmap ○ General

Worldmap: Enter level 0

General: 0

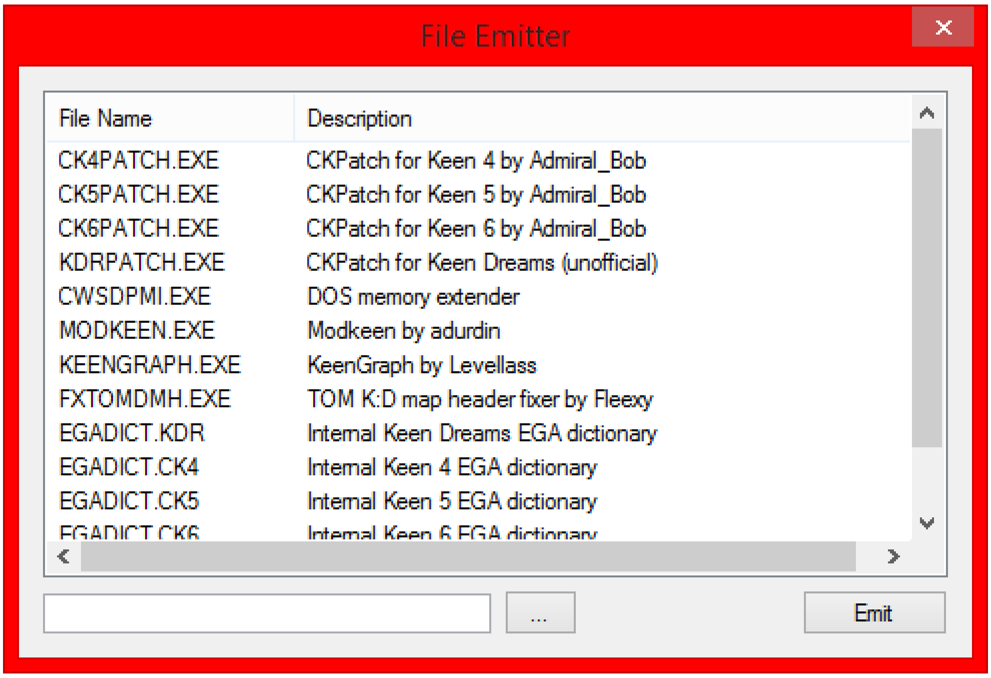
The following high (first) byte values have special effects on the world map. The low (second) byte specifies the level ID to which it applies.

- C0 allows the player to enter the level when on that tile
- D0 deletes the foreground tile after the level is completed
- F0 will receive the K flag when the level is completed

For example, an infoplane value of F00A marks the flag holder for level 0A (10).

File Emitter

One of Abiathar's important goals is to remove the necessity for keeping original game files and smaller utilities around. To this end, a collection of these files is contained in the Abiathar executable. Some are used by the Dependency Collector to start from game defaults, but the rest can be taken out via the File Emitter. This utility can always be found under the File menu:



File Name	Description
CK4PATCH.EXE	CKPatch for Keen 4 by Admiral_Bob
CK5PATCH.EXE	CKPatch for Keen 5 by Admiral_Bob
CK6PATCH.EXE	CKPatch for Keen 6 by Admiral_Bob
KDRPATCH.EXE	CKPatch for Keen Dreams (unofficial)
CWSDPMI.EXE	DOS memory extender
MODKEEN.EXE	Modkeen by adurdin
KEENGRAPH.EXE	KeenGraph by Levellass
FXTOMDMH.EXE	TOM K:D map header fixer by Fleexy
EGADICT.KDR	Internal Keen Dreams EGA dictionary
EGADICT.CK4	Internal Keen 4 EGA dictionary
EGADICT.CK5	Internal Keen 5 EGA dictionary
EGADICT.CK6	Internal Keen 6 EGA dictionary

To dump a file, select it, specify the output directory in the box below, and click Emit. Multiple files can be selected at once.

Using Extensions

.NET programmers are able to create extensions for Abiathar via its extension API, Interop.dll. These extensions are published as .aex files, but are really just normal class libraries with the extension changed from .dll.

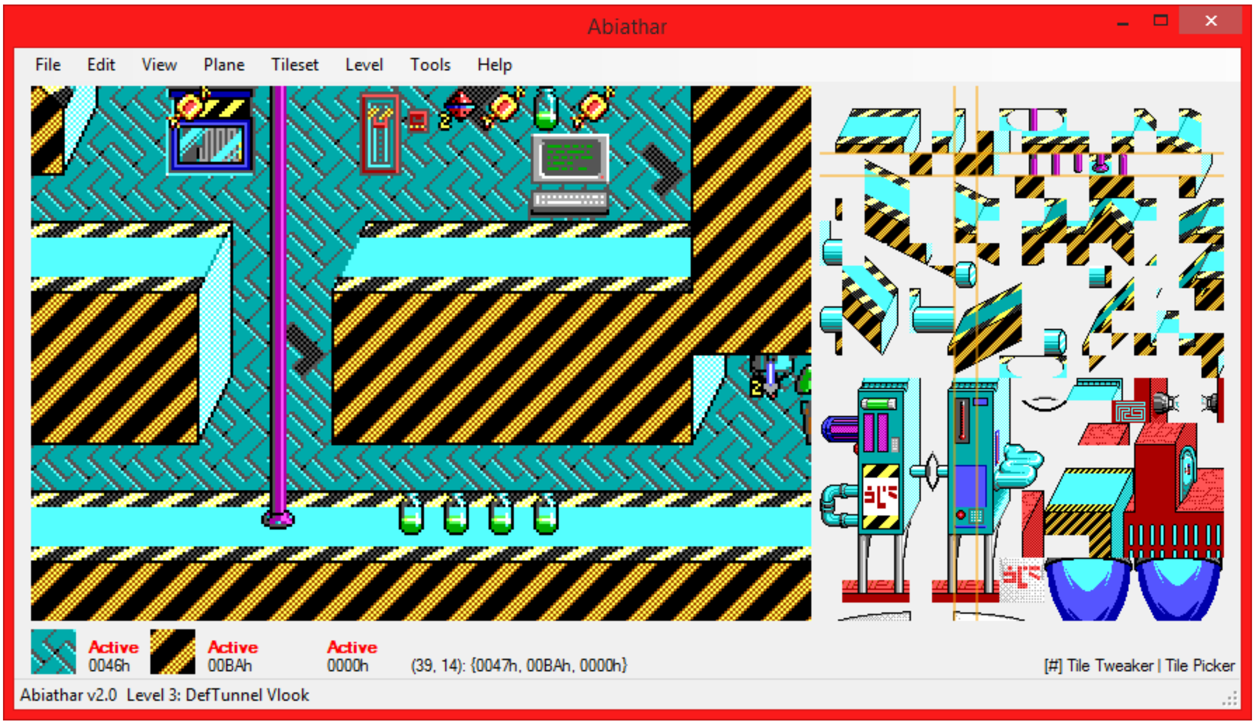
To make use of these extensions, you must notify Abiathar that it should look for extensions. Create a folder named `AbiatharExt` under the directory containing the Abiathar executable. Copy or emit Interop.dll into this folder. Then, copy the .aex files you wish to use into the folder (along with their supporting .dll's, if they have any).

