What is Lunar Magic?

Lunar Magic is a level editor for the American and Japanese version 1.00 Super Mario World SNES ROMs, and the SMW portion of the American version 1.00 Mario All Stars + World SNES ROM. It's a Windows program with a fairly easy to use WYSIWYG interface that includes clipboard and drag/drop support, external level file saving and loading, graphics editing, palette editing, enemy support, world map editing, text editing, plus numerous 65816 ASM enhancements to expand the original game's capabilities.

The project was first started in February of 2000, and has been worked on and improved at various times over the past 20 years. Although the program took a substantial amount of time and effort to build along with figuring out how SMW worked, the result has been a rather powerful editor that should provide a fairly easy way for just about anyone to make their own customizations of this classic Mario game.

Anyway, have fun making your own SMW levels! ^^

Programming

FuSoYa (Defender of Relm)

3rd Party Enhancements

VRAM optimization patch - smkdan
(testing - Vic Rattlehead)
Dynamic Levels patch - Vitor Vilela
Memory Item 3 Index - BMF
FastROM Address List - Ersanio
LC_LZ3 patch - edit1754

LC_LZ2 patch - Ersanio, edit1754, 33953YoShI, Min, smkdan
LMSW DLL - Alcaro

Development Tools

ZSNES debugger
Jeremy Gordon's 65816 assembler (custom build)
SNES Professional ASM kit (SPASM)
SNES9X tracer feature
Naga, Tile Layer Pro
UltraEdit 32
Borland's C++ IDE version 5.02
MSVC

What's New in Lunar Magic

Version 3.31 September 24, 2021 (21 year Anniversary of Lunar Magic!)

- fixed a bug from 3.30 where Control + Shift + Page-Down would no longer work in the 8x8 Overworld Tile Selector, although viewing the tiles could still be enabled with the new option. Thanks goes out to mariofreak4500 for reporting this.
- fixed a bug from 3.30 where the new custom sprite GFX information in the "Add Sprites" window would use the custom GFX information of regular sprites without any extra bit set for all extra bit settings. Thanks goes out to Rykon-V73 for reporting this.
- fixed a bug from 3.03 where the "Tile Surface Outlines" view option was not immediately updating the outlines when the "POW" view option was changed and tile animation was turned off. Thanks goes out to TheBiob for reporting this.
- fixed a bug from 3.00 with the sprite "Smart Spawn" option where if it was enabled and you went to the right edge of a 32 screen wide horizontal level there was a chance the game could try parsing random bytes for the sprite list, causing lag spikes and other strangeness. Thanks goes out to yoshifanatic for reporting this.
- fixed a bug from 3.00 where the "Process While Offscreen" sprite flag was not being honored when doing vertical despawning checks in horizontal levels. Thanks goes out to TheBiob for reporting this.

- fixed a bug from 2.41 where the custom user toolbar was not set up to correctly wrap on program startup or window maximize/restore. It would only work on window stretching. Thanks goes out to Imamelia for bringing this up.
- fixed a bug from 2.00 where the LM_NOTIFY_ON_NEW_LEVEL option for custom user toolbars did not send a notification when saving a level with a different level number. Thanks goes out to Underway for reporting this.
- added LM_NOTIFY_ON_SAVE_LEVEL, LM_NOTIFY_ON_SAVE_MAP16, and LM_NOTIFY_ON_SAVE_OV to button options for custom user toolbars.
- made it possible to load/save edits to the bottom row of tiles for the title screen image, as the game shows the top pixel of this row.
- tweaked undo and delete in the event editor modes of the overworld editor so it's a bit nicer about maintaining the current event step progress shown.
- removed "Custom Sprites" from the level editor's view menu, as it's unlikely anyone needs to turn that off.
- added "Exit Enabled Tiles" to the level editor's view menu to allow seeing which tiles are exit enabled in the level editor. You can also have the editor treat custom blocks that use screen exits as exit enabled (for undefined exit scans or for viewing) using the tooltip file.
- added largest free area within a 32KB bank that can be protected with a RAT to "Scan ROM" information, and converted it to a regular dialog window. Also updated the help file on what the values mean.
- conflicting RATs detected using "Scan ROM" will now have their locations logged to the RATS.log file.
- added a new "VRAM Patch Options for this ROM" dialog in the options menu, which can disable the VRAM patch

from being installed if it hasn't been already. This is now a per-ROM setting.
- removed the "Install VRAM Patch on Save to ROM" option from general options, as this has now been replaced by the new dialog listed above.
- added support for unicode filenames/paths when running on a unicode OS.

Version 3.30 May 1, 2021

- added an "Auto-Enable custom palette on edit" option to the level and overworld palette editors.
- added edit buttons next to the graphics slots in the bypass dialogs that can open the GFX/ExGFX file in an external tile editor. It can also optionally replace the yychr.pal file with the current palette, and offer to create the graphics file if it doesn't exist in the folder or ROM. Also added a dialog to configure it in the "File", "Graphics" menu.
- added the ability to set separate appearances and tooltips for sprites in the editor based on the lower bits of a single extension byte (see "Technical Information" in the help file for more details).
- added the ability to display custom sprite GFX information in the "Add Sprites" window (see "Technical Information" in the help file for more details).
- made it so that original sprites that are set to have extra bytes will still display the sprite in the preview area for the original lists in the "Add Sprites" window.
- added a bit of ASM code to store the level number to $10B, as some people may find it useful.
- added an option to General Options that allows the 8x8/16x16 editors to show other tiles without having to

press Control + Shift + Page-Down, which is disabled by default.

- added an option to DirectX windows for 100% zoom increments. Note that with this enabled and the zoom filter option disabled, you can now get the same non-filtered zoom display and behavior of past versions without having to disable the DirectX option and restarting the program.
- removed the option in General Options for disabling the menu for the zoom button, as it probably isn't used much these days.
- added a new option to the "Extra Options" dialog of the overworld editor that controls if translucency (CGADSUB) is enabled for palettes C-F in sprites when the event path fade effect is off. This will now be turned off by default when you disable the path fade effect, in case you want to use those palettes for regular non-translucent custom sprites.
- made it possible to type level and event numbers in the drop down list boxes for some of the overworld editor dialogs.
- changed the remap button in the 8x8 Overworld Tile Selector window so that it applies to all tiles by default in layer 3 editing modes unless you use F9.
- made it so that when you copy a tile to the windows clipboard in the 8x8 tile selector of the Map16 editor or the 8x8 tile selector of the overworld editor, it will also copy the tile's hex value in text to the clipboard so you can paste it into edit fields in the ExAnimated Frames dialogs.
- fixed a bug from 3.20 where if you used any of the 8x8 Overworld Tile Selector Tools to make changes to a selected area in the overworld then attempted to resize it as a pattern of tiles, the tiles would first revert back to how they were before the changes were made. Thanks goes out to BlueToad for reporting this.

- fixed a bug from 3.20 where if the DirectX option was on, animation was off and zoom was at a multiple of 100%, the level/layer 2/overworld editor windows would sometimes appear to scroll 1 tile too far with a row/column of tiles repeated until the screen was refreshed.
- added a fix for sprite EF (scroll layer 2 sideways) so that it can now be used in tall horizontal levels and not just in vertical levels.
- adjusted palette used for a tile in sprite A2 (MechaKoopa) in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to WhiteYoshiEgg for reporting this.
- added a warning about palette usage to the Big Boo Boss tooltip. Thanks goes out to Klug for bringing this up.
- the .msc, .dsc, .ssc, .sscov, .extmod, and usertoolbar.txt files must now use UTF-8. Existing english text files are not affected, but other languages will need to change the encoding used for those files.

Version 3.21 October 24, 2020

- fixed a bug from 1.80 where the restore system would not correctly restore auxiliary files that were larger than 64KB although the files in the restore point were intact. This was more noticeable in version 3.20 as the .s16 file for custom sprite Map16 went beyond 64KB for the first time. Thanks goes out to Imamelia for submitting his ROM folder to figure this out.
- made it so that saving the .s16 file in the Map16 editor for custom sprite Map16 only saves up to the last page

of data that was actually used.

Version 3.20 September 24, 2020 (20 year Anniversary of Lunar Magic!)

- added an option to general options to use DirectX 9 in the main level editor window, background layer 2 editor window, and the overworld editor window, which allows using a zoom filter. If DirectX is not available, it will automatically fall back to using GDI instead.
- added a zoom filter option to the zoom menu for windows that use DirectX, along with a few more zoom levels (125%, 150%, 175%). Also made it so that using Zoom in/Zoom out (Ctrl+Scroll Wheel) in those windows will go in increments of +/-10% when DirectX is available.
- added a button to the main level editor that opens the overworld editor and loads layer 3 of the level into it.
- made it so that if layer 3 of the level is loaded in the overworld editor and tile animation is on, changes in the overworld editor are reflected immediately in the level editor.
- added the ability to display sprites in the editor using external graphics by making it possible to set the base graphics for individual sprite Map16 tiles in the sprite tooltip file (see the help file for more details). The external graphics must be in the "ExternalGraphics" folder and be named "ExSpriteGFXxx.bin" where "xx" can be from 00-07 and the files can be up to 32KB in size (these files are currently loaded starting at page 0x20).

- added the ability to display sprites in the editor using external palettes using a setup similar to external graphics (see the help file for more details). The external palettes must be in the "ExternalGraphics" folder and be named "ExSpritePalette00.mw3" (or "ExSpritePalette00.pal") and can hold up to 0x400 palettes of 0x10 colors.
- added an extra 0x18 pages for custom sprite Map16 data.
- added an "Export Multiple Levels to Image Files" item to the "File", "Levels" menu which can export multiple levels to images at once.
- added an option to report on music tracks used in levels to the "Analyze Resources in Levels" menu command.
- added an option to use custom command line arguments for running the currently loaded ROM in an external emulator.
- added an option to "General Options" to show old/obsolete menu items, which are now hidden by default.
- added the ability to fill an area in the Background Editor with a pattern of tiles from either a selected area in the Background Editor (with Shift + Right Click) or from the Map16 Editor (with Control + Shift + Right Click).
- added the ability to resize an area in the Background Editor as a pattern of tiles by using the mouse on resize borders, similar to how you can resize Direct Map16 objects in the level editor.
- added the ability to resize an area in the Overworld Editor in Layer 1 16x16 Editor Mode or Layer 2 8x8 Editor Mode as a pattern of tiles by using the mouse on resize borders, similar to how you can resize Direct Map16 objects in the level editor. With 8x8 editing however, you must hold down the control key to get the mouse resizing cursor if zoom is below 200%.

- added an option to the overworld editor that forces using the control key to resize a group of tiles.
- added buttons to clear text and clear all text to the "Edit Boss Sequence Text", "Edit Message Box Text", and "Edit Level Names" dialogs of the overworld editor.
- added LM_NO_CONSOLE_WINDOW as a button option for user toolbars, which can be useful for running .bat files or other console programs without showing a console window if they never require user input.
- updated the sprite 19 fix (for displaying level message 1) so it will no longer cause the intro level to change from 0xC5/0x1C5 depending on the starting submap, and made it now install automatically on level/overworld save. However to avoid disruption for people with older hacks, it will not change the existing 0xC5/0x1C5 behavior if the old fix is installed. Also Shift+F8 will remain available to cause the 0xC5/0x1C5 behavior for those that still want it for porting older hacks.
- removed the warning option for when sprite 19 is used without the fix, as it's no longer needed.
- added a new option to the overworld editor that can save the game after the intro message with sprite 19 even if Mario's starting position on the overworld has been changed (normally that save gets disabled along with the intro march on the overworld if the position is changed).
- fixed a small issue from 3.00 where the "Scan Exits on Save to ROM" option didn't exclude exits to the overworld when checking the level destination. Thanks goes out to Ninja Boy for reporting this.
- fixed a bug from 3.00 where if Mario did a cape spin while entering a pipe he would do zigzags when shot out of a diagonal pipe. Thanks goes out to Aurel509 for reporting this.
- fixed a bug in the original game with sprites 0xC9 and 0xCA (Bullet Bill shooter and Torpedo Launcher) that can

cause the game to keep reading right past the end of the level's sprite data which could cause lag, seemingly random sprites to appear, or a bunch of green Koopas without a shell to appear. Thanks goes out to Tattletale for reporting and submitting the fix for this.

Version 3.11 February 9, 2020

- fixed a bug from 1.70 in smkdan's VRAM patch where pipes on layer 2 in a horizontal level may be displayed with the wrong colors on level load, which changed to the right colors if the player left the initial screen and came back.
- fixed a bug from 2.20 for SA-1 ROMs where Conditional Direct Map16 objects could not read their RAM table. Thanks goes out to HammerBrother for bringing this up.
- fixed a display issue in the overworld editor where it was still showing how the original game handled destroying castle tiles placed at Y=0xF or 0x2F even though it was already fixed in-game for 3.10. Thanks goes out to Thomas for reporting this.
- made it possible for 3rd party sprite programs to notify LM that the sprite count limit per level has been increased to 255 to avoid the sprite count warning message appearing.
- added an option to disable all the hardcoded default layer 1 paths to the overworld's "Extra Options" dialog.
- added a new "Fast" layer 3 scroll setting that uses a 1.2:1 scroll rate, which goes faster than layer 1. This is intended for the type of foregrounds you can see in SMW2:Yoshi's Island.

Version 3.10 September 24, 2019 (19 year Anniversary of Lunar Magic!)

- added support for tides in horizontal levels that are taller than in the original game.
- added support for having vertical scroll in tide levels when you use the "Constant" vertical scroll option for layer 3.
- made the layer 3 auto-vertical scroll options detect tides to support vertical scroll of the level.
- added support for negative Y scroll offsets for layer 3 and greatly expanded the possible range of values (-400 to 3FF) to better allow for adjusting the tide height. The layer 3 scroll offsets are also now displayed in units of 16x16 tiles instead of pixels.
- made it possible to have tides in horizontal levels that are 1-2 screens wide if you enable the advanced bypass settings in LM.
- added support for tides in vertical levels if you enable the advanced bypass settings in LM (minimum level height of 2 screens required).
- added a new setting for layer 3 that lets you make the tides act as water, 3 types of lava, solid, or tiles 0x200-0x20A.
- added a new setting that makes sprites beyond level boundaries interact with air instead of water, which should usually be turned on for tides.
- added ExportSharedPalette, ImportSharedPalette, ExportAllMap16, ImportAllMap16, ExportMultLevels, ImportMultLevels, TransferLevelGlobalExAnim, TransferOverworld, TransferTitleScreen, TransferCredits,

ExportTitleMoves, ImportTitleMoves to command line functions.