Launch Abiathar. If the extensions folder has been set up correctly, there will be an extra "Extensions" menu on the far right of the main menu bar, which will contain information about and settings from the loaded extensions.

Note: If Abiathar crashes when loading extensions, delete the configuration files and try again. If that fails to fix the problem, contact your extensions' publishers.

Simultaneous Tileset

Users of z-one's Keen Next editor may have grown attached to its having the tileset visible at the same time as the level. To help those users migrate to Abiathar, it features a Simultaneous Tileset mode, which can be found under the Tileset menu.



Active 0046h
Active 00BAh
Active 0000h
(39, 14): {0047h, 00BAh, 0000h}
[#] Tile Tweaker | Tile Picker
Abiathar v2.0 Level 3: DefTunnel Vlook

In this mode, the use of multiple tools can be in progress at once. Since many tools are made more efficient by the use of keyboard buttons or the mouse wheel, the "focus" can be switched between the tileset and the level by pressing Tab or simply moving the mouse into the other pane. The [#] symbol on the selected tool label shows which pane will get to process keyboard buttons and scrolling first.

The simultaneous tileset does not respect the zoom level, but zooming still works on the level.

FxTomDMh.exe

A bug in The Omegamatic results in it creating an invalid map header for Keen Dreams levels. Though Galaxy map headers are only 402 bytes long, Dreams headers contain an extra 100 bytes: one for each level, giving the length of the compressed header. Using these corrupted files in Keen may work for small modifications, but if levels are added, Keen will not be able to use them. (Abiathar is able to deal with a truncated maphead and will write a complete one when saving.)

To remedy this, Abiathar's File Emitter contains FxTomDMh.exe, a small utility that performs a quick fix on `MAPHEAD.CK7`. Once emitted, drag the bad map header file onto FxTomDMh. 100 bytes of "42" will be appended, allowing Keen to use the attached game maps. Using this tool is only necessary if you wish to repair existing TOM-corrupted levels without using Abiathar.

ManTliMh.exe

Since the map headers and tile properties are stored right next to each other in the game executables, some mod or tool authors prefer to place them in one file. Doing so makes it very difficult for other editors to find the information. (Abiathar can be set in the New Project Wizard to handle maphead and tileinfo together.)

To help you manage your mod's resources more easily, Abiathar comes with a small utility called ManTliMh, so called because it manages tileinfo and maphead files. It can be emitted from the File Emitter. To separate tileinfo and maphead, simply drag the combined file onto ManTliMh and enter the names to which the individual files will be outputted. It can also recombine them into one if you drag two files onto its program. (In the Windows command line, the names of files can be passed as command-line arguments or typed directly into the program.)

High Speed Mode

On slower computers, some of Abiathar's special features may cause unresponsiveness. To remedy this, High Speed Mode disables lag-causing features, notably high-quality alpha blending and **the undo/redo**

stacks. Even if tools from extensions do not respect HSM and attempt to push to the undo stack anyway, it will not have any effect.

Abiathar does not have full control over the behavior of extensions. If you have questions about their compliance with Abiathar standards, contact their publishers.

Configuration File

After Abiathar runs for the first time, it will produce an `editor.aconf` file in its residing directory. That file stores the persistent configuration of Abiathar in an easy-to-modify FrofConf (Fleexian Reflective Object/Field Configuration) structure. It can be modified with any text editor; Notepad++ is recommended.

- `LastSavedVersion` should not be modified; it helps Abiathar adjust the structure of the configuration file after updating.
- `Colors` stores various GUI colors as signed 32-bit integers in standard ARGB order. They are most easily modified using the Windows Calculator in Programmer view.
    - `Grid` controls the color of the grid overlay.
    - `Links` controls the color of the links overlay, both the lines and rectangles.
    - `TileCrosshair` controls the color of the lines that make up the selected tile rectangle
    - `PropsSide` controls the color of the lines that represent a solid left or right side of the tile property overlay.
    - `PropsTop` controls the color of the lines that represent a solid top side or slope of the tile property overlay.
    - `PropsBottom` controls the color of the lines that represent a solid bottom side or slope of the tile property overlay.
    - `PropsExtra` controls the color of the letter overlays that represent special properties of the tiles.
    - `Selection` controls the color of the translucent selection rectangle.
- `ShowAdvancedMenuOptions` enables the visibility of menu items belonging to rarely-used advanced settings. Set a child value to `True` to make its item visible.
    - `HighlightCurrentTile` permits the current-tile crosshairs to be disabled.
    - `DowngradeLocked` allows the use of the Downgrade Locked plane interaction change.
    - `CacheAll` allows pre-rendering all tiles and level planes.
    - `LegacyDepsCollector` allows the use of the v1.x Dependency Collector in the New Project Wizard.
- `UseHexadecimalDisplay` changes the numeric format of various number types in the status box.
    - `SelectedTile` affects the tile IDs shown in the selected tile bay next to the images.
    - `MouseOverCoordinate` affects the Cartesian coordinates of the mouse pointer in the level.
    - `MouseOverTile` affects the tiles at the mouse pointer's location in the level.
- `OpaquePastePreview` makes the paste preview overlay opaque, as seen in The Omegamatic and Keen Next.
- `KeenNextNostalgia` enables additional highlights as seen in Keen Next.
    - `SingleTileBrush` enables the preview overlay for the Tile Tweaker and the Freeform Tile Placer.
    - `CopyToolMousePosition` enables the selection overlay for the Copier and Palette Copier even when not copying.
- `DefaultView` causes various toggle menus in the GUI to be activated upon opening a dependency file.
    - `AutoToolCancel` enables Auto Tool Cancel under Edit.
    - `HighSpeedMode` enables High Speed Mode under Edit.
    - `Zoom` sets the default zoom level in pixels per tile.
    - `ShowGrid` enables Grid under View.
    - `ShowLinks` enables Links under View.
    - `ShowTileProps` enables Tile Properties under View.
    - `LoneEditing` enables Lone Editing under Plane.
    - `AssimilateInfoplane` enables Assimilate Infoplane under Tileset.
    - `SimultaneousTileset` enables Simultaneous under Tileset.
    - `ContextualHelp` enables Contextual under Help.
- `BitmapExporter` controls the behavior of the Export Image option under View.

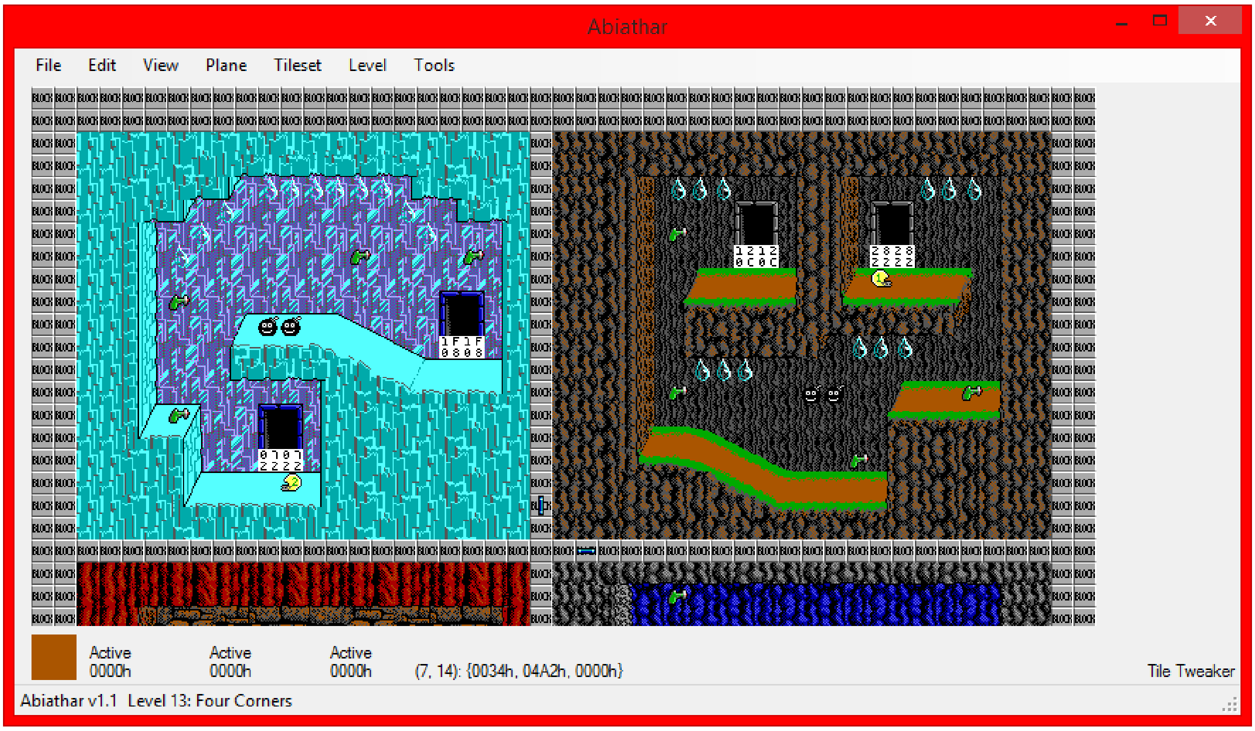
  - `ClipLevelBorders` removes the 2-tile-wide level border from the resulting bitmap.
  - `PreserveZoomLevel` respects the selected zoom level; otherwise, the bitmap is rendered at 16 pixels per tile.
  - `SuppressConfirm` removes the "Image saved" message box.
  - `AutoFileName`, if specifies, suppresses the save-file-dialog and instead generates the filename automatically, replacing text macros with information about the level and dependency file. "`%l`" becomes the level name, "%n" becomes the level number, "%p" becomes the directory containing Abiathar, "%d" becomes the directory containing the dependency file, and "`%f`" becomes the name of the dependency file.
- `ExtendedTileinfo` enables the tile property overlay to display more special properties.
- `CheckNonzeroAnimOffset` causes the inspection mode of the Tile Property Modifier to produce alerts about tiles with an animation delay but no animation offset.
- `ChangePaneFocusOnMouseover` allows the mouse pointer to change which pane is focused in Simultaneous Tileset mode.
- `FreeScroll` allows the viewport to be moved all around without level bound checks. If this is `False`, scrolling will automatically turn at corners.
- `ArrowScrollMul` is how many tiles will be panned for each unmodified press of an arrow key.
- `MouseScrollDetScale` is how many mouse wheel detents are required to scroll one tile.
- `NostalgicPlaneControl` enables TED5-style plane controls, as opposed to Abiathar v1.x function-key plane controls.
- `QueryImfType` causes the IMF importer in Music Mappings to ask about the type of IMF files. If this is False, they are assumed to be Type-1 (full chunks).
- `VeriMapsCertPath` is the relative path where Abiathar will look for a VeriMaps signing certificate.
- `ShowDidYouKnow` enables or disables the display of the "did you know?" tips on start-up and from tools. This can also be disabled by the check mark on those tips.
- `DefaultLevelToolID` sets the zero-based ID of the tool to be loaded by default for modifying levels. If it is `1` (for the Freeform Tile Placer), the Paster will exhibit behavior reminiscent of Keen Next.
- `DefaultTilesetToolID` sets the zero-based ID of the tool to be loaded by default for viewing the tile palette.

Modding NetKeen

Lemm's NetKeen formats are, for the most part, identical to those used by Keen Galaxy. The most important difference is that NetKeen places its map headers and tile properties in the same "maphead" file. Therefore, you need to select "Manage tileinfo as part of the map header" in the Tileinfo Source step of the New Project Wizard.

Simply choose "NetKeen" from the template dropdown in the New Project Wizard and use it as usual.

Sign your levels in the Properties window! NetKeen uses the internal four-character signature as the Author field in the map setup.



Cache Forcing

To improve speed, Abiathar performs render operations only when they are necessary. However, on slower machines, this may cause noticeable lag after major view state changes. On those, it may be a good idea to force the cache of all graphical resources, which will be used for the rest of the editing session. Once a cache is forced, the forcing command will be disabled for the rest of the session.