- added a separate "Auto-Set Number of Screens" option to the "Import Multiple Levels from File" dialog.
- added an extra 0x10 pages for custom sprite Map16 data.
- added a new ASM hack to make it so that revealing/destroying overworld layer 1 tiles will show tiles from the same page instead of temporarily showing tiles from page 0.
- fixed an issue in the original game where castle tiles placed at Y=0xF or 0x2F on the overworld would not collapse correctly when destroyed.
- fixed a crash bug from 3.00 that happened if you tried to right click to paste objects/sprites in a boss battle level that can't be rendered. Thanks goes out to TheBiob for reporting this.
- fixed a bug from 2.30 where using the %5 placeholder in custom user toolbars would cut off the rest of the string. Thanks goes out to SmokedSeaBass for reporting this.

Version 3.04 June 1, 2019

- fixed a crash bug from 3.03 that could happen if the "Select All" command was used on objects in the level editor that were then moved or deleted.
- fixed a crash bug from 3.02 that would occur if you tried to do a remap in the Map16 editor on pages A3+. Thanks goes out to Koopster for reporting this.
- fixed a bug from 3.00 where if you went beyond the top of the level with Yoshi, he could eat berries that were on

the previous screen. Thanks goes out to Thomas for reporting this.

- fixed a bug in the original game where if Mario flew beyond the top of the level, he might interact with block numbers formed by the lower 8 bits of a block number that another sprite was interacting with. This was particularly noticeable if the resulting block number was for an invisible question block or jumping note block. Thanks goes out to Thomas for reporting the issue that led to figuring this out.
- fixed an issue from 2.20 for SA-1 ROMs where if you re-enabled the path fade effect and saved the overworld, a warning message about a failed pointer remap would appear (although the ROM would be fine). Thanks goes out to DPBOX for submitting the hack that revealed this.
- made the "Block Contents" view option show the contents somewhat translucent.
- added a few extra options for the "Analyze Resources in Levels" menu command.

Version 3.03 April 1, 2019

- fixed an issue with some code in the original game where if Mario was above screen 0 in a horizontal water level, there was a chance it could cause the game to freeze (with LM 1.30+) or crash (depending on code inserted by block tool programs). Thanks goes out to GreenHammerBro for reporting this.
- fixed a bug from 2.43 where using F3 to copy then paste a color in the palette editors would temporarily display the original PC color instead of the converted SNES color, and the converted SNES color could be slightly off

from what it should have been. Thanks goes out to DPBOX for reporting this.
- added "Block Contents" to the level editor's view menu to allow seeing the contents of question blocks, etc in the level editor. You can also specify contents to show for custom blocks using the tooltip file.
- combined viewing the main, midway, and secondary entrances into a single menu item to simplify the view menu a bit and reclaim a couple keyboard shortcuts.
- added "Analyze Resources in Levels" to the level editor's "File", "Levels" menu that can check which Map16 tiles, GFX/ExGFX files, and sprites are used in which levels then generate a text file report.
- added "Select All" to the level editor's "Edit" menu for selecting all sprites or objects of the layer you're editing, depending on which mode you're in.
- made the extension field in Add Object/Sprite Manual dialogs wider. You can also now put optional spaces between the values.
- coordinates in the status bar of the level editor are now in hex instead of decimal.

Version 3.02 February 9, 2019

- fixed a bug from 3.00 where ROMs upgraded from version 2.43 or earlier of LM (meaning you had never used 2.5x on it) didn't convert secondary entrances that used method 2 for horizontal levels, resulting in their positions being reset to the top subscreen. If you've already used 3.00/3.01 on such a ROM, you can fix it for all levels by using Alt+Shift+F11 in this version then saving the current level.

- fixed a crash bug from 2.50 that would occur if you imported a Map16 file in the Map16 editor while the remap dialog in the BG Editor was open at the same time. Thanks goes out to MarioFanGamer for reporting this.
- fixed a limitation in the original game with tileset specific object 33 (forest tree top) so that it will now work correctly at any Y offset instead of just Y=10 to 15. This also fixes it to work in horizontal levels that have different heights than the default. Thanks goes out to Shiny Ninetales for reporting this.
- added "Line Guide Outlines" as an option in the view menu, for cases where you want to use line guided tiles yet don't have the right graphics loaded to see them.
- added the ability to set a rectangle of Map16 tiles to use a rectangle of "Act as" settings with the specified base (such as R200-211,S25) in the Map16 editor's remap dialog.

Version 3.01 January 1, 2019

- fixed a bug from 3.00 where vertical levels with a "No Yoshi" intro would have tiles missing in the intro. Thanks goes out to Shiny Ninetales for reporting this.
- fixed a bug from 3.00 where if there was only a horizontal scroll bar you couldn't use the scroll wheel to move objects backwards/forwards by one. Thanks goes out to Daizo Dee Von for reporting this.
- fixed a bug from 2.41 where the warning for sprite 19 (display level message 1) could be triggered in level 0xC5 when using the "Import Multiple Levels from File"

function. Thanks goes out to WhiteYoshiEgg for reporting this.
- fixed a bug in the original game where if you placed Mario's starting position near the far right side of a horizontal level, the game would not set the initial X scroll position correctly.

Version 3.00 December 25, 2018

- integrated an ASM hack submitted by Vitor into LM that allows changing the height of horizontal levels, effectively making it possible to change the level dimensions. You can find the new setting controlling this in the "Change Properties in Header" dialog, along with a new option that lets you view the full bottom row of tiles in horizontal levels.
- the sprite loader now has a cache added by Vitor to reduce the performance impact from parsing most of the sprite list to access sprites on later screens.
- added a setting to the "Change Properties in Sprite Header" dialog to control the sprite vertical spawning range for horizontal levels and an option for smart spawning.
- levels can now have up to 128 sprites instead of 84 for non-SA1 ROMs.
- a limitation of the original game where sprites could not start with an FF byte has been removed (previously LM would silently move the sprite on save when it encountered this to avoid premature termination of the sprite list for the level).
- in vertical levels most objects will no longer break up on horizontal subscreen boundaries (but they still will on

vertical ones).

- added an option for entrances to use a new FG/BG init system which can set the FG relative to the player, and calculates the BG position relative to the FG position, scroll settings, level height, and BG height. The BG height is a new setting in the "Change Other Properties" dialog, along with an option to just set the BG relative only to the FG which is meant mainly for layer 2 levels.
- added 4 new vertical scroll settings for layer 2 (Variable 2, Variable 3, Variable 4, and Slow 2). Note that these take up slots that were previously blank (H/V Scroll of None), so you may want to double check your scroll settings in older levels. To have H/V scroll set to None, you should be using the 4th entry in the list rather than one of the later ones.
- added a new option for entrances to have Mario face the left direction.
- made it so that using the "Shoot From Slanted Pipe Right" entrance action combined with the "Face left" option will make Mario shoot left instead of right.
- added a new option for midway entrances to redirect a midway entrance to another level. This allows you to have the midway entrance for an overworld level be in another level entirely.
- added a new option for secondary entrances to make the level a water level. Note that the older option for the same thing in screen exits can also still be used.
- added a new option for secondary entrances to exit to the overworld. You can also optionally set an exit to the overworld to pass the level with the normal/secret exit of your choice or just switch players, use a different base event when passing the level, and teleport to a different location.
- added a new toolbar button to the overworld editor that can set secondary exit teleport locations, which uses the same table as the Star/Pipe tiles.

- made it possible for the overworld Star/Pipe index table to hold twice as many entries (0x100), to accommodate secondary exit teleport locations.
- made it possible to use Secret Exit 2 and Secret Exit 3 in the game. This means the overworld editor now has direction to enable settings on level tiles for both of these, and the Secret Exit 2 and Secret Exit 3 goal point tape sprites have been added to the sprite list.
- made it possible to change level tile settings for tiles 0x82-0x86 in the overworld editor. While they can't be entered as levels, they are valid stopping points so this allows you to assign them a level number in case you want them to display a name or give them initial enabled directions.
- moved the music and time limit settings from the "Change Properties in Header" dialog to the "Bypass Music and Time Limit Setting" dialog, then renamed the latter to "Change Music and Time Limit Settings".
- made it so that when you copy a color to the windows clipboard in the palette editors, it will also copy the SNES RGB hex value in text to the clipboard so you can paste it into edit fields in the ExAnimated Frames dialogs.
- added a new toolbar button to the level editor and overworld editor that will insert all GFX and ExGFX then reload the graphics.
- added a new view menu option that allows viewing tile surfaces. This is mainly meant for showing the shape of slopes and for revealing if a tile is solid or not.
- added a few basic command line functions for some common operations. See the technical section in the help file for more info.
- added a new option to the overworld editor that allows disabling the Forest of Illusion ghost from being hidden until an event has been passed. Also made it so you can change the events that it gets shown on.

- when linking star/pipe tiles, the program no longer requires that the second tile be a star/pipe tile. This makes it easier to create warps from a star/pipe tile to tiles that you can't warp from.
- changed the Modify Screen Exits dialog to allow typing full values directly into the index combo box for faster selection.
- changed how the "Insert Manual" dialog works for objects and sprites in the level editor. Instead of specifying X/Y and the screen number, LM will now simply insert the object or sprite near the upper left corner of the level editor window.
- made it so that the sprite size table is loaded on level load, not just on ROM load.
- added "Delete All" to the level editor's "Edit" menu, which was previously only available through the Ctrl+Del keyboard shortcut.
- moved Shift+Scroll Wheel for Bring to Front/Send to Back to Ctrl+Alt+Scroll Wheel, added Ctrl+Shift+Scroll Wheel for vertical scroll, and made Shift+Scroll Wheel and Ctrl+Shift+Scroll Wheel unaffected by objects/sprites selected.
- tweaked some code so that scroll bar thumbs are now sized proportionally more like the way you'd expect. Thanks goes out to Vitor for pointing this out.
- fixed a bug from 2.43 where the undo/redo/F3 keys in the palette editors would not function while any other modal dialog window in the program was open at the same time.
- fixed a crash bug from 2.30 that could occur under certain circumstances when selecting tiles in the Layer 1 Event Editor Mode of the overworld editor. Thanks goes out to Wiimeiser for reporting this.
- fixed a bug from 2.30 where if you saved a level then used undo to revert a change to a secondary entrance then saved again without making any other changes to

secondary entrances, LM would not save the reverted change for the secondary exit. Thanks goes out to Ramon for reporting this.

- fixed a bug from 1.70 in smkdan's VRAM patch where if you rapidly scrolled past the top of the BG image (such as by setting layer 2's vertical scrolling to constant), there was a chance you could get a brief 1 frame glimpse of a couple rows of pixels at the top or bottom of the screen showing the wrong tiles. Thanks goes out to Vitor and Super Maks 64 for reporting this.
- fixed a bug from 1.10 where the program would not load/save/export/import the last color of the Special World shared palettes for when Special World had been passed. Thanks goes out to yoshifanatic for reporting this.
- corrected the description for Mario Action 7 of midway entrances to "Do Nothing or Pipe Exit Down (Water Level)" and changed how it's displayed in LM. Normally midway entrances that use this setting skip the pipe exit part. It never came up in the original SMW levels, though it seems to have been used in SMA2. However it still acts as a pipe entrance if accessed via a screen exit. To avoid confusion, it may be better to instead use the "Vertical Pipe Exit Down" action with the separate "Make this a Water Level" setting.
- renamed Layer 1 vertical scroll setting 2 to "No Vertical Scroll at Bottom unless Flying/etc" to more accurately describe what it's for. Also adjusted the code for this setting to work for horizontal levels that have different heights and vertical levels.
- fixed an issue in the original game where when switching between levels if you entered a level with layer 2 horizontal scrolling set to None, the layer 2 X position was carried over from the last level instead of being reset to 0 like it does when entering from the overworld.

- fixed an issue in the original game where if a "No Yoshi" entrance intro is activated for a vertical level, the layer 2 Y position in the intro could be wrong and even possibly display garbage tiles.
- fixed an issue in the original game where using a pipe exit at the very top of a vertical level could malfunction if Mario's Y coordinate ended up being beyond the top of the level.
- added a fix for the original game where the sprite load status for indexes 64-127 were not cleared between sublevels, leading to some sprites not appearing if a sprite at that index was killed in another sublevel.
- adjusted tile used for sprite 99 (Volcano Lotus) in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to WhiteYoshiEgg for reporting this.
- added a bit of info for sprite 70 (Pokey), as it can't detect if Mario is riding Yoshi if it's within spawning distance of Mario entering the level. Thanks goes out to Wiimeiser for pointing out that something was up with this.

Version 2.53 February 9, 2018

- fixed a bug from 2.40 with ExAnimation on the overworld where if you tried to use multiple slots for the same destination tile to do faster animations, the frames would not always appear in the order you'd expect. Thanks goes out to Noivern for reporting this.
- if overworld ExAnimation is installed, it will also now fix a small issue present in the original game where the

bottom waterfall tile animation would not consistently match up with the main waterfall tile.

- fixed a glitch in the original game where sprite 52 (moving ledge hole) would appear in the wrong position if placed at the top of a sub-screen. Thanks goes out to Thomas for discovering this, and zacmario for reporting it.
- fixed a minor issue from 1.70 where the timing of the victory pose at the end of the level didn't really match up to the music in non-SA1 ROMs.
- corrected info on sprite 45 (directional coin) as it does in fact have a time/coin limit. Thanks goes out to Katerpie for reporting this.
- replaced some non-standard apostrophe chars in some tooltips in the General Options dialog that could get displayed oddly on non-english systems. Thanks goes out to Akaginite for reporting this.
- fixed a small issue that's been around forever where using a mouse gesture on a screen exit that goes to a vertical level did not scroll the screen for the corresponding entrance of that level into view.
- made it possible to use Ctrl+Alt+F7 in level editor for advanced users to edit more local ExAnimation slots by sacrificing global slots (when global animations have been disabled for level/submap).
- made it possible to use Ctrl+Alt+Shift+F8 in overworld editor for advanced users to enable editing more levels and events (the game does not currently support this at the moment, so don't enable it unless you're adding your own ASM for it).
- added the ability to drag and drop a ROM, mwl, map16, or palette file into the main level editor window to open the file.
- made it possible to use Alt-Middle click to edit screen exits.

- added LM_MOUSE_EDIT_SCREEN_EXIT to function names for user toolbars.
- added an option to open levels at the main level entrance.
- added an option to pause animations when the program loses focus.
- added an option to display the ROM file name and path in the main window title bar.

Version 2.52 September 30, 2017

- fixed a bug from 2.50 where SA-1/SFX ROMs upgraded from a previous version of LM could use the wrong "Act as" settings during gameplay after the Map16 was saved. Resaving the Map16 in this version will correct the issue. Thanks goes out to Christian07 for reporting this.

Version 2.51 September 25, 2017

- fixed a crash issue from 2.50 with the 8x8 tile selector in the Map16 editor. Thanks goes out to yoshifanatic for reporting this.

Version 2.50 September 24, 2017 (17 year Anniversary of Lunar Magic!)

- made it possible to resize objects from top/left and related corners, for most objects that support it. Note that a few objects that resize in increments larger than 1 or in opposite directions may seem to resize a bit oddly.
- Map16 space has been expanded to 2x the previous limit. There are now 0x80 pages for FG Map16, and 0x80 pages for BG Map16. Note that if you use a third party block tool type program, you shouldn't use the new FG tiles (0x4000-0x7FFF) in LM until your program has been updated to deal with the new max.
- expanded the number of secondary exits in the game to 0x2000, up from 0x200.
- changed the Modify Secondary Entrances dialog to allow typing full values directly into the index combo box for faster selection.
- added tooltips to sprites in the overworld editor.
- added the ability to add custom sprites in the overworld editor with the insert key. Note that these require an external third party patch to show up in the game.
- made it possible to use alt-right click in the overworld editor to modify custom sprites.
- made it possible to use right click in the overworld editor to copy custom sprites.
- made it possible to customize sprite appearances and tooltips for sprites in the overworld editor (editor display only).
- added LM_FILE_RELOAD_ROM and LM_MOUSE_SCREEN_EXIT to function names for user toolbars.

- added VK_MBUTTON, VK_XBUTTON1, VK_XBUTTON2 (for middle mouse button, mouse button 4, mouse button 5) to keyboard definitions for user toolbars.
- fixed a bug from 1.70 where the ExAnimation dialog would limit the max number of frames to 0x80 based on the slot's type when it was supposed to be based on the trigger. Thanks goes out to Skewer for reporting this.
- fixed a bug from 2.43 that allowed saving with an invalid level number. Thanks goes out to KDeee for reporting this.
- fixed a bug from 2.30 where if you used LM's layer 3 auto-scroll on tide levels with buoyancy enabled that were 3-8 screens long, sprites would tend to act like they were in water periodically even if they weren't (note that this still happens for levels 1-2 screens long, but the original game also does this). Thanks goes out to Thomas for reporting this.
- adjusted some code so that closing the overworld editor while minimized would no longer cause a few odd things on reopening. Thanks goes out to Erik557 for reporting this.

Version 2.43 December 25, 2016

- added a new keyboard shortcut of F3 to the palette editors to allow copying the color of any pixel on the computer screen to the windows clipboard. Note however that the palette editor must have the window focus, and it may not work on all windows or protected media.

- enabled undo/redo keyboard shortcuts in palette editors.
- updated filter in Add Objects window for extended object 7F (skull and crossbones) for ghost houses. Thanks goes out to imamelia for reporting this.
- adjusted the appearance of sprite 0x7A (fireworks) in the editor to match the game. Thanks goes out to Footsteps_of_Coins for reporting this.
- made it so that if you attempt to go past the min/max level in the level editor, it won't bother with bringing up the save dialog. Thanks goes out to Erik557 for noticing this.
- saving level 0 or 100 to a different level number will no longer mark half the available secondary entrances as used if the "transfer secondary entrances" option was on.
- tweaked some code to workaround coordinate rounding issues in Microsoft's DPI virtualization code that can occasionally violate ClipCursor functionality.
- added "System DPI Aware" option to general options, to avoid DPI virtualization done by Windows Vista and later when possible.

Version 2.42 February 9, 2016

- fixed a crash bug from 2.40 that could occur under certain circumstances with the Background Layer 2 Editor. Thanks goes out to Luks for reporting this.
- added tooltips for the preview icons in the Add Objects/Sprites windows.
- added 2 forest tileset specific ledge edge subobjects to the "Add Objects" list that had been overlooked. Thanks

goes out to yoshifanatic for reporting this.
- increased the tile tracking limit per object to account for largest possible DM16 size. Thanks goes out to yoshifanatic for reporting this.
- enabled rest of SuperFX RAM remap from 2.41.

Version 2.41 December 25, 2015

- fixed sprites not triggering custom block ASM when touched from the side in layer 2 levels on layer 2. Apparently it's been this way for as long as custom blocks have existed (version 1.30). The fix will be installed when the Map16 is saved.
- added proper medium auto-scroll sprite F3 with extra bit of 1 to the sprite list and updated the tooltip about using other Y positions. Thanks goes out to imamelia for reporting this.
- added the ability to make custom Map16 tiles in the level editor appear translucent using the custom tooltips for Direct Map16 file. Check the help file for more info.
- added an option that will display a warning when using sprite 19 (display level message 1) in a level other than level 0xC5 when the fix for it has not yet been installed.
- added an option that allows toolbar buttons in the main level editor and overworld editor windows to wrap if there isn't enough room to display them on one row.
- added "All Objects" list and "All Sprites" list to the Add Objects and Add Sprites windows.
- added another area for SuperFX RAM remap.

Version 2.40 September 24, 2015 (15 year Anniversary of Lunar Magic!)

- added a new ASM hack to implement ExAnimation for the overworld, which is based on the same system as the one for levels. Check the help file for more details and a small list of differences from level ExAnimation.
- added a new ASM hack that expands the number of overworld exit path indexes to 0x80.
- made the previously unofficial support for the Super FX ROM map official, along with support for a future Super FX RAM remap patch.
- added a dialog to the overworld options menu that allows you to change the rate of animation the editor uses.
- made the bowser sign sprite in the overworld editor animated.
- made it possible for the overworld editor to display the lightning palette animation.
- added an option to make the overworld palette lightning animation use colors from the ROM instead of the working copy of the palette in the overworld's "Extra Options" dialog (frees up colors 9-F of palette 2 for maps that use lightning).
- added an option to disable player life exchange in the overworld's "Extra Options" dialog.
- added an option to disable player life exchange using L/R only in the overworld's "Extra Options" dialog.
- moved the keyboard shortcuts for loading/saving the credits in the overworld editor from Ctrl+F11/F12 to Shift+Alt+F11/F12 so that the same shortcuts used in the level editor for ExAnimation triggers can also be used in the overworld editor.

- added "Bring to Front" and "Send to Back" menu commands for objects/sprites in the level editor. You can also activate these with the mouse scroll wheel by holding shift while objects/sprites are selected.
- added "Bring Forward" and "Send Backward" menu commands for objects/sprites in the level editor as replacements for Increase/Decrease Z Order. These act much like the latter, but will skip past non-overlapping objects on the same screen so that you don't have to use the command as much.
- assigned keyboard shortcuts for the older "Increase Z Order" and "Decrease Z Order" to Ctrl+Alt+Shift+Plus/Minus, and removed them from the edit menu as they likely won't be used much anymore.
- added support for .palmask files. These provide a method for palette files to specify which colors in the palette should be imported into your level. Also added a couple new buttons to the palette editor to work with this. Check the palette editor section in the help file for more details.
- changed the Super GFX Bypass and Layer 3 Bypass dialogs to allow typing full values directly into the combo boxes for faster selection.
- added an option to the "Restore Point Options" dialog that scans the user area of the ROM for nested RATs on Full Restore Point creation. This can sometimes help detect data loss and corruption caused by third party tools and patches. Previously a user had to initiate a scan themselves if they wanted to check on this, which meant issues could go unnoticed for long periods of time.
- added an option to the "General Options" dialog to notify the user when RATS.log is written to. Also adjusted the information reported in the log file a bit.
- added a "Scan ROM" item to the file menu, which reports some information about the ROM user area. This

has actually been around for a very long time via a keyboard shortcut, but not everyone was aware of it.

- made it possible to display a custom image in the black non-editable areas of the level editor, background editor and overworld editor for cosmetic purposes. Check the help file for more info.
- added several more levels of zoom to all editors that have a zoom button.
- added an option to the "General Options" dialog that allows zoom buttons to either bring up the zoom menu or to act as a toggle between default zoom and the last used zoom setting (2x by default).
- made it possible to zoom in/out with the mouse wheel while holding down the control key. Also added an option to turn this off in the "General Options" dialog.
- moved the "Conditional Direct Map16 On" keyboard shortcut in the level editor from Ctrl+0 to Ctrl+5 so the former can be used as a zoom shortcut.
- added DPI scaling for the toolbar button images for when the minimum height adjusted for the screen's DPI is more than 25% larger than the image provided.
- added basic DPI scaling to the 8x8 editor, in 50% increments.
- improved mouse scroll wheel support. You can also now scroll horizontally with the wheel when there is no vertical scroll bar or by holding down the shift key.
- added the ability to fill an area of the overworld with a pattern of tiles in Layer 2 8x8 Editor Mode from either the selected area in the overworld (with Shift + Right Click) or from the 8x8 Overworld Tile Selector window (with Control + Shift + Right Click).
- made it so that left clicking a tile in Layer 1 Event Editor Mode in the overworld editor also selects the tile in the Layer 1 16x16 Tile Selector.
- made it so that left/right clicking in Layer 1 Event Editor Mode in the overworld editor will update the submap

palette selected in the palette editor and the submap palette/graphics used in the Layer 1 16x16 Tile Selector.
- made it so that left/right clicking in Layer 2 Event Editor Mode in the overworld editor will update the submap palette selected in the palette editor and the submap palette/graphics used in the Layer 2 Event Tile Selector.
- made the Layer 2 Event Tile Selector window for the overworld editor twice as high to show more tiles at once.
- moved the older Star/Pipe/Exit dialogs to a submenu.
- gave pitchin' chuck slightly different poses when displayed in the editor depending on how many balls he throws.
- added LM_NOTIFY_ON_CLOSE, LM_NOTIFY_ON_CLOSE_FORCE_ALL, LM_AUTORUN_ON_NEW_ROM, and LM_NO_AUTORUN options plus a %8 placeholder to get the current version of LM for user toolbars.
- removed window visibility checks on notification messages sent by LM to programs started from the user toolbar. Thanks goes out to HyperHacker for reporting this.
- fixed a display bug from 1.70 where ExAnimations that used the "Precision Timer Palette Rotate" trigger would be displayed by the editor at a wrong or uneven speed. The editor also now emulates the side effects of incorrect usage of this trigger.
- fixed a display bug from 2.30 where performing an undo in the overworld editor while in one of the event editor modes would not show undone changes in the layer 1 Map16 data or Layer 2 source event tiles until after you had switched out of that mode.
- fixed a bug from 2.30 where editing screen exits did not add an undo, yet could still be undone by a prior undo even though the editor would not update the display of

screen exits right away. Thanks goes out to Hinalyte for reporting this.

- fixed a bug from 2.30 where double clicks in the new BG editor didn't take into account scrolling. Thanks goes out to yoshifanatic for reporting this.
- changed the appearance of overworld path tiles 0xC and 0x13 (which are unused in the original game) to match land versions of path tiles 0x30 and 0x3A. Functionally they act the same way, but the 8x8 tiles Nintendo used suggests they were intended to be used as a visual guide for this. Also added an option to the view menu to display them the old way. Thanks goes out to Wiimeiser for pointing this out.
- made it so that the preview icons in the Add Objects and Add Sprites windows will update when certain view options are changed. Thanks goes out to Thomas for reporting this.
- slightly tweaked tile movement on right click pasting in the overworld editor for layer 1 event editor mode. Thanks goes out to Wiimeiser for bringing this up.
- added alternate GFX file info for sprites 33 (vertical fireball) and B6 (reflecting fireball). Thanks goes out to Footsteps_of_Coins for reporting this.
- added alternate GFX file info for sprite 1B (bouncing football). Thanks goes out to Katerpie for reporting this.
- corrected info and appearance for sprite D4 (bubble generator with goombas and bob-ombs), as it will also generate fish in bubbles after some time has passed. Thanks goes out to Thomas for reporting this.
- moved sprite 8E (warp hole) from "Tileset Specific" list to "Special Commands and Generators".

Version 2.32 January 1, 2015

- fixed a bug from 2.30 where ROM locking wasn't working correctly. Thanks goes out to S.R.H. for reporting this.
- fixed a minor issue that's been around forever with other windows being activated when closing windows from the toolbar. Thanks goes out to Ruberjig for pointing this out.

Version 2.31 October 31, 2014

- fixed a bug from 2.30 where the flying switch blocks on the overworld that appear when you pass a switch palace would temporarily corrupt part of the overworld's animated tile GFX in the game as the RAM they use overlapped with the extra space that the new 4bpp tiles take up. The fix will be installed when the GFX/ExGFX are inserted. Thanks goes out to 33953YoShI for reporting this.
- fixed a bug from 2.30 with the M16-7k dialog, where it wouldn't draw/behave correctly.
- fixed a minor issue from 2.30 where the layer 3 CGADSUB checkbox would not always be disabled when it should be. Thanks goes out to Hinalyte for reporting this.
- fixed a minor issue from 2.30 where if x2 zoom was already enabled in the Add Object/Sprite windows when the program was started, the list height was shorter than it should be. Thanks goes out to Underway for reporting this.

- fixed a minor issue from 2.30 where closing the Add Objects/Sprites windows would not let you use the level history mouse gestures without having to hold down the Shift key. Thanks goes out to Wiimeiser for reporting this.
- added the ability to set custom sprite appearances and tooltips in the editor based on the last 1, 2, 3, or 4 bits of the sprite's X and/or Y position. Check the help file for more info.
- expanded the Map16 space for custom sprites in the editor from 8 to 0x10 pages.
- tweaked the "Insert all Slots" button of the "Edit ExAnimated Frames" dialogs so that it will insert starting at the currently selected slot instead of always from the start.

Version 2.30 September 24, 2014 (14 year Anniversary of Lunar Magic!)

- made it possible for the level editor to display layer 3.
- made it possible for the overworld editor to edit layer 3 tilemaps for levels and submaps and store them in an ExGFX file (check the file menu).
- added a new ASM hack to allow layer 3 GFX and tilemap bypassing on a per level and per submap basis. Note that if you have already used a third party ASM hack for bypassing layer 3 GFX, it should be removed prior to using this version to avoid compatibility issues.
- added a new "Change Layer 3 Settings" dialog, which can modify layer 3 scrolling and some other advanced options. Also moved the Layer 3 Priority option from the "Change Properties in header" dialog and the Layer 3

Options setting from the "Change Other Properties" dialog into this new one.