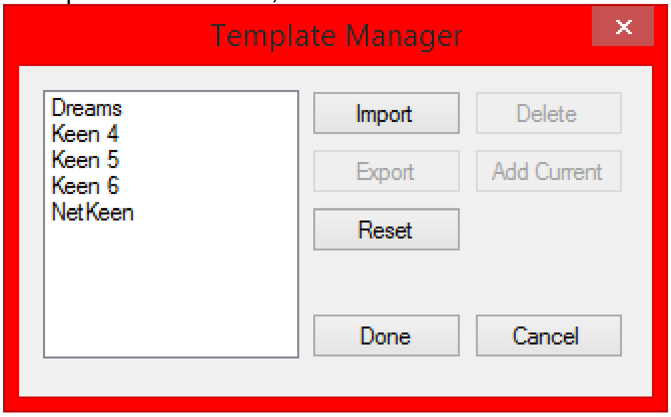
"Tileset>Cache All" processes every tile, transforming them from FleexCore2 format to GDI+ bitmaps. Tile resources are called upon whenever tiles are rendered and heavily so when rendering the tile palette.

"Level>Render All" compiles all aspects of every level, calling on tile resources to do so. Level images are called upon, obviously, to render levels.

Template Manager

The New Project Wizard’s template dropdown can be modified by advanced users to load special EGA structures. It is represented in the FrofConf structure stored in “defaults.aconf”, which is a list of the structures found in the “adeps” files. Basically, the selected template is duplicated and used as a starting point for the new dependency file. Dependency files can be modified and reconfigured without affecting other dependency files created with the same template.

Though it does not help much with creating custom templates, the Template Manager, labeled as “Templates” under File, can assist in the redistribution of custom templates.


Template Manager

Dreams
Keen 4
Keen 5
Keen 6
NetKeen

The list on the left contains the same entries as the template dropdown in the New Project Wizard. Select one to allow deletion or export.

The Import button takes an “atemp” (Abiathar template) file and adds it to the list.

The Delete button removes the selected template.

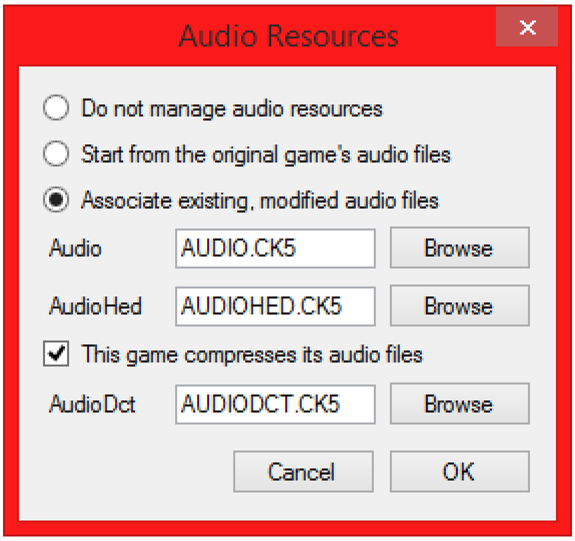
The Export button creates a distributable “atemp” file containing the selected template.

The Add Current button establishes the currently open dependency file as a new template. This is probably only useful if you have been sent a dependency file with special configuration and want to make its structure duplicable.

The Reset button will reestablish the default “defaults.aconf” as the current version.

Music & Sound

To edit sounds, you must use a program designed to work with Keen Galaxy sounds, like WDC. To bring them into the game’s audio resources, you must first link audio resources to your dependency file with Edit>Audio>Resources.



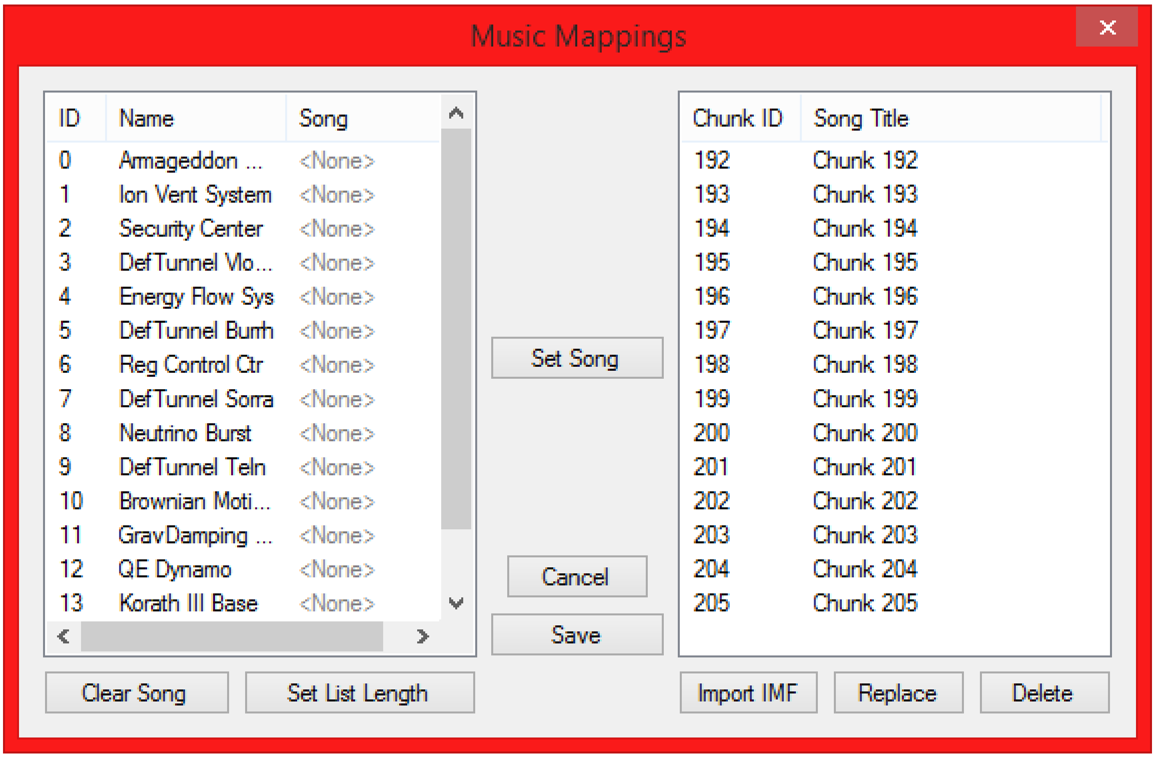
Audio Resources

○ Do not manage audio resources
○ Start from the original game's audio files
● Associate existing, modified audio files

Audio: AUDIO.CK5
AudioHed: AUDIOHED.CK5
☑ This game compresses its audio files
AudioDct: AUDIODCT.CK5

Then, export all sounds with Edit>Audio>Export. It is recommended that you create a directory to store the ASNDS file because one file will be created for each sound in the directory you place the sound binder file. Replace the appropriate files with your custom sounds, then re-import them into the audio resources: Edit>Audio>Import.