- for anyone using only 3bpp tiles (which is probably nobody, but anyway): the ExGFX file range of E00-FFF has now been set aside for files to be inserted/extracted "as is" so they can be excluded from 3bpp/4bpp conversions. This range should be used only for layer 3 GFX and tilemap files, and for files intended for slot AN2 in the level editor. If you already have other files in that range, you should move them prior to using this version. But those that are using 4bpp tiles (likely everyone) can disregard this and continue using any file as they wish.
- added a new ASM hack for the overworld to allow setting the sprite GFX slots and the animated AN2 slot on a per submap basis. You can also now access all the ExGFX files for use on the overworld, not just the ones from 80-FF.
- added a new ASM hack for the overworld to reload the overworld (which includes the GFX) when swapping players that aren't on the same submap.
- added a new ASM hack for the overworld to reload all the GFX when using an exit tile. You must insert the GFX as 4bpp using this version then save the overworld for the hack to be enabled.
- added a new ASM hack for the overworld that allows merging GFX slots FG1 and FG2 into SP3 and SP4 to create room for 2 more FG slots. This option is off by default, but can be turned on in the "Extra Options" dialog of the overworld menu.
- added a new ASM hack for the overworld to remove the 3bpp limit on the overworld animated tiles, so they can now be 4bpp.
- added a new ASM hack for the overworld so that silent layer 2 event steps can appear below regular steps of higher event numbers. While Lunar Magic had always displayed it this way, apparently the original game

couldn't quite do it as it would place layer 2 silent steps above all regular steps once the overworld map had been refreshed. Thanks goes out to Falconpunch for drawing attention to this.

- made a couple minor optimizations to smkdan's VRAM patch.
- made a couple minor optimizations to LM's block gameplay ASM, which will be installed when you save the Map16.
- added undo and redo buttons to the level editor and overworld editor windows. Since this can affect the amount of RAM the program uses, you can change the max number of undo operations in the "General Options" dialog.
- replaced the "Add Objects" and "Add Sprites" windows with new ones that include a separate find text field and options for x2 zoom, displaying preview icons beside each item, hiding objects/sprites based on tileset/GFX files loaded, turning the preview area on/off, and horizontal/vertical layouts. The window can also now be resized both vertically and horizontally.
- made it possible to delete entries from the "Custom Collections of Sprites" list in the "Add Sprites" window, and to move entries by holding down shift and using the arrow keys.
- replaced the "Background Tile Map Editor" window with a new one that includes x2 zoom, undo/redo, and uses a regular window instead of the tool window in previous versions so it can be maximized/minimized as well as resized.
- moved the "Change Background Map16 Bank", "Remap Background Tiles", and "Copy Background Image" menu items from the level menu of the level editor to buttons in the new Background Editor window.
- added a new fill operation that can be used with control+shift right click to fill an enclosed area of the

level with Direct Map16 tiles from the Add Objects window. This can be useful for quickly filling in non-rectangular areas within ledges built with Direct Map16. You can also add the alt key to do it from the Map16 editor instead (which can be skipped if the Add Objects window is closed or isn't in the Direct Map16 Access list).

- made it possible to use control+alt right click in the level editor to copy FG tiles from the Map16 editor. You can also just use a right click if the Map16 editor is open and the Add Objects window is closed.
- made it possible to copy FG tiles from the Map16 editor into the level editor through the windows clipboard.
- made it possible to copy tiles from the Background Editor into the Map16 editor through the windows clipboard.
- made it possible to copy tiles between the Background Editor and the overworld editor through the windows clipboard when the latter is in a compatible mode that supports it.
- made it possible to copy tiles between the Map16 editor and the overworld editor through the windows clipboard when both are in compatible modes that support it. This was already possible if the Map16 editor was in 8x8 mode, but it has now been expanded to include 16x16 mode.
- added windows clipboard support to the layer 1 editing mode of the overworld editor.
- added "Auto-Deselect on Editor Select" options to the main level editor, Background Editor, and Map16 Editor.
- made it so that double clicking a tile in the Background Editor to select it in the Map16 editor will also open the Map16 editor if it isn't already open.
- added a new "Edit Reveal Tile List" menu item to the overworld editor, which allows changing which layer 1 tiles are revealed into another tile when you pass an

event. This was used in DW:TLC for creating red star levels.

- marked a few "Mario Path" tiles in the overworld editor with a black X, to indicate that Mario cannot enter them as levels even though he can stop there.
- fixed a display issue in the overworld editor that's been around forever where all sprites would always be shown with the palette of the main map without special world passed.
- made it so that loading the title screen in the overworld editor will use the layer 3 GFX files of the title screen level.
- modified the overworld editor's "Edit Level Names" dialog to use x2 zoomed tiles, DPI scaling, and added a new control for selecting which submap Layer 3 GFX and colors to use to display the preview.
- modified the overworld editor's "Edit Message Box Text" dialog to use x2 zoomed tiles, DPI scaling, and added a new option for using the Layer 3 GFX and colors of the currently loaded level in the level editor to display the preview.
- modified the overworld editor's "Edit Boss Sequence Text" dialog to use x2 zoomed tiles, DPI scaling, and to use the correct colors.
- added the ability to export a level to a PNG image file.
- made it possible to use Snes9x savestates for recording title screen moves.
- changed the "Export Title Moves Playback Data" menu command to create a fake ZSNES savestate rather than having it require an existing savestate to save into.
- added the ability for programs launched from LM to make requests to LM to reload the current level or ROM. Check the help file for more info.
- fixed an issue from 2.20 where saving to a different level number would not update the displayed secondary entrance sprites in the currently loaded level to reflect

possible changes. Thanks goes out to Hinalyte for reporting this.

- adjusted the auto-set number of screens feature so it ignores entrances, as it did before 2.20.
- adjusted the exit scan option so that for "clear the original level data area" and "import multiple levels from file" operations, it will no longer give a warning for screen exits that are explicitly set to level 0/100. Thanks goes out to Wiimeiser for bringing this up.
- made it possible to change the Mario sprite tile arrangement displayed in LM for level entrances, using the same method as with custom sprites. Check the help file for more info.
- added the mouse gestures for back/forward to the list of function commands for custom user toolbars.
- fixed a couple issues that have been around forever with trying to use the "Copy BG" function with the TEST levels. Thanks goes out to Sokobansolver for bringing this up.
- fixed a bug that's probably been around since 1.70, where if LM ever tried to save a BG to a particular address the save would be aborted and a warning message saying "Could not save BG data in a bank-appropriate area" would be displayed. Thanks goes out to Mini-Tech for helping to identify this issue.

Version 2.22 February 9, 2014

- fixed a bug from 1.70 in smkdan's VRAM patch where sprite 49 (growing/shrinking pipe end) could cause a few pixels of some tiles to be corrupted if the sprite was near enough to the entrance to be processed during fade in. Thanks goes out to WhiteYoshiEgg for reporting this.

- fixed a potential bug that's been around forever where if your overworld had no silent events at all and was not yet using the new format, LM would create a single entry for the table as required by the game's original ASM, but the contents weren't specified so it could end up holding something invalid/harmful. Note that while LM now sets it to "no event" and recognizes that as valid, the entry created by previous versions still has a chance to trigger the new "silent event corruption" warning added in 2.21. In that case simply resaving the overworld in this version will remove the offending entry. Thanks goes out to Wiimeiser for reporting about this.
- fixed an issue from 2.20 where opening a level from the "Open Level Address" menu would result in the main and midway entrance not being displayed although the labels still were.
- corrected the "Mario Path" displayed for overworld layer 1 tile 0x36, which is actually a water version of tile 0x1C. Thanks goes out to Wiimeiser and mariofreak4500 for drawing attention to this.
- increased the allowed number of sprites for the sprite count warning to 255 if Vitor Vilela's RAM remap for SA-1 ROMs has been used.
- added some code to prevent attempts at opening a second overworld window under certain conditions. Thanks goes out to Hinalyte for reporting this.
- made a minor tweak to the optimized LZ2 and LZ3 ASM code to not attempt decompression if the GFX file has not been inserted, to aid with ASM debugging. Will only take effect once you switch compression formats with this version.
- moved most of the options in the options menu to their own "General Options" dialog.

- removed the "Highlight Mouse Cursor in BG Editor" option, as it doesn't seem likely that many people turn it off anyway.
- added a new "Check if Vertical Fireball has Buoyancy" option to the general options dialog, at Alcaro's suggestion.

Version 2.21 December 25, 2013

- fixed a bug from 2.20 where the new "view sprite data" menu item in the overworld editor would display black squares for the sprite data text if the editor had not yet loaded an overworld that wasn't using a custom palette. Thanks goes out to Wiimeiser for helping to figure this out.
- fixed a bug from 1.40 (which became more noticeable in 2.10) where changes to the last 8 entries in LM's "Destroy Level Tile Settings" dialog could cause Ghost House/Castle/etc tiles to no longer give the player a save prompt when the event had already been cleared. This is partly due to a mistake in Nintendo's code, as it treats the list as 0x18 bytes long when it's really only 0x10 bytes. This version of LM relocates the table so it can safely have all 0x18 entries, and repairs the other 8 bytes that may have been changed in previous versions. Thanks goes out to Wiimeiser for helping to figure this out.
- fixed a minor entrance label display positioning issue from 2.20 when switching between vertical/horizontal level layouts. Thanks goes out to Koopster for pointing this out.

- fixed a small bug that's been around forever, where if the save prompt for castle/ghost house/etc tiles was disabled and the player passed a level using one of those tiles, the player would move off the level even if the event had already been passed.
- added a new ASM hack to fix a timing issue in Nintendo's code, where there was a chance that the game could turn the screen on a frame too early during black screen transitions, resulting in a brief full screen flash of color.
- disabled some Nintendo ASM that apparently served no purpose, which was preventing ExGFX from being used in tiles 4A-4F and 5A-5F of SP1. This also now makes these tiles 4bpp.
- added some filtering and warning messages to detect and remove bad event data in the overworld in case people corrupt it with external tools/patching.
- made it so that pressing Page-Up in the overworld editor in event editor mode while on event 0x77 will cause the event to be shown as passed, so you don't have to advance through each step with Home/End for that event. Thanks goes out to Everest for bringing this up.
- added a small bypass warning in the old GFX bypass dialogs and the tileset change dialog when Super GFX Bypass is enabled, at Alcaro's suggestion.

Version 2.20 September 24, 2013 (13 year Anniversary of Lunar Magic!)

- made the previously unofficial support for the SA-1 ROM map official, and added 6 & 8 MB expansion menu items for SA-1 ROMs (see the help file on emulator compatibility with large SA-1 ROMs).

- added support for Vitor Vilela's RAM remap for SA-1 ROMs.
- made it possible to drag/copy/delete/alt-right click level entrances as part of sprite editing mode. Dragging an entrance automatically updates its FG/BG init position (except for the BG init position in vertical levels), copying an entrance creates a new secondary entrance in a free slot, deleting a secondary entrance clears the slot, and alt-right click lets you modify the entrance properties and change which slot a secondary entrance should use.
- changed the "Add/Modify/Delete Screen Exits" dialog so that changes to the current entry aren't lost when you switch which screen you want to edit, and renamed it to "Modify Screen Exits".
- added a "Clear All Screens" button to the "Modify Screen Exits" dialog and renamed the "Delete" button to "Clear Screen".
- the "Modify Screen Exits" dialog now displays information about exits that have been set for each screen.
- changed the definition of a "free" secondary exit slot to be more than just an exit that points to level 0 or 100. If you have secondary entrance slots that you manually cleared in the past by only changing the level destination, you will need to use the "Clear Slot" button or delete them so they'll show up as unused again.
- level entrances are now displayed using the player graphics from the game.
- replaced the mostly redundant "Screen Boundaries" item in the view menu with a new "Game View Screen" option. This creates a draggable SNES-sized screen that you can use when designing levels to figure out what will be visible on the screen. It also optionally has

highlighted areas for locations Mario will be during scrolling in different directions, to aid in determining where the screen will be during gameplay.

- made it possible to have Lunar Magic load external files in place of GFX32.bin and GFX33.bin for display purposes in the editor. Just place files called ExternalGFX32.bin and ExternalGFX33.bin in the Graphics folder for your ROM.
- added a new ASM hack where you can set the midway level entrance to coordinates separate from the main level entrance.
- added a new ASM hack where a screen exit can go to a level's midway entrance if it's using the new separate coordinates option.
- fixed an issue in Nintendo's original code, where if the midway entrance was set to screen 0 the game would not record obtaining the midway point for that level.
- adjusted the control+delete keyboard shortcut to also delete all secondary entrances in the current level.
- added the ability to set the Koopa Kid teleport locations in the overworld editor, and a menu item to view them (check the help file for more info on the Koopa Kid sprites). Thanks goes out to Wiimeiser for providing the test patch to examine these.
- added the ability to view sprite data to the overworld's view menu, as it helps in showing which slot and subslot a sprite on the overworld belongs to.
- set LM to pause the internal emulator if viewing animations is turned off. Thanks goes out to MarioFanGamer659 for reporting on this.
- added tile/page jump in place of text search for the Direct Map16 list in the Add Objects window. Thanks goes out to telinc1 for bringing this up.
- made conditional Direct Map16 objects appear translucent in the editor if the "Other Invisible Objects" view menu option is on and "Conditional Direct Map16

On" option is off. Translucency will not be applied to objects using the +100 option.
- adjusted tiles used for sprite 4D (Monty Mole) in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to Wiimeiser for pointing this out.
- changed the name of the "Cloud Ledge" for ghost houses to remove the "Ledge" bit as you can't stand on it. Thanks goes out to Wiimeiser for pointing this out.

Version 2.12 February 16, 2013

- fixed a bug from 2.11 where using the "Edit Boss Sequence Text", "Edit Message Box Text", or "Edit Level Names" dialogs in the overworld editor then saving the overworld would change the list of GFX files to load for the main overworld map. Thanks goes out to Wiimeiser for reporting on this.
- added a save prompt for Windows logoff/shutdown events. Thanks goes out to Vitor Vilela for bringing this up.
- removed the "Allow Fragmentation" option from the options menu, as it's a very advanced option that's been sitting there since version 1.00 but no one is really going to want to turn it off anyway.
- removed the "Level Insertion Complete!" message box that pops up after using the "Save level to ROM as..." command, as it's another holdover from version 1.00 and not really necessary.