There is a simpler way to manage music if you only want to change the IMF songs. Choose Edit>Song Mappings and link audio resources if you have not already done so. This opens the Music Mappings dialog:



ID	Name	Song
0	Armageddon ...	<None>
1	Ion Vent System	<None>
2	Security Center	<None>
3	DefTunnel Vlo...	<None>
4	Energy Flow Sys	<None>
5	DefTunnel Burrh	<None>
6	Reg Control Ctr	<None>
7	DefTunnel Sorra	<None>
8	Neutrino Burst	<None>
9	DefTunnel Teln	<None>
10	Brownian Moti...	<None>
11	GravDamping ...	<None>
12	QE Dynamo	<None>
13	Korath III Base	<None>

Chunk ID	Song Title
192	Chunk 192
193	Chunk 193
194	Chunk 194
195	Chunk 195
196	Chunk 196
197	Chunk 197
198	Chunk 198
199	Chunk 199
200	Chunk 200
201	Chunk 201
202	Chunk 202
203	Chunk 203
204	Chunk 204
205	Chunk 205

Select the song chunk you want to replace in the right-hand list. Click Replace to select the new IMF file, then give it a name in the resulting dialog. If you want to add more songs than the game has space for, click Import IMF and follow the same process. This creates an extra song slot that can be worked with like any other, but requires patching to work into the game.

To assign songs to levels, click the level name on the left, select the song on the right, and click Set Song. If you added more levels or if you need songs in phantom level slots for advanced patching, click Set List Length to expand or contract the song mapping list.

All the patching for this dialog will be handled by File>Generate Patches.

VeriMaps Signing

To help players of mods ensure that the mod's level files were not modified after being saved by the mod author, Abiathar includes a security feature called VeriMaps. VeriMaps places the cryptographically signed hash of the level data in the gamemaps file where it does not affect gameplay but can be read by Abiathar or the VeriMaps inspection tool. If the level files are modified in any way, the signature becomes invalid. Because asymmetric-key cryptography is used, it is impossible to fake the signature of a level creator.

You can request a VeriMaps signing certificate by contacting Fleexy. The certificate file must be placed in the location indicated by the VeriMapsCertPath config entry to be detected by Abiathar. Once you place the file there, you can enable Edit>VeriMaps Signature to sign the level file when you save.

Note: VeriMaps does not prevent others from viewing or editing your level files. It only ensures that the last modifier is who he/she claims to be.

Tools

Abiathar's tools, though they make look like those of other common editors, contain many specialized features that can be unlocked with the use of the right button at the right time. This section contains one article for each tool in the default set, each of which contains documentation on the full power the tool has. If you have a question about a tool added by an extension, contact that extension's publisher.

All the information in this section can be accessed in-program by enabling contextual help (Help>Contextual).

Tile Tweaker

The Tile Tweaker is the most basic of Abiathar's tools. It provides the bare-bones functionality necessary to actually be a level editor: picking up and placing individual tiles. It will modify exactly one location per click - hence "tweaker." To allow pencil-like drag-and-draw functionality, use the Freeform Tile Placer.

Left mouse: For each active plane, place the plane's selected tile in the level at the mouse pointer.
Right mouse: For each active plane, copy the tile at the mouse pointer into the selected tile bay.

Freeform Tile Placer

The Freeform Tile Placer is very similar to the Tile Tweaker, but it can modify multiple tiles with one click-and-drag operation. It can be set as the default using the general configuration file. All changed tiles can be rolled back in one undo operation.

Left mouse drag: For each active plane, place the plane's selected tile in the level at the mouse pointer.
Right mouse: For each active plane, copy the tile at the mouse pointer into the selected tile bay.

Row Adjuster

The Row Adjuster handles both deletion and addition of rows into the level. It will also detect and repair links as directed in its configuration window.

Left mouse: Insert blank row(s) into the level to the left of the mouse position.
Right mouse: Delete row(s) from the level starting at the mouse position and proceeding right.
Space key: Bring up a window which sets how many rows will be inserted or deleted in one operation, as well as the policy on updating links.

Column Adjuster

The Column Adjuster handles both deletion and addition of columns into the level. It will also detect and repair links as directed in its configuration window.

Left mouse: Insert blank columns(s) into the level above the mouse position.
Right mouse: Delete column(s) from the level starting at the mouse position and proceeding down.
Space key: Bring up a window which sets how many columns will be inserted or deleted in one operation, as well as the policy on updating links.

Linker

The Linker creates infoplane links or generates the infoplane value to link to a position. It can also create bidirectional links.

In start-up mode
Left mouse: Set the starting link position (the tile whose infoplane value will be changed) to the mouse position and enter linking mode.
Right mouse: Copy the infoplane link value for the mouse's position to the infoplane tile bay and exit the Linker.

In linking mode
Left mouse: Create a link from the first point to the mouse's position and exit the Linker.
Right mouse: Create a bidirectional link between the first point and the mouse's position; exit the Linker.

Path Plotter

The Path Plotter automatically places the infoplane arrows that direct a platform given the turn points. It is, by default, only enabled for the episodes that have directional platform capability.

Left mouse: Add a turn point at the mouse position. If the new point already exists, plot the path and exit the Path Plotter.
Right mouse: Add a turn point at the mouse position, plot the path, and exit the Path Plotter.

Copier

The Copier, as one may gather, copies a rectangular block of tiles for later duplication in a different place. It can also function as a Cut tool. The Copier only works in levels; to copy from a tile palette, use the Palette Copier.

In start-up mode
Left mouse: Set the first corner of the rectangle to copy and enter anchored mode.

In anchored mode
Left mouse: For each active plane, copy the tiles of that plane inside the bounding rectangle to the clipboard; switch to the Paster.
Right mouse: For each active plane, copy the tiles of that plane inside the bounding rectangle to the clipboard and set all tiles inside to 0; switch to the Paster.

Paster

The Paster uses tiles that have been copied to the clipboard via the Copier or Palette Copier and places them as stamps in the level. It can also function as an eraser.

If the default tool is the Freeform Tile Placer, the Paster will act similarly, allowing the stamp to be smeared across the level by dragging the mouse.

Left mouse: For each plane in the clipboard, stamp the clipboard's tiles to the level with the mouse position as the upper-left corner and exit the Paster. If non-zero mode is enabled (which it is by default), only non-zero tiles will be placed in the level.
Right mouse: For each plane in the clipboard, set all tiles inside the clipboard’s rectangle to zero.
Enter key: Toggle non-zero mode.

Rectangle Filler

The Rectangle Filler fills a rectangular area in the level with one tile.

In start-up mode
Left mouse: Set the mouse position as a corner of the rectangle, enter anchored mode.

In anchored mode
Left mouse: For each active plane, fill the rectangular area determined by the first point and the mouse pointer with the selected tile; exit the Rectangle Filler.

Circle Filler

The Circle Filler fills a circular area in a level with a single tile.

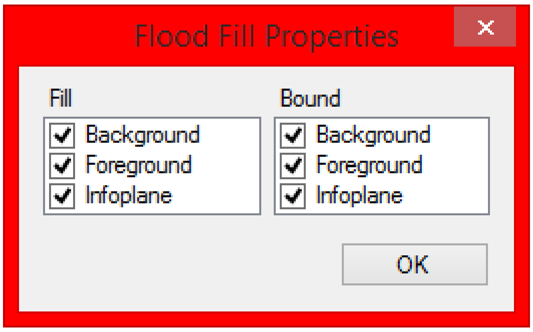
In start-up mode
Left mouse: Set the mouse location as the circle's center; enter anchored mode.

In anchored mode
Left mouse: For each active plane, fill the circle defined by the center and the mouse location (as a radius) with the selected tile; exit the Circle Filler.

Flood Filler

The Flood Filler, given a point, sets every similar tile that can reach the starting point to a tile.

Left mouse: Perform the fill operation according to the rules below, starting at the mouse pointer's location.
Space key: Bring up the fill settings dialog.



Flood Fill Properties

Fill
☑ Background
☑ Foreground
☑ Infoplane

Bound
☑ Background
☑ Foreground
☑ Infoplane

The properties determine which planes will have tiles modified ("Fill") and which planes can stop the flood ("Bound"). When the Flood Fill starts, the active planes will be set in both Fill and Bound. When the fill operation is performed, the fill will spread out from the mouse pointer, stopping if the tile in a Bound layer is not the same as that plane's selected tile and, if it can continue, place each Fill plane's selected tile in the level.

Tile Instance Locator

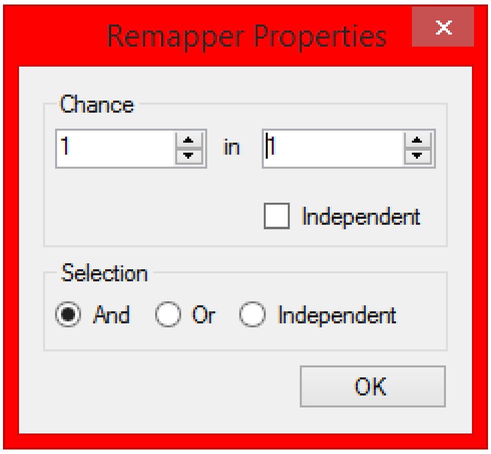
Something like a Find tool, the Tile Instance Locator highlights points in the level that contain the selected tile for each active plane. It also displays the number of such points in the lower-right.

Left mouse: For each active plane, place the plane's selected tile in the level at the mouse pointer; refresh the instance count and update the highlights.
Right mouse: For each active plane, copy the tile at the mouse position into the selected tile bay; refresh the instance count and update the highlights.

Tile Instance Remapper

Like an overpowered Replace tool, the Tile Instance Remapper is capable of advanced combinations of settings - including probabilistic replacement and replacing only tiles inside a rectangular region.

When the TIR starts, it will copy each active plane's selected tile to the find buffer. That buffer contains the tiles that will be checked when determining whether a location will be replaced.



Remapper Properties

Chance
1 in 1
□ Independent

Selection
● And ○ Or ○ Independent

In the properties, the Chance section sets the probability with which a replaceable location will actually be replaced. If Independent is checked, the chance for each plane at the replaceable location will be evaluated separately. So if Independent is not checked, either all active planes will be replaced or none will, but if it is checked, none, some, or all may be replaced. In the Selection section, the selector for whether a location is replaceable can be set. The default, And, will only select a location if the tile in each active plane's tile is the same as in its find buffer. Or will select a location if any active plane's tile is equal to its find buffer, but will still perform the replace operation on all active planes at that location. Finally, Independent will check each plane individually - but still can process the chance for each location together (if Chance's Independent check box is not checked).

In any mode
Space key: Bring up the Remapper Properties window.

Enter key: Perform the replace operation in the entire level if there is no selected region or in only the selected region if there is; exit the TIR.

In startup mode
Left mouse: Set the mouse pointer's location as one corner of the selection region; switch to region select mode.
Right mouse: For each active plane, copy that plane's tile at the mouse pointer to the selected tile bay.
R key: Switch to region select mode.
X key: Re-copy the selected tiles to the find buffer.

In region select mode
Left mouse: If the first region corner is set, set the selected region to the rectangle determined by that point and the mouse pointer's location, returning to startup mode. If not, set the mouse pointer's location as the first selection point.
U key: Cancel region select, clear the selected region, and return to startup mode.

Resource Accountant

The Resource Accountant uses tile properties and infoplane inspector configuration to count the resources available to a player in the level. When run, it will display a dialog containing the types of resources available and how many there are.

Level Inspector

The Level Inspector, as one may guess, inspects the level for problems that will cause a crash in-game. It uses some configuration information to count the number of sprites and how much memory they will take. That information is stored in the InfoplaneInspector section of the dependency file, which also has some information about the resource value of the sprites (the Resource Accountant uses these). This format is somewhat tricky and is usually unnecessary to modify. Contact Fleexy if your mod rearranges the infoplane.

The Level Inspector can check for out-of-memory errors, animation errors, scroll division errors, object-array-full errors, and various small issues. It can also check individual difficulty levels for having too many sprites.

Detected problems are put into two groups: errors (which will almost certainly cause a crash in-game) and warnings (which might result in a crash or unintended behavior).

Tile Picker

The Tile Picker is the normal tileset tool. It selects a single tile from the active palette and places it in the selected tile bay for that palette.