Version 2.11 February 9, 2013

- fixed an issue in the new overworld 8x8 tile selector, where it wouldn't show the correct colors for title screen or credits editing. Thanks goes out to Zeldara109 for pointing this out.
- fixed an issue in the new overworld 8x8 tile selector, where the overworld controls did not enable the save button when used. Thanks goes out to Zeldara109 for pointing this out.
- made an adjustment to the new overworld 8x8 tile selector, where it will paste tiles with priority enabled by default for title screen and credits editing, at Zeldara109's suggestion.
- changed the "Modify Secondary Entrances" dialog, so that changes to the current entry aren't lost when you switch which entrance you want to edit.
- added a "Clear Slot" button and a "Clear All Slots" button to the "Modify Secondary Entrances" dialog.
- added a new option when importing multiple levels from files, to clear all existing secondary entrances in the ROM before importing.

Version 2.10 December 25, 2012

- added x2 zoom to both the level editor and the overworld editor.
- added a new 8x8 tile selector to the overworld window. The new selector allows multiple tile selecting, is x2 zoomed by default, and has controls similar to the level editor's new map 16 editor for manipulating the current

selection in the main overworld window (including a remap button).

- added a new "Auto-Deselect on Editor Select" option to the overworld's option menu, which causes tiles in the main overworld editor to be deselected automatically if you make a selection in one of the selector windows.
- removed the "Modify Selected 8x8 Tiles" menu item from the overworld menu, as the new 8x8 tile selector makes it redundant.
- removed the "Exit Path Tile Settings" and the "Star and Pipe Tile Settings" buttons from the toolbar, to discourage their use in linking tiles in favour of the alt-left click or link button methods. They can however still be found in the overworld menu.
- added the ability to create one-way links when using alt-left clicks to link 2 star/pipe/exit tiles together.
- added "Show Star/Pipe Numbers" and "Show Exit Path Numbers" to the overworld's view menu.
- added a warning message when trying to link Star and Pipe tiles placed at X=0 with Y>=0x20, as they don't function correctly from there due to a limitation with one of SMW's tables.
- fixed a bug in the overworld editor that's been around forever, where if you placed an exit tile that goes left/right at X=0 with Y>=0x20 then moved it, the tile would partially unlink from its destination tile and would have to be relinked.
- fixed a bug in the overworld editor that's been around forever, where if you changed certain settings of a selected tile then moved that tile without first deselecting it, the tile would either revert to its previous setting or not move the setting to the new location. The settings affected by this were the Star/Pipe/Exit tile settings, the "Reveal this level tile on any of these events" setting, and the "Destroy Castle/Fortress/Switch Palace tile" setting.

- fixed a bug in the overworld editor that's been around forever, where if Star/Pipe/Exit tiles were selected then you copied or inserted new Star/Pipe/Exit tiles without first deselecting the old ones, the settings of the previously selected Star/Pipe/Exit tiles were vulnerable to getting attached to the new Star/Pipe/Exit tiles if they happened to get dragged over the old ones.
- fixed a bug in the overworld editor that's been around forever, where if a level tile was placed in the main map at the same layer 1 VRAM location as another tile placed in the submap area, the "Reveal this level tile on any of these events" setting would be changed for both tiles whenever you tried to change one of them. Thanks goes out to Gamma V for discovering this.
- fixed a display bug in the overworld editor that's been around forever, where the overworld animation tiles in the second half of GFX file 0x14 were supposed to be shown using only the first 8 colors in a palette but LM wasn't enforcing it. Thanks goes out to Shurik Kid for reporting this.
- fixed a bug from 1.80 where if someone who was keeping all their GFX at 3bpp inserted the GFX, the overworld 4bpp modifications for FG3/4 might still be applied, causing overworld layer 1 tiles to use the wrong colors.
- you can now clear the entry for a Star/Pipe/Exit location by deleting the tile at the location it's currently assigned to (for exit locations, there must be an actual exit tile there).
- increased the animation rate in the overworld editor to 15 fps, to better match the game.
- added support for LMSW's SwitchFlags function, to allow viewing state changes to the switch palace blocks in the emulator.
- upgraded the level editor's "Add offset to Background Tiles" dialog to a full remap dialog.

- made some adjustments so that the restore system can deal with ROM files marked as read-only/hidden on its own when possible during a restore operation.

Version 2.01 May 20, 2012

- fixed a bug from 2.00 where if the LMSW DLL requested a level switch there was a chance LM could crash. Thanks goes out to TRS for discovering this.
- set LM to automatically leave sprite editing mode when starting the internal emulator.
- fixed a bug from 2.00 where the LM_NEWIMAGE directive for user toolbars wouldn't reset the base index right away like it should. Thanks goes out to jimbo1qaz for discovering this.
- fixed an odd bug from 1.00 in the "Add Sprites" window where if a sprite's tiles in that window were placed at certain locations and in a certain order, LM would refuse to paste the sprite into the level. While in LM's sprite lists this was only evident in version 2.00 with sprite 66 (upside down chainsaw), user supplied sprites in the "Custom Collections" list could trigger it in previous versions as well. Thanks goes out to Wiimeiser for reporting this.
- fixed a small display issue from 1.00 where the back area color was dimmed in certain level modes. Thanks goes out to Epsilon for reporting this.
- corrected the descriptions for sprites 41 and 42 (jumping dolphins), as they were switched around. Also changed the direction they face. Thanks goes out to Zeldara109 for reporting this.

Version 2.00 May 5, 2012

- added the ability to load the LMSW DLL from Alcaro, which allows using an emulator to play levels within LM itself (downloaded separately).
- added an optional user defined second toolbar to the main window, which can be used to launch external applications or activate internal Lunar Magic functions. It also allows for assigning keyboard shortcuts which can replace Lunar Magic's existing keyboard shortcuts. Check the technical info section of the help file for more details.
- fixed a bug from 1.00 where LM didn't preserve the Z order of sprites when possible and only cared about following the sorting rules imposed by the game. Thanks goes out to King Dedede for noticing this.
- fixed a bug from 1.00 where changing a level from having objects on layer 2 to an image would cause LM to draw all objects that were on layer 2 on layer 1 instead until the level was reloaded.
- fixed a small bug from 1.91 where if you opened the Add Objects/Sprites windows then closed and reopened them again, the last used currently selected item could not be pasted (and would not be displayed in the preview area for sprites) until the selection was changed. Thanks goes out to Dakress for reporting this.
- fixed a bug from 1.90 in the new Map16 editor's Remap button dialog and the new Remap DM16 dialog where if you just entered a space for the list, LM would freeze. Thanks goes out to DiscoMan for reporting this.
- fixed a bug from 1.70, where a check for JW's old animated tile editor had a very rare chance of being

triggered by LM's own ExAnimation ASM, causing junk to be displayed in LM for the game's original animations. Support for this tool has been disabled, as it hijacks the same addresses as LM anyway.

- fixed a minor issue from 1.70 where if you forgot to enter the number of colors for a new palette animation when editing ExAnimation frames and left it blank, LM would use a value of 1 but would interpret the entered frame values as RAM offsets instead of SNES RGB values. Thanks goes out to supernova38 for discovering this.
- further tweaked some code so that the new double clicks introduced in LM 1.90 also count as regular clicks for the middle mouse button. Thanks goes out to Alcaro for reminding me of this.
- renamed the "Save level to ROM" menu item to "Save Level to ROM as...", then added a new "Save level to ROM" menu item that works the same way as the save button on the toolbar and assigned it the Control+S shortcut. "Save Level to File" has been moved to Control+W instead of Control+S.
- reduced the maximum number of screens LM displays in levels that use layer 3 tides to reflect what's safe to use.
- adjusted tiles used for sprite 99 (Volcano Lotus) in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to MSA/Gamma_V for discovering this.
- adjusted the position of the light switch block, message block, and Banzai Bill sprites slightly in LM to match the game. Thanks goes out to Alcaro for noticing this.
- changed the palette used in the white blocks in the bonus game in LM to match the game. Thanks goes out to Alcaro for reporting this.
- adjusted the positions of sprites 65-68 (line guided chain saw, upside down chain saw, grinder, fuzz ball) in LM to more closely match the game. Also greatly

corrected their "set to go right" positions and updated their descriptions.
- changed the appearance of sprite 30 (dry bones that throws bones) to actually show it with a bone ready to throw, to better distinguish it from sprite 32.
- horizontally flipped the koopa shells in LM so they'll appear the same way in the editor as they do in the game.
- corrected the tooltip for sprite 3C (wall following Urchin) as the directions were reversed. Thanks goes out to King Dedede for pointing this out.
- tweaked several other tooltips and sprites here and there.
- added some menu items to the view menu for custom ExAnimation triggers, to allow previewing them in LM.
- added displaying the size of a level in bytes within the status bar when a level is saved.
- added displaying the size of GFX/ExGFX in bytes within the status bar when GFX/ExGFX are inserted.
- added the ability to customize button images for the new Map16 editor (using a bitmap file named "Lunar Magic.ff5"), and the palette editors ("Lunar Magic.ff3" for the level palette editor and "Lunar Magic.ff6" for the overworld palette editor).
- adjusted M16-7k a bit, as it's been difficult to get into since 1.70 and became mostly unusable in 1.90. Thanks goes out to MaZ18 for reporting this.
- did a few minor tweaks to the object/sprite window searching at Zeldara109 and Alcaro's suggestions.

Version 1.91 December 25, 2011

- fixed a bug from 1.70 in smkdan's VRAM patch where dynamic changes (such as coin collection) on layer 2 were not always displayed if layer 1 and 2 scrolling were not in sync, and another bug where a column of garbage tiles could appear under rare circumstances. Thanks goes out to smkdan for submitting the fixes for his patch, and Zeldara109 for reporting the coin issue.
- added a new "Prefer Saving in 2MB+ ROM area" option to the options menu, which is on by default. This will cause Lunar Magic to first try saving data and code past the 2MB mark in the ROM before using earlier space, but only if the ROM is already larger than 2MB. This will help save space below bank 40 for less versatile 3rd party ASM patches/tools to use, as many of them unfortunately do not take bank 40+ into account and will malfunction without warning when installed there.
- added a new "Check Object Placement on Save to ROM" option to the options menu, which will warn if an object is placing tiles beyond the viewable level area into unsafe SNES RAM locations which could cause various issues during gameplay.
- added a new "Export Title Moves Playback Data" menu command to the overworld editor, to allow exporting existing title screen playback data from a ROM back into a Zsnes savestate, so it can be transferred to a different ROM.
- added a new "Edit Manual" command to the edit menu, which allows manually modifying existing objects/sprites. You can also Alt-Right click on an object or sprite to use this.
- added the ability to remap 16x16 gameplay "act as" settings in the new Map16 editor using the remap button.
- added a warning message to be displayed when attempting to use Lunar Magic on a ROM that has already been edited by a newer version of the program.

- saved the new Map16 editor's zoom setting to the registry.
- enabled the new Map16 editor to use the "translucent text and outlines" option from the main editor, but set the translucency level a bit lower than what's used in the main editor.
- enabled using Ctrl+Z/Ctrl+Y keyboard shortcuts in the new Map16 editor for undo/redo.
- tweaked some code so that the new double clicks introduced in LM 1.90 also count as regular clicks.
- added text search to the Add Sprite/Object windows, which you can activate just by typing.
- added support for non-256 color bitmaps when using custom toolbar images, and support for using images larger than 16x16 for the toolbar buttons.
- added sprites A0, A1 and A9 (Bowser Scene, Bowser's Bowling Ball, Reznor) to sprite list.
- added a few alternate sprite GFX file suggestions to tileset specific sprites in Add Sprites window.
- adjusted sprite 33 (Vertical Fireball) in LM to include some bits of fire dropping off it, at Zeldara109's suggestion.
- horizontally flipped sprite 2C (Yoshi Egg) in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to Wiimeiser for discovering this.
- adjusted palette used for sprite 45 (Directional Coin) in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to andy_k_250 for reporting this.

Version 1.90 September 24, 2011 (11 year Anniversary of Lunar Magic!)

- replaced the Map16 editor with a new one that features multiple tile selection, reload/save and undo/redo buttons, 2x zoom, 8x8 and palette remapping capabilities, an 8x8 mode with an 8x8 tile selector, etc. This release also introduces a new .map16 file format intended to replace the old generic .bin files that were used in previous versions for exporting/importing single pages of Map16 data. The new files can hold variable amounts of tiles, can optionally be imported to the same location they were exported from, and multiple files can be imported at once. There are also 2 new buttons allowing you to export/import all the Map16 data for all tilesets from/to a single file. LM remains backward compatible with both loading and saving the older raw .bin Map16 data files.
- added a feature to double left click on a tile in the level or background editors to select it in the Map16 editor, and you can also double left click on a tile in the Map16 editor to select it in the 8x8 editor.
- removed being able to paste the current page from the Map16 editor into the background editor with Ctrl+Shift+Insert, as the new Map16 editor makes it unnecessary.
- added a new "Add Offset to Background Tiles" menu item to the level menu, which allows adding or subtracting an offset from all tiles in the background, much like F9 does to selected tiles in the background editor. Also made this function able to change the BG Map16 bank setting automatically when needed.
- added a new "Remap Direct Map16" menu item to the edit menu, which can remap specified Direct Map16 objects in a level to others in case you've moved the original Map16 tiles to a different location.
- made it possible to set Luigi's starting position on the overworld to a different location than Mario's.

- added an option to the overworld editor that allows disabling the original game's path fade effect, which frees up colors 1-7 of palettes 0-3 and C-F, and GFX slots SP3 + SP4. Also added a setting to control the path revealing speed when this option is used.
- expanded the overworld's layer 1 Map16 data to 2 full pages. Tiles on the second page will generally act like tiles on the first page.
- included an ASM hack from DW:TLC that expands the number of overworld pipe and star indexes to 0x80.
- updated the overworld editor so that if the tile that replaces the top of a castle when destroyed is edited, the game will correctly display the edited tile in the destruction sequence.
- added an entry to the options menu to allow using an alternate GFX Bypass dialog that lets you type in the GFX files to use instead of selecting from a list. Previously this option could only be changed by pressing Shift-F4.
- added using Control-Alt-Left/Right click in the palette editor for copy/pasting an entire row.
- added a "Paste all Slots" button to the Exanimated Frames dialog which pastes all copied slots to the same slots they came from, and moved the "Insert all Slots" button to the side.
- made it possible to use text labels for custom sprite tile arrangements, which appear the same way that LM shows certain sprite commands. Check the help file for more details.
- added a prompt to create an IPS patch after locking a hack.
- changed some code so that applying an IPS patch of a locked hack to a ROM will cause LM to close the ROM after patching instead of closing the program. But for convenience you can still use F4 to run the previously

open ROM in your emulator. Thanks goes out to GlitchMr for bringing this up.