Left click: Copy the tile at the mouse pointer into the selected tile bay for the palette, or the foreground plane in Assimilate Infoplane mode.
Right click: Copy the tile at the mouse pointer into the selected tile bay for the palette, or the infoplane in Assimilate Infoplane mode.

Palette Copier

The Palette Copier is like the normal in-level Copier, but takes tiles from the tile palette. It copies only one plane.

In startup mode
Left mouse: Set the mouse pointer's location as one corner of the selected rectangle and switches to anchored mode.

In anchored mode
Left mouse: Copy the tiles in the rectangle described by the first point and the mouse pointer's location to the clipboard; exit Palette Copier.

Tile Property Modifier

The Tile Property Modifier, a tileset tool, can modify tile properties and animation sequences. It also provides means of copying tile properties and inspecting the tileset for animation problems.

When the TPM starts, a non-modal dialog box will appear. The controls inside it modify the currently active tile. If it is closed, the TPM will exit; if the TPM is exited, it will close.

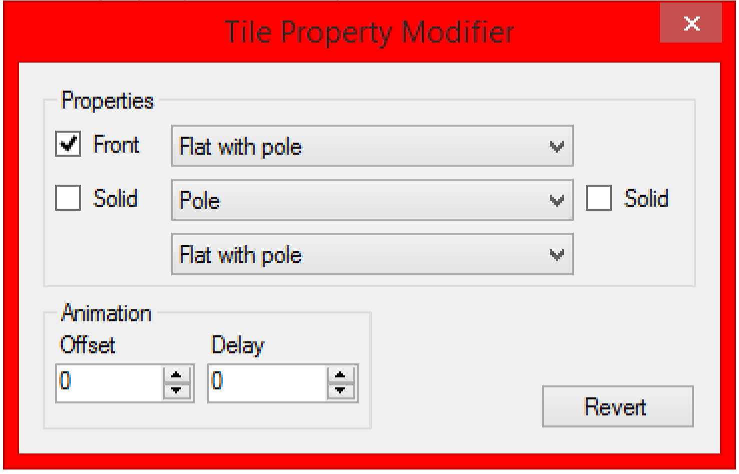
Please note that most of the functionality in this tool is disabled if the dependency file has been configured to have a read-only or null tileinfo resource.

In startup mode
Left mouse: Select the tile at the mouse pointer, display its Abiathar Tile Property ID in the lower right, make it the active tile, and highlight the tile to which it animates. Save any modifications to the previously active tile and update the detached dialog box to show the properties of the new active tile.
I key: Starting at the beginning of the tileset, check for animation link errors. If a problem is found, present details of the error and offer to move the active tile to the problem area for adjustment. Inspection mode can check for the following problems: animation to tile 0, broken loop, animation to outside the tileset, and animation collision.
C key: Copy the active tile's properties and enter copy mode.
L key: Enter continuous animation link mode.
K key: Enter non-continuous animation link mode.

In copy mode
Left mouse: Paste the copied properties to the tile at the mouse position and make it the active tile.
Right mouse: Paste the copied properties, but not animation information, to the tile at the mouse position and make it the active tile.
X key: Switch to startup mode.

In animation link mode
Left mouse: Create an animation link from the currently active tile to the tile at the mouse position. Make the mouse's tile the active tile. If in non-continuous mode, return to startup mode.
X key: Switch to startup mode.

The tile property detail dialog looks like this:



☑ Front
□ Solid
Flat with pole
Pole
Flat with pole
□ Solid

Offset 0
Delay 0

Since changes are automatically saved upon moving to a different tile, an option is provided to reset the active tile's properties to what they were before beginning to modify it: the Revert button.

The "solid" check boxes on the left and right determine whether the tile is solid on the respective side. From top to bottom, the drop-downs determine: top property (e.g. flat, slope), special effect (e.g. deadly, point item), and bottom (similar to top but affects the bottom face). Each dropdown contains properties that can be set for the tile, and add a "<custom>" entry with which you can enter a patched-in property ID. Episode numbers in brackets tell you what episodes the property has an effect in. The "front" check box determines whether the tile will appear in front of sprites. The entire Properties section is disabled for background tiles.

The Animation section is less useful because its settings are more easily changed with the animation linking mode. It simply determines the offset to the next tile in the animation sequence and how many game-ticks will elapse before changing to it. A delay of zero is used for switches and things that are only controlled by sprites. Tiles that do not animate have an offset of zero.

Format Specs

This section provides other developers with file formats Abiathar uses. It does not document the extension API, which is currently not documented anywhere. (Contact Fleexy if you're interested in creating an extension.)

editor.aconf and defaults.aconf

These files are FrofConf structures. Each datum has its own line, consisting of a name and colon (if in a normal environment) or a hyphen (if in a list), always followed by a value. Leading whitespace is not significant, but always written in a human-friendly manner by Abiathar. Compounds have their initial line's value as an open curly brace, followed by a substructure, and closed with a single right curly brace on a line. Lists are similar, but using square brackets and never naming elements. Comment lines begin with an apostrophe, optionally after whitespace, and are not parsed. See `defaults.aconf` for examples of all possible structures.

*.adeps

Abiathar stores its dependency collections in user-named files with an `adeps` extension. These files are a collection of FrofConf structures.

The first line is the "VERSION " followed by the string version ID of the Abiathar instance that saved it.

A collection of FrofConf structures follows. Before each one is a name – "Core" for Abiathar's normal configuration – followed by the structure. An extra "}" follows the structure on its own line. Finally, the string "`.~`" (a period and a tilde) ends the named structure. These continue until the end of the file and are claimed by extensions on load.

For information on legacy collection formats, contact Fleexy.

*.atemp

Abiathar template files are a single FrofConf structure, the file structure from defaults.aconf and the Core entry in an "adeps" file.

*.aslev

Abiathar-exported single level files are stored in non-human-readable files with an `aslev` extension. Their format is as follows.

- The string `Fleex NG`
- The level's name, 16 bytes long, padded with nulls if necessary
- The level's four-character signature
- Level's width as UINT16LE
- Level's height as UINT16LE
- The level's data, first RLEW compressed (flag word `$0539`), then LZW compressed in Vorticons EGA style with a maximum code size of 24 bits
    - Uncompressed data stores each tile ID as a UINT16LE
    - Layout in decreasing majority: plane, row, tile

Contact Information

If you've found a bug, have a feature request, or just want to say something to me, here's my contact information.

E-mail:
ben.nordick@mediacombb.net

fleexy@mchsi.com

PCKF/KM:
Fleexy

Skype:
ben.nordick
fleex255

IRC (Foonetic):
#Abiathar
#KeenModding
#pckf

If you're reporting a bug, please give specific details as to what you did that caused it. If Abiathar crashes and a stack trace is not available, sending the files that caused the crash would help. Before reporting a bug, try unloading extensions. If you determine that an extension is causing the crash or bug, contact your extensions' publishers for assistance.

History

- V2.3 (December 24, 2014): Fixed Tile Property Modifier once again, added “did you know?” tips, added contextual help, fixed IMF song overwrite bug, improved compatibility with TOM-saved Dreams levels.
- V2.2 (November 28, 2014): Added audio management, Song Mappings, Patches, and VeriMaps.
- V2.1 (August 6, 2014): Added the Colors dialog, fixed rendering of EGA resource colors.
- V2.0.1 (August 4, 2014): Fixed a crash bug in the New Project Wizard.
- V2.0 (August 2, 2014): Added New Project Wizard, added bitmap exporter, improved performance, improved Row/Column Adjuster, added Project Settings, added error reporter, expanded compatibility, added program icons.
- V1.4.1 (June 14, 2014): Added smart memory manager, reduced size of defaults.aconf.
- V1.4 (June 8, 2014): Made saving more convenient, added the ability to share private beta versions, drastically changed the configuration format, split the editor and template configurations, added Template Manager, added UNLZEXEr, added patch generator, fixed File Emitter.
- V1.3.2 (March 30, 2014): Fixed minor bugs.
- V1.3.1 (March 29, 2014): Tweaked the TPM yet again, added the Help menu, made the Tile Instance Locator not useless, fixed minor bugs.
- V1.3 (March 27, 2014): Added the inspectors and point counter. Tile Property Modifier might actually work now.
- V1.2.4 (March 21, 2014): Fixed another bug in the Tile Property Modifier, fixed bounded scrolling (for real this time), and fixed some internal inconsistencies in Simultaneous Tileset mode.
- V1.2.3 (March 20, 2014): Fixed a bug in the Tile Property Modifier and made bounded scrolling a little more convenient.
- V1.2.2 (March 20, 2014): Fixed the Freeform Tile Placer’s single-tile overlay and added the Core/Defaults config options.
- V1.2.1 (March 19, 2014): Fixed some GUI bugs, hid the level signature box for profiles that don’t use it, and re-added Abiathar’s signature to the resulting game maps files.
- V1.2 (March 19, 2014): Improved the Tile Property Modifier and changed a few default config settings. Added the ability to modify level signatures and changed how the Paster works. Fixed a few bugs reported by Ceilick. Improved the internal versioning system.
- V1.1.1 (March 17, 2014): Fixed a very bad bug that caused Abiathar to delete all levels if a gamemaps file is loaded that is missing level 0. Some other GUI bugs were also fixed.
- V1.1 (March 17, 2014): Added tileinfo modification, lots of internal polishing, some GUI glitch fixes, help file corrections.
- V1.0 (March 14, 2014): Original release!

Many little-known historical events contributed to the creation of Abiathar. This timeline follows the history of Commander Keen up to Abiathar's release.

- December 14, 1990: "Marooned on Mars", the first Keen game, is released as shareware.
- December 15, 1991: "Secret of the Oracle", the first Galaxy Keen game, is released as shareware.
- December 18: 1991: "The Armageddon Machine", the chronologically last Keen game, is made available for purchase.
- November 22, 1995: TED5, the original Keen: Galaxy editor, is compiled.
- March 4, 2002: Andy Durdin releases TEDSETUP, making Galaxy level editing possible.
- January 13, 2006: "z-one" releases the first version of Keen: Next, the first Windows-based Galaxy editor
- January 23, 2008: Alan Faubert ("ckguy") releases the first version of TPT, another Win32 Galaxy editor
- July 12, 2010: Gregory Yauney ("Mink") releases the first version of The Omegamatic, yet another K:G editor
- March 14, 2011: Mink releases the final version of TOM with Dreams support
- December 21, 2013: Fleexy starts work on a full Keen resource API
- January 7, 2014: Fleexy finishes the Galaxy and Dreams level API in FleexCore 2
- January 10, 2014: Fleexy announces Fudge, a Dreams editor which will eventually become Abiathar
- February 16, 2014: Fudge is renamed to Abiathar, already mostly complete
- March 14, 2014: Abiathar is publicly released!

Acknowledgements

A lot of people, some knowingly and some not, contributed to Abiathar's development. I would like to dedicate this final section to them.

- **Andy Durdin "adurdin":** Published "Plumbing the Depths of Keen" and other works that advanced the entire Keen community's ability to create amazing derivative works. He also had a huge part in making available the first Keen: Galaxy editor, TED5. Though he doesn't post on the forums anymore and probably will never see this, every member of PCKF and K:M should thank him for his hard work.
- **Brian Clavadetscher "Ceilick":** Inspired all of us to reach for the Galaxy with his amazing "The Keys of Krodacia" back in 2005. Also performed critical testing, suggested some very important features, and reported most of the bugs.
- **Sarah Williams "Levellass":** Reverse engineered a lot of the graphics and level formats, or at least wrote them up. She also did most of the beta testing *months* before any of the general population got to see Abiathar. Fortunately for the rest of us fools, the Lass is still around helping out with this and that.
- **"lemm":** Did a little bit of beta testing. Most importantly, he received the first ever deployment of Abiathar, which didn't actually start up at all. When we did get it working, he found some very confusing GUI text and helped fix it.
- **“Gridlock”:** Suggested lots of features, did lots of testing.
- **Alan Faubert “CK Guy”:** Published documentation on the tileinfo values.
- **Spencer B.:** A personal friend of mine, did some of the most important beta testing for Abiathar. He actually isn't a Keener at all, and that's exactly why I wanted him to be able to work this program. Thanks to his messing around, I was able to observe user behavior that I wouldn't have otherwise expected.
- **Gregory Yauney "Mink":** Created The Omegamatic, which nobody will ever use again because they have Abiathar now! ;) TOM is an excellent editor and served the community well for three years. Mink is also a great guy in general.
- **“wiivn”:** Made the 16x16 and 64x64 program icons.
- **“MoffD”:** Made the 32x32 program icon.
- **William Kulich "Keening_Product"**

Thanks for using Abiathar!

Ben Nordick
"Fleexy"