- made a slight adjustment to sprite 61 in LM so it will appear the same way in the editor as it will in the game. Thanks goes out to GlitchMr for pointing this out.
- made the edit controls in the Exanimated Frames dialog slightly larger to allow entering all digits for direct offsets. Thanks goes out to imamelia for reporting this.
- fixed a bug from 1.80 where having auxiliary .msc .dsc etc files with a size of 0 bytes would cause LM to crash if creating a full restore point. Thanks goes out to iRhyiku for submitting the data that helped figure this out.
- fixed a bug from 1.80 where the restore point created by locking a hack was invalid.
- fixed a bug from 1.70 where opening an mwl file with an unmodified background could sometimes cause LM to display the background using the wrong BG Map16 bank (although if saved to the ROM, the correct setting would still be preserved and used when the level was reloaded). Thanks goes out to Ultimaximus for reporting this.
- fixed a bug from 1.64 where locking a hack prevented LM from opening any other ROMs and had to be closed.
- fixed a bug from 1.50, where if the gameplay table in the ROM was corrupted for any reason (like the user overwriting it with a patch), Lunar Magic could crash while moving the mouse cursor over a block that referenced the invalid data. Also fixed a similar bug from 1.30 in the old Map16 editor where it could crash while right-clicking to copy a tile with the invalid data.
- fixed a minor display issue from 1.20, where edits to the Map16 tiles that make up a pipe would not be displayed in the main level editor until you either saved the changes to the ROM or changed the pipe view index.
- changed window border style setting for main and overworld editor as apparently Win7 doesn't handle that

border type on application windows when Aero is enabled. This had resulted in weirdness from Aero like the caption/title bar of the window being partially offscreen when maximized, and the inner window border occasionally appearing as unerased pixels.

Version 1.82 October 31, 2010

- fixed a bug from 1.81 that causes the game to glitch/crash midway through the enemy list at the end of the game.
- fixed a bug from 1.81 for SMAS+W involving the Direct Map16 Access ASM.
- fixed an issue from 1.80 for SMAS+W, where the restore feature was using the wrong CRC value to check for an original unmodified ROM. Thanks goes out to GlitchMr for reporting this.
- the above made me realize that I've apparently been using an overdumped SMAS+W ROM this whole time (3MB instead of 2.5MB). This means Lunar Magic hadn't been coded to recognize the extra 500KB as free space. But it was installing some fixed location ASM hacks and tables there, so to avoid compatibility issues this version of LM will only recognize that area as free space when starting a new hack.
- fixed a bug that's been around forever where turning off the "Correct Fatal Errors" option and inserting an object that would normally be interpreted as an error would cause object resizing with the mouse to use single-pixel increments, making it rather difficult to control. Thanks goes out to Imamelia for discovering this.

- included a palette fix for the "S" in "Mario/Luigi Starts", as it was wrong in the original game.
- made it possible to resize the Add Objects/Sprites windows vertically, at Zeldara109's suggestion.
- added compression options to the options menu, which allows switching to either the optimized for speed version of LC_LZ2, or to LC_LZ3 for better compression using a patch submitted by edit1754.
- added an option to the options menu to disable converting the sprite berry GFX tile. Technically this option has been around since version 1.60, but the only way to change it until now has been through the registry since it was never added to the GUI.

Version 1.81 September 24, 2010

- fixed an issue from the 1.80 release where it wasn't actually installing the required ASM for the Direct Map16 Access upgrade (errr... oops. Always ends up being one of those last minute/week changes that trip you up.)

Version 1.80 September 24, 2010 (10 year Anniversary of Lunar Magic!!)

- fixed a bug from 1.70 where the Iggy/Larry battles could glitch depending on certain conditions. Thanks goes out to Markus for reporting this.
- added a new restore feature to the program that can optionally track changes to the ROM and allow the user

to revert the ROM to any previous restore point. Note that this requires the program have access to a copy of the original ROM, so you may have to browse to a copy the first time you save in the new version. Also added create/apply IPS to the restore menu.

- the Direct Map16 Access feature has been upgraded so you can select an area with multiple tiles in the "Add Objects" window and paste them as a single object. Resizing the object will repeat the pattern of the original tiles pasted. It can also now be resized to larger areas than before, and has been made vertical level compatible so it won't break up like regular objects do. All of which can help reduce level data size.
- added a Conditional Direct Map16 Access feature to the edit menu, which allows you to set any DM16 object to only be rendered if a certain flag is enabled within a table of values in RAM. Check the help file for more information.
- added Ersanio's FastROM modifications as an option in the options menu.
- made several adjustments to the palette editor (moved it to the editors menu, moved the "enable custom palettes" option into it, added undo and redo buttons, etc).
- made the same adjustments to the overworld palette editor that were made to the level palette editor.
- added an option to the overworld palette editor to allow custom palettes for the overworld.
- cleaned up the shared palettes for the overworld and levels by removing the bits left over after transitioning between the two as they generally can't be relied upon to always be the same due to the use of custom palettes.
- adjusted overworld to use full 4bpp graphics for FG3 and FG4. To fully enable, you must extract then insert the original GFX. Note: before you start playing with

overworld custom palettes and graphics, be sure to read the warnings about path revealing in the help file sections for the palette editor and the submap FG graphics dialog.

- enabled the "extension" field in the "Add Sprites Manual" dialog to allow multibyte sprites with 3rd party utilities. Check the help file for more information.
- added an "extension" field in the "Add Objects Manual" dialog to allow entering more information for multibyte objects. More specifically, it's for object 2D which has now been set aside as a 5 byte object for user defined purposes (do not use this object unless you've inserted code into the game for it, or in-game level parsing errors will occur).
- added viewing tile grid in level and overworld editors with F8 (use control+alt+F8 to switch grid color).
- moved existing F8 functionality (disable title screen color hack) to control+shift+F8.
- added a option to display a warning when an ips with the same name as the ROM is detected in the same directory.
- reorganized the toolbar buttons a bit.
- updated sprite A5 so it displays in LM the way it does in the game when used in sprite tilesets it was not originally intended for. Thanks goes out to Alcaro for reporting this.
- corrected sprite 7F to use the correct palette in LM. Thanks goes out to imamelia for pointing this out.
- corrected part of sprite 9B to use the correct palette in LM. Thanks goes out to EL santo for finding this.
- mostly fixed the delay that would occur when bringing up the Super GFX dialog on Win7/Vista.
- added code to automatically remove the block that IE7+ puts on the help file when you use that browser to download the program. Also added a warning to be displayed when the help file cannot be found.

- updated the program to stop always using the Win95 default font. Should improve readability on LCDs with ClearType.

Version 1.71 April 17, 2010

- fixed a bug from 1.70 where attempting to initialize any of Manual Frame Triggers 1-F would crash the game. Thanks goes out to Davros for noticing this.
- corrected a display glitch from 1.70 where the secret goal tape sprite was being displayed in LM as though it were a custom sprite.
- fixed a display issue from 1.70 where tile 0x25 was not highlighted when selected. Thanks goes out to Dotsarecool for reporting this.
- corrected an issue where attempting to save more sprites than a LoROM bank could hold would not display an error message like it was supposed to and LM would only save the sprite header. Thanks goes out to Glitch for reporting this.
- re-added setting 3 for the "Item Memory Index", along with an ASM hack that changes the setting so it doesn't record any items obtained. Thanks goes out to BMF for submitting the ASM for this.
- added a few ROM expansion menu commands to the file menu, in case someone wants to expand the ROM for data or code they intend to insert themselves.
- updated object 3B and extended objects 49,80,84, and 85 to break up when near or on screen boundaries like the game does. Also removed some junk subobjects from the list for object 3A. Thanks goes out to Alcaro for reporting this.

Version 1.70 April 1, 2010

- smkdan's VRAM modification patch has been integrated into LM. This means levels can now use 50% more 8x8 FG/BG tiles (0x100 tiles), which can be accessed using 2 extra GFX slots (BG2 and BG3). A big thanks goes out to smkdan for developing and submitting this improvement, and to Vic Rattlehead for testing it for him.
- Map16 space has been expanded to 4x the previous limit. There are now 0x40 pages for FG Map16, and 0x40 pages for BG Map16.
- Map16 page 2 can now be set to be either tileset specific or not, with Control + F1 in 16x16 editor window. New hacks will have this set to off by default.
- the page limit for backgrounds has been replaced with a bank limit. A single background can now access up to a bank's worth of Map16 tiles (0x10 pages).
- backgrounds have been upgraded to use a height of 512 pixels instead of the 432 that Nintendo went with, which translates to 5 extra rows of 16x16 tiles. While this doesn't make much difference in horizontal levels, it means vertical levels can now have backgrounds that will seamlessly wrap vertically. It also effectively means the end of the "animated tile garbage" that used to appear when the background was vertically scrolled past the viewable range.
- an option was added to the view menu for viewing backgrounds that are 512 pixels high.
- the ExAnimation feature has been completely rewritten. The changes include: addition of several new 8x8 line types and new palette types (such as Palette Rotate),

palettes can have multiple colors per entry by using ExGFX files for the source data, multiple new triggers for use in custom blocks (such as custom/manual/one-shot), the ability to use a few non-compressed ExGFX files for source tiles, the number of frames can now be set up to a limit of 0x100 frames per animation, the addition of a global animation list for animations to be run in all levels, and settings that can disable both the original game's tile animation and color 64 palette animation as well as Lunar Magic's own level and global animations on a per-level basis. ExAnimation now also spreads entries across multiple frames instead of doing them all at once every 8th frame, to improve the amount of vblank time available for animations. *Note that this may have implications for custom blocks programmed to change behavior based on LM's ExAnimated frame counter at $7FC004. While the counter is being maintained for backwards compatibility, it's only truly accurate now for slots 0,8,10 and 18 (although if you have a block that hurts mario, it seems the next 2 slots after those are still close enough). If you find this is an issue you can either move your animations to the mentioned slots, or recode the blocks to use the new animation per-slot counters starting at $7FC080 (add 0x20 for global entries).

- the 2 8x8s stacked ExAnimation type has been changed slightly, so that the source tiles must now be one after the other (previously the two tiles had to be separated by a single tile, which was inconsistent with all other line types).
- added a dialog to the options menu that allows you to change the rate of animation Lunar Magic uses. I've also bumped up the default rate to 15 fps. The faster settings are only really useful if you're using the ExAnimation speed tricks.

- added an option that makes LM use FastROM addressing for its own data and ASM hacks, in case you want to convert your ROM to use FastROM.
- added an ASM fix for fading colors above color 7 in each FG/BG palette at the end of a level. Nintendo didn't bother with this as they didn't use those colors much. Colors related to the status bar (colors 9,A,B of palette 0 and colors

  9,A,B,D,E,F of palette 1) will not be affected. If BMF's version of the fix is already installed, LM's won't be inserted.

- modified the original graphics insertion code to disallow saving within the unexpanded ROM area unless it's within the original GFX area.
- added a "Clear Original Level Data Area" menu command to the file menu, which will resave all non-modified levels except the test level then clear most of the data in the original level data area to free it up for 3rd party ASM hacks.
- added a dialog that lets you add a specified value to all selected 16x16 background tiles with F9.
- added a way to paste the current page in the Map16 editor into the background editor with Ctrl+Shift+Insert.
- the "custom sprites" and "translucent text and outlines" options were moved from the "options" menu to the "view" menu.
- improved "translucent text and outlines" drawing performance.
- stopped drawing the unused secondary entrances in levels 0 and 100, as there were a couple hundred of them overlapped and drawing them tended to slow down older computers.
- made it possible to customize sprite appearances and tooltips for sprites that have the extra bits set to 1.

- made it possible to select sprite indexes B-FF in the overworld editor, at Roy's request.
- fixed the overworld editor where if animation was turned off and event or level number labels were turned on and you moved around level tiles, it would leave a trail of unerased labels behind. Thanks goes out to LobsterIan for reminding me of this.
- fixed a bug that's been around since version 1.50, where just moving tiles around in one of the unmodified original backgrounds in the background editor was not enough to cause the changes to be saved... you had to first paste/insert/delete a tile in the background for that.
- updated extended objects 8A-8D (switch palace switches) to break up when near or on screen boundaries like the game does. Thanks goes out to era64 for reporting this.
- updated some info for noteblock tiles 24 and 115 which are apparently unused. Thanks goes out to Kaijyuu for discovering this.
- corrected info on swooper bat, as its behavior apparently isn't related to its X position after all. Thanks goes out to Kaijyuu for reporting this.
- adjusted the "open level" button on the toolbar so it gets repainted as soon as you've saved to a different level number. Thanks goes out to TRS for noticing this.
- corrected info on Level Mode F, which reported that you could interact with layer 2, except you can't. Thanks goes out to smkdan for reporting this.
- removed setting 3 for the "Item Memory Index", as the game reportedly didn't implement this which meant it wouldn't function correctly. Thanks goes out to smkdan for reporting this.

Version 1.65 October 1, 2009

- fixed "export multiple levels to mwl files" feature, where LM was trying to open the ROM twice when it shouldn't have. This fails in 1.64, due to certain file operations having been tightened up to detect this kind of thing in old code.
- fixed saving to mwl files, where if the level had ExAnimations it wouldn't save them correctly and would then disrupt the level's Super GFX bypass settings. While this shows up in 1.64, the code was wrong in all 1.6x versions yet apparently still worked due to a strange coincidence.
- updated extended object 46 (midway point bar) to break up on screen boundaries like the game does. Thanks goes out to TRS for noticing this.

Version 1.64 September 24, 2009 (9 year Anniversary of Lunar Magic!)

- fixed a memory leak with sprites that exists in all prior versions of LM.
- fixed an issue where if you used entry 0x1F in the Extended Animation Frames list and saved to the ROM or a mwl file, LM would save the data but upon reopening the level it would not read anything from the list at all. Thanks goes out to Jimmy52905 for discovering this.
- fixed a problem with mouse gestures where holding down the right mouse button inside the editor window and then left clicking outside it would cause LM to crash. Thanks goes out to crushawake for finding this.

- adjusted a slope object in LM to emulate a game bug that causes the slope to break up on screen boundaries in some cases. Thanks goes out to Alex99 for noticing this.
- fixed the "view screen exits" toolbar button so it doesn't stay pressed if you switch to view screen or sub-screen boundaries, as reported by Maxx.
- corrected tileset info for Thwomp and Thwimp, reported by Noobish Noobsicle.
- fixed the animation of the wave tile in the overworld editor so it moves in the correct direction. Thanks goes out to Sukasa for finding this.
- fixed loading an unmodified title screen in the overworld editor when it's from an 8 MB ExLoROM game.
- fixed an issue in the overworld editor where if LM ever had to report an error while reloading the overworld and tile animation was turned on, LM would start spawning multiple message boxes. Thanks goes out to Obsidian for reporting this.
- updated the overworld palette editor so that alt-right click does a gradient instead of ctrl-right click, as in the level palette editor. Should have been done in version 1.63.
- fixed overworld exit tile 0x4D so it can connect directly to a level tile without malfunctioning, which the original game doesn't allow... I think Nintendo originally set it up as a corner path tile and didn't completely convert it when they decided to use it as an exit tile.
- added a new method to the overworld editor for linking star/pipe/exit tiles together with fewer steps using alt-left click.
- added an option to the overworld editor that allows turning off level 24's redirection of exits based on coins and time.
- added a setting to the overworld editor that can change the one level used to activate a secret exit when

beating a boss, which was used by the Big Boo Boss.

- fixed sprite interaction with custom block ASM in vertical levels. I actually fixed this years ago and temporarily released the fix as a patch, intending to later integrate it into LM for version 1.62. But mysteriously the fix never made it into the program... I suspect I was waiting till I had time to check/adjust the fix for the Japanese ROM as well, and forgot about it.
- implemented the "Mario touched top corner of block", "Mario within block (body)", and "Mario within block (head)" custom block ASM hooks. Two of these were used in a few of DW:TLC's blocks. Should of been done before, but likely wasn't for the same reason as the sprite fix.
- adjusted the 16x16 program icon for WinXP, since apparently XP has something against using 16 color icons at times.
- fixed mwl file association for Windows Vista, which should also fix it for non-administrator accounts on Win2k and up.
- file types mw0, mw1, mw2, and mw3 are no longer associated with Lunar Magic, as the program hasn't used those for saving levels since 2003. Note that this does not affect LM's ability to open levels saved in the older format.
- added a feature that allows LM to add a copier header to the ROM for you if it's missing, plus an option to do it silently.
- added exporting/importing palettes in YY-CHR .pal format.
- added a feature that allows LM to display custom tooltips and sprite tile arrangements for custom sprites based on the extra info bits (as done by a 3rd party ASM hack). Also added an option to turn this off. Check the help file for more info.

- added an extra category to the "Add Sprites" window called "Custom Collections of Sprites", which uses external files to allow custom sprites to be added a little more easily than using the "insert manual" command. See the help file for more information.
- enlarged the area used for LM's internal sprite Map16 data, and split it up into 2 separate files so custom sprites can use their own file. Check the help file for more information.
- replaced Lunar Magic's old code for handling RATs with a modified version of the code in Lunar Compress. This removes the limitation LM had about not honoring RATs that existed in a different LoROM bank than the data being protected, and also lets LM use the full size allowed by the RAT format.
- The FG Map16 data will now be protected by a RAT when saved, to make it easier for third party programs to avoid this data.
- converted LM's help file into a compiled HTML help file, as Microsoft is phasing out the old WinHelp system.
- thanks goes out to Smallhacker for gathering suggestions and bug info.

Version 1.63 September 24, 2005 (5 year Anniversary of Lunar Magic!)

- Adjusted some code so that importing a palette would enable the save level button. Thanks goes out to Juggling Joker for bringing it up.

- Fixed a bug in one of LM's ASM hacks which could cause the game to crash if a user inserted custom block ASM that tried to dynamically change a tile into a tile that is at or above 0x400. The fix will be installed when you save the Map16 data. Thanks goes out to Darkflight for submitting the hack that revealed the problem.

- Fixed a bug where the 3 8x8s line animation type was copying 4 tiles instead of 3 (in the editor). Thanks goes out to Smallhacker for discovering this.

- Fixed some code to prevent a rare case where pasting objects/sprites/tiles on the left side of the screen while moving left could, if timed correctly, paste the items on the other side of the editable area. Thanks goes out to Smallhacker for pointing this out.

- Fixed a couple bugs in the BG editor that involved dragging a tile pasted from the 16x16 editor.

- Lowered the max file size allowed for the ExAnimated file to reflect the true safe limit for that file (0x1B00, or 0x1A00 for All Stars + World, down from 0x2000).

- Fixed the tile arrangements of several platform sprites (5F, 62, A3, C4, and E0) in the editor to emulate how the game does it. Thanks goes out to Glyph Phoenix for pointing this out.

- Fixed the tile arrangement of Reznor (A9) in the editor. Thanks goes out to Bio for pointing this out.

- Updated info on the climbing net door sprite (54), to include a warning for not putting it on a sub-screen boundary. Thanks goes out to HyperHacker for discovering this.

- Updated info on sprite 88, which is apparently a winged cage that Nintendo didn't use. Thanks goes out to mikeyk & Smallhacker for discovering this.

- Moved the control-right click functionality in the palette editor for gradients to alt-right click. This way, when using control-left click to copy a color, you don't have to release the control key when you right click to paste (making it less annoying and more consistent with the other editors).

- Moved the control-right click functionality in the 16x16 editor to alt-right click, to make room for the new clipboard shortcuts.

- Added windows clipboard copy/paste using control left and right clicks in the 16x16 and 8x8 editor windows.

- The previously hidden credits and title screen editor modes of the overworld editor are now standard features.

- Added a new feature that allows overriding the sprite tile arrangements that LM uses by including new tile arrangement instructions in the sprite tool tip file. Read the help file for more details.

Version 1.62 April 11, 2004

- Fixed a bug that was introduced by the "Count Sprites" option in version 1.60. If the maximum sprites exceeded message box was ever actually displayed, it could cause various problems. Symptoms included strange values showing up in the GFX, ExGFX, Insert Level and Insert Directory address fields, as well as areas filled with garbage tiles showing up in unused portions of the BG Map16 data. The latter is caused by RAT tags getting embedded in the data (1.60 only), leading to your ROM slowly filling up with useless copies of it each time you save the Map16 data. This version of LM scans the BG Map16 data for the tags caused by the bug, and removes them before saving to prevent further space from being wasted. You may also need to set the address to insert GFX back to 40200, the address for ExGFX back to 100200, and erase garbage tiles left over in your BG Map16 data. Thanks goes out to TheClaw for submitting the hack that revealed the problem.

- Fixed a crash bug that would occur if you tried to insert a GFX or ExGFX file of size 0.

- Changed some code so that failure to extract a single GFX or ExGFX file from the ROM will not abort the rest of the extraction process.

- Fixed a bug introduced in 1.61 where if you have a sprite/object selected in the editor and try to use the clipboard keyboard shortcuts in the background editor window, the keys would be passed to the main editor window and replace the contents of the clipboard with the object/sprite selected.

- Updated sprite 5C,5E for info about floating on water with sprite buoyancy enabled. Thanks goes out to Kenny for bringing this up.

- Updated objects 82 and 83 to emulate garbage tiles effect when other tiles are placed behind the empty tiles of the bush object. Thanks goes out to thesubrosian for noticing this.

- Added an option in the options menu for turning off the sprite/object ID and object size info in the Add Object/Sprite windows, at Jesper's request.

Version 1.61 December 25, 2003

- Fixed a bug where if you set a level to use palette animation for color 0, Lunar Magic would shut down whenever it tried to read more data from the ROM. Thanks goes out to SomeGuy for pointing this out.
- Fixed a bug where the "Run in Emulator" function wasn't putting quotes around the file name and path, causing problems for snes9x when the path or file name contained a space. Thanks goes out to Excel for bringing this up.
- Fixed a bug where Lunar Magic would freeze if the user canceled expanding in the 8 MB dialog warning, but only if it was done after a “not enough room” message. Thanks goes out to Sendy for discovering this.
- Fixed the palette numbers being used by the editor to display the bob-omb, para bob-omb, bubble bob-omb, bullet bill, the torpedo launcher hand, puff of smoke, and fishin' Boo. Thanks goes out to KT15X and BMF for bringing this up.
- Updated the overworld editor to change the “Nothing?” sprite to “Koopa Kid x3”, and added display support for it. Thanks goes out to BMF for discovering this!

- Updated sprite 64 (rope mechanism), since it will be long or short depending on X&1, and will always be short if the sprite memory index is not set to 1. Also, only 2 long ropes can be on the screen at once during gameplay. Thanks goes out to Sendy for bringing this up.

- Updated tool tips for sprites 4D and 4E (Monty Moles), since they will follow Mario or walk with a hop depending on X&1. Thanks goes out to MetaRidley for noticing this.

- Adjusted some code so that using features like the Super GFX dialog will no longer cause tileset-specific Map16 data to be unnecessarily reloaded from the ROM. Thanks goes out to Jeffrey for bringing this up.

- Added keyboard shortcuts for the object and sprite editor windows and the "Open Level Number" dialog, highlighted the level number in the dialog, and added a feature to delete all sprites, objects and exits in a level with Ctrl+Delete at HyperHacker's suggestion.

- Added short tileset descriptions to the entries in the "Change Graphics in header" dialog, at HyperHacker's suggestion.

- Added buttons to copy and paste the extended animated tile data in the appropriate dialog, at Sendy's suggestion.

- Added Opera-style mouse gestures to the right mouse button. Left and right go back and forth in level history (you can hold shift to force this if the Sprite or Object editor window is open). Shift + Alt with left and right goes down/up by one level, and alt right goes to the level of the exit on the screen you clicked on. This feature can be disabled in the options menu if you don't want to use it.

Version 1.60 September 24, 2003 (3rd year Anniversary of Lunar Magic!)

- Included 64 Mbit ExLoROM support (8 meg ROM files). However, since it involves physically moving the ROM banks around which will make the ROM incompatible with other SMW utilities, **usage is not recommended** except for those that need an insanely large amount of space. You can use Shift + Page Down to force the conversion. You will need Snes9x 1.39a (or 1.39mk3) or higher to play the ROM (zsnes doesn't yet support it).

- The mwl file format has been changed so that levels can be in a single, self-contained file instead of the split files used in the old format. Lunar Magic will still be able to open files in the old format to maintain backward compatibility.

- Added a new ASM hack that allows you to create new animated tiles and do simple palette animation, on a per-level basis.

- Updated the ExGFX ASM hack to allow using up to 0xF00 additional ExGFX files, for a total of 0xF80 files. Also added a new dialog to allow Super GFX bypassing, which lets you access the new files and setup the FG/BG/SP/ExAn GFX from one dialog. The settings in it are level specific, meaning you don't need to use a global list of GFX files to load as with the older bypass code. The old list method is still supported, but it cannot access the new files.

- Added a feature to allow importing a custom palette directly from a ZSNES save state.
- Added an option in the options menu for highlighting the tile under the mouse cursor in the BG editor.
- Added an option in the options menu to count the number of sprites in the level when saving to the ROM, in case you exceed the maximum that the game can usually handle.
- Made a couple minor changes to LM's toolbar (added a "Scan for Undefined Exits" button, a "Super GFX Bypass" button, and a "Edit Extended Animated Frames" button).
- Fixed a bug in how Lunar Magic displays extended object 0x90 (the Boss Door). Due to strange coding on Nintendo's part, the door tiles break up when the door is on or beside a screen boundary. LM has been updated to do the same.
- Fixed a special case involving Lunar Magic displaying certain layer 2 level tiles (on levels using FG/BG tileset 3) with the wrong palette.
- Fixed a bug where hitting Ctrl-6 to advance to the next animated frame would have no effect on Yoshi coin palette animation when using a custom palette. Thanks goes out to BMF for pointing this out.
- Added a feature that lets you test the currently loaded ROM in the emulator of your choice by hitting F4 in the level or overworld editor.

- Using F5 & F6 in the Map16 editor will also now save and load the gameplay settings for the FG Map16 tiles.

- Added a feature that lets you use your own custom map16 data for displaying sprites in the editor (does not affect the actual game).

- Adjusted the palette editors so that ctrl-left clicking puts the color on the windows clipboard, so you can copy and paste between multiple instances of Lunar Magic.

- Added a new mode to the overworld editor to allow moving around the sprites. You can also set Mario's starting position in this mode by moving the Mario sprite around.

- Added a tool bar to the overworld editor.

- Added a dialog to the overworld editor to allow changing/deleting the overworld sprites for each slot. Thanks goes out to JW for initial research on this.

- Added a dialog to the overworld editor to allow changing the No-Auto-Move levels, which is usually used for the switch palace levels.

- Added Windows Clipboard support to the overworld editor's layer 2 8x8 editing mode for standard cut, copy, and paste operations.

- Added a new ASM hack for the overworld that allows up to 0x4000 silent event steps for layer 1 and 2 combined. The previous limit in the editor was 0x100 steps, which turned out to be a mistake since the actual limit used by the game was only 0x80 and exceeding it caused all silent event steps to not show up during gameplay.

- Included a new setting in the "Extra Options" of the overworld editor for toggling the ability to use the start button for scrolling on the main overworld map.
- Added a setting to the Boss Sequence dialog of the overworld editor for changing which level the earthquake sequence is activated on.
- The OV editor window was not redrawing itself to reflect changes made in the "Modify Level Tile Settings" dialog if it was activated using an alt-right-click, which has been fixed.
- The OV editor now has an option for viewing tile animation.
- A few of the keyboard shortcuts in the OV editor have been changed to be more consistent with the level editor. Keys 3-6 have been moved to 7-0. Sorry if that causes any confusion, I should have thought ahead...
- Added a new menu command to the OV editor to allow setting the FG GFX of each individual submap. You can even specify ExGFX files. However, you must use a star or pipe exit to make the game load the different graphics... a normal exit tile won't do it.
- Updated sprites 0x22, 0x23, 0x24, and 0x25 (the net koopas) to indicate whether they're below or above the net, which is apparently based on their X position.
- Added the ability to shift/wrap the currently selected 8x8 tile in the 8x8 tile editor left/right/up/down by one line of pixels using the shift and arrow keys.
- Fixed a few places where changing something wouldn't enable the quick save button in the main editor

(background editor, palette editor). Thanks goes out to Sendy and Someguy for bringing this up.

- If the midway entrance of a level is at the same location as the main entrance, the main entrance label will now simply display a leading ">" character. It just looks less messy this way when translucent text is enabled.

- Inserted some code to stop Windows from setting the window focus to external programs when closing LM's tool windows under certain circumstances. Thanks to the Juggling Joker for mentioning this.

- Fixed the palette number that the Ninji was using when displayed by the editor. Thanks goes out to BMF for pointing this out.

- Fixed an issue where the "Auto-Set Number of Screens" option should have really been internally turned off for level (omitted). Thanks goes out to BMF for bringing this up.

- Fixed an issue in the overworld editor where if you modified the Map16 data of a layer 1 level tile that can be "revealed" or "destroyed" through an event, in the game the revealed or destroyed tile still looked like the old unedited tile until the screen was refreshed.

- Fixed a bug where one of the extra events in the overworld that were removed by Nintendo (event 0x77) was not being enabled for use like the others, causing certain problems for any level that tried to use it.

- Fixed an issue where using other dialogs while the overworld palette editor was open at the same time could cause it to behave strangely.

- Fixed a bug where if you loaded the (omitted) screen in the overworld editor, then went back to the level editor and saved the currently loaded level, it would corrupt the level's custom palette (if it had one).

- Allowed editing the (omitted) screen palette from the overworld editor.

- Fixed a bug with the scan for undefined exits, where it wasn't checking for horizontal exit pipe tiles. Thanks goes out to Gandalf for pointing this out.

- Removed all except one of the Boss Door tiles from the exit scan, since they don't act as a door.

- Fixed a bug in ExGFX support for the Japanese ROM. Apparently the FG/BG GFX tracking code wasn't disabled like it should have been, which would cause bypassed FG/BG GFX to not load if you exited to a level that used the same GFX file in the standard list as the level you came from. Simply insert your ExGFX with the new version of LM to install the fix for this.

- Updated the tool tip for sprite AD (wooden spike moving up/down). It will move up or down first depending on its X position. Thanks goes out to Sendy for pointing this out.

- Updated the tool tip for sprite 19 (instantly displays level message #1). Apparently this sprite also causes Mario's current and last saved at overworld map number to be set back to the starting map of the game. Also included an optional fix for the sprite to remove this limitation (read the tool tip for more info).

- Updated the tool tip for sprite 9. The green koopa will jump high or low based on (Y&1), not on its height from the ground. Thanks goes out to Lord Allan for bringing this up.

- With the overworld editor completed, this will likely be the last major release of Lunar Magic. Future versions will probably just be for fixing any bugs that appear. A big thank you goes out to the whole SMW hacking scene, for helping to make Lunar Magic the program it is today. The last 3 years have been very memorable. And who knows, maybe I'll make another editor some day... -_^
- Set a new record for time between new releases (1 year).

Version 1.51 September 24, 2002 (...)

- Fixed a bug that caused the program to crash on startup in Win NT/XP based systems.
- Set a new record for time between new releases (4 hours).

Version 1.50 September 24, 2002 (2nd year Anniversary of Lunar Magic!)

- Added a tool bar to the main Lunar Magic level editing window. Buttons that are described as "quick" in the tool tips will skip the dialogs that normally appear when using the standard menu items for those functions.

- Added a menu item to scan for exit-enabled objects that lead to the bonus game levels, and included an option to do the scan automatically whenever you save a level to the ROM. This may help people who accidentally use exit-enabled objects without setting them up to go anywhere.

- Added the ability to create gradients in the palette editor using control right-clicking, as suggested by Sendy.

- Updated creating/eating sprite block information, thanks to Iggy.

- Corrected Dry Bones SP GFX info, thanks to Iggy.

- Updated information for a few sprites that can be made to go away using Sprite Command 0xD2, thanks to Iggy.

- Fixed various spelling errors and names, thanks to KJAZZ.

- Apparently there are far more sprites that turn into silver coins when a silver POW is hit than I had realized. Lunar Magic has been updated to reflect this. Thanks goes out to KJAZZ and Dejiko.

- Added extra information on sprite DF, thanks to EYE.

- Corrected the door tool tip, since apparently if Mario is small, he CAN enter it while riding Yoshi. Thanks goes out to EYE for pointing this out.

- Corrected a problem where Lunar Magic was not looking for the optional *.dsc file when opening a ROM using the command line. Thanks goes out to EYE for pointing this out.

- Lunar Magic would not reload GFX structures 32 and 33 for rendering of animated tiles unless you reopened the ROM. This has been changed so that they'll be reloaded whenever you reinsert the GFX.

- Fixed a small bug where hitting cancel in the palette editor would not reverse the changes if you had a custom palette.

- Fixed a nasty bug where if Lunar Magic tried to save a table of size 0 in the ROM, the RAT system would save it with a size of 0x10000. This could later cause large sections of data to get erased since SMW is a LoROM game and LM didn't expect data tables to be larger than 0x8000 bytes. Thanks goes out to EYE for supplying the hack that revealed this problem.

- Added a warning dialog to notify the user when one of the FG/BG/SP GFX or ExGFX files loaded by the level is larger then it should be, since some tile editors will modify the size of smaller files (which can cause glitches in the game).

- Fixed a bug where if you take a level that has a modified background image and changed the SNES Registers and Level Settings mode to have a level on layer 2, the custom BG flag would not be cleared when saving the level. This could cause several problems. One, SMW would try to decode the data as both a level and a background, though it would only display the level data. In some cases this could cause the game to freeze/crash. Two, Lunar Magic would decode the level data as a background but then discard it and display a blank level on layer 2, creating the illusion that LM was not saving layer 2. LM would also allow you to use the level number as the source for a background to copy in the "Switch BG" menu. And three, LM would attempt to erase the layer 2 level data as background data when saving over the level. Which means that if the level being erased was last saved in LM version 1.10 to 1.31, it could potentially cause other data in the same bank that was last saved in LM version 1.31 or lower to be erased! (Data protected by RATS is unaffected in versions of LM that support it, ie. LM 1.40+.) Thus this version of Lunar Magic correctly clears the custom BG flag when saving levels for layer 2, and also has extended checking to make erasing old levels with this problem safe for older non-RAT protected data. Thanks goes out to several people on the SMW forum for pointing out symptoms of this bug, which has apparently been in Lunar Magic for over a year and a half... O_O

- Cleaned up how Lunar Magic handles changing modes for levels on layer 2 to images, and vice versa. When switching to level modes that have fewer screens available than the current level mode, Lunar Magic will now simply delete the objects and sprites beyond the new maximum number of screens. Also added a warning dialog for this.

- Objects and sprites that are on screens beyond the maximum number of screens allowed for the current level mode will no longer be displayed on the last screen in Lunar Magic. The sprites will simply not be shown at all, and the objects may overwrite portions of the other level layer. This is to more accurately emulate how SMW displays it. If you have levels created in older versions of LM where the maximum number of screens were reduced due to a level mode change, you can hit CTRL-F8 to delete any extra objects and sprites.

- Changing the level mode setting will now cause Lunar Magic to automatically update the "Use Vertical Entrance Positioning" flag to reflect the level layout.
- Added some directional information to the tool tips of certain sprites.
- Added support for ROM_File_Name.ssc files to override the sprite tool tips.
- Tool tips will now show up in the Add Objects and Add Sprites windows.
- Tool tips for objects on layer 2 will now show up regardless of which editing mode you're in.
- Added a new option for making the text labels and outlines for entrances and exits translucent.
- Added support for exporting and importing Tile Layer Pro SNES palette files for levels.
- Added an item to the level menu for changing the Map16 page the BG is using.
- The background editor has received a long overdue update, and now includes multiple tile selection and drag/drop support, plus windows clipboard support.
- Attempting to insert non-existent GFX files will no longer mess up the ROM.

- Lunar Magic now saves the view animation option in the view menu to the registry, so that those with faster computers can start the program with this on all the time if they like.

- Lunar Magic will no longer switch to layer 1 editing mode when opening a different level. It will stay in the editing mode you're currently in, if possible.

- Lunar Magic now specifies the current directory when opening a file dialog to avoid the "My Documents" default folder that Windows 98 likes to use.

- Updated the dialog for the sprite header properties to explain the effects of sprite buoyancy on the game's speed and on interaction with layer 2.

Version 1.43 June 13, 2002

- Sprite 66 is apparently an upside down chainsaw, not just a duplicate of sprite 65. Lunar Magic has been updated to reflect this. A thank you goes out to Dwario for pointing this out.

- The OV editor window was not redrawing itself to reflect changes made in the "Modify Level Tile Settings" dialog, which has been fixed. Thanks goes out to Sendy for noticing this.

- Added a menu command to allow exporting the level as a 24-bit bitmap file.

- Added a "Recent Files" section to the file menu.

- Fixed a minor bug that caused the editor to refuse to open a MWL file if any of the file names within it were longer than 100 characters.
- Added a warning dialog to prevent people from inserting ExGFX as 4bpp when the ROM's GFX are still 3bpp or vice versa, as that will appear to scramble the graphics of one or the other. The help file already warns against doing this, but apparently not everyone reads it.
- Added a more informative dialog for those that try to write to a read-only ROM file.
- The Easter egg for support of SMW in the Mario All Stars + World ROM has been turned into a standard feature.

Version 1.42 February 9, 2002

- Fixed a bug from version 1.40 which could occasionally cause the bonus game entrance to use secondary entrance 0 or 100, and the Yoshi wings to use secondary entrance C8 or 1C8. The fix will be automatically inserted the first time you save a level to the ROM using the new version.
- Fixed a bug from version 1.40 where the X value in the main/midway entrance dialog for method 2 was sometimes being initialized to the wrong setting.
- Fixed a bug in the overworld editor where copying over *all* the tiles of the Layer 2 event path tile area in the Layer 2 8x8 editor mode with the blank "X" tile could

cause rather nasty things to happen if saved to the ROM.

- Fixed what appears to be some sort of flaw or oversight on Nintendo's part, where the midway point entrance ASM code does not check the "Use Vertical Level Positioning for all entrances" flag. The fix will be inserted the first time you save a level to the ROM with the new version. Thanks goes out to Dwario for pointing out that there was a problem.

- Added level name editing to the overworld editor (not supported for the Japanese version).

- Added level message box text editing to the overworld editor (not supported for the Japanese version).

- Added boss sequence text editing to the overworld editor (not supported for the Japanese version).

- Added viewing options to show a text label with the level number and/or event number beside levels on the overworld in the overworld editor. Thanks to Pikachu14 for the suggestion.

- Added a view option to display all Mario Paths as translucent tiles.

- Added a save prompt for when you try to open a different level/ROM/etc without saving the current level/overworld first. You can turn it off in the options menu if you want.

- Added an option that allows LM to remember its window size for the next time you run the program.

- Added a dialog to the overworld editor that allows changing the level numbers the game checks for when determining if a defeated boss sequence or the end game sequence should be displayed.

- Added a dialog to the overworld editor that allows changing the level numbers the game checks for to display special messages like the switch palaces and Yoshi's house (not supported for the Japanese version).

- Added a dialog to the overworld editor that allows altering the level numbers the game checks for to enable the changes that occur when special world is passed.

- You can now use the "Modify Level Tile Settings" dialog in the overworld editor for certain non-level tiles that can be revealed by events, such as the translucent bridge and door tiles.

- Sprites 83 and 84 can give a flower or feather, but LM was only displaying a mushroom for both, so the program has been updated to display the correct images. Thanks goes out to Sendy for pointing this out.

- Added support for ROM_File_Name.dsc files to override LM's tool tips for tiles.

- You can now use the middle mouse button to switch between sprite editing mode and the last used layer editing mode.

Version 1.41 January 1, 2002

- Fixed a bug from version 1.40 where there was a possibility of part of the overworld's layer 2 data being accidentally stored in a separate bank from the rest, making it inaccessible to the game and Lunar Magic. Thanks goes out to SomeGuy for submitting his hack which revealed the problem.

- Fixed a bug from version 1.40 where one of the sections of data saved by the overworld editor was not being erased prior to saving a new copy of it. This means repeatedly saving the overworld would result in the gradual build up of undeleted segments of data in the ROM.

- Fixed a compatibility bug in the new exit ASM code of version 1.40 which was causing old style secondary exits to use mangled "Mario Action" values during gameplay. The fix will be automatically inserted the first time you save a level to the ROM. Thanks goes out to VinnyMack for pointing this out.

- Corrected a minor flaw in the OV editor's simulation of passing events. Apparently only one tile can be "destroyed" per event... in cases where there are multiple entries for the same event, the one nearest to the bottom of the list is used.

- The "Future Layer 1 Tiles" option in the View menu of the overworld editor will now also make future path tiles translucent.

- You can now use the mouse scroll wheel in Win98 or above to increase/decrease the Z order of selected objects/sprites in Lunar Magic (I don't have a mouse with a scroll wheel though, so this feature hasn't been tested).

Version 1.40 December 25, 2001

- Added two new editing modes and several dialogs to the overworld editor for editing layer 1. It's now possible to move around the levels, pipes, star roads, the paths Mario walks on, the event activated by passing a level, the direction enabled by passing an event, etc.

- A far better method of keeping track of used areas in the expanded portion of the ROM has been implemented. This should prevent any future problems like the one in version 1.30. However, only data saved with the new version of LM will use it.

- Added unofficial support for JW's tile set specific animated tile modification.

- A new option in the view menu can enable tile animation in the editor to mimic the animation you would see in the game for coins, question blocks, backgrounds, etc.

- Added two new viewing options to simulate hitting POWs and Silver POWs.

- Using a control right click to paste an object/sprite/tile/etc from an editor window regardless of what you already have selected has been implemented in all editing modes of LM that didn't already have it. The previous function handled by control right clicking in the Layer 2 Event Editor Mode of the overworld editor has been moved to the shift key.

- Several ASM modifications have been done to SMW's level screen exit code. One such modification has resulted in the removal of the old limitation of levels from 0XX not being able to exit to levels in 1XX and vice versa... any level can now exit to any of the 0x200 levels, and any level can access any of the 0x200 Secondary Entrances. Note however that this is dependent upon Lunar Magic converting the exits in a level from the old style to the new style; the old style exits in unmodified levels will still follow the old rule. For example, if you use a new exit to go from one level block to another, and then use an old style exit in that level, the old style exit will unexpectedly divert you back to the original level block. But this can easily be solved simply by resaving the level that contains the old style exit so LM can automatically convert all the old style exits in the level to use the new enhancement.

- Secondary Entrances can now be set for individual screens, allowing you to use both standard and secondary entrances within the same level. Thus the "Enable Secondary Entrances" option in the level menu has been moved to the "Add/Modify/Delete Screen Exits" dialog.

- Both the standard and secondary entrance dialogs now have a second method for setting the X/Y and Sub-Screen position which allows for any X/Y coordinate, with an option for using either the original method or the new method for setting Mario's position.

- The standard and secondary entrances can now directly specify if the level should be a water level or slippery (or both!), rather than using one of the Mario Action settings (which was previously the only way one could enable either of these).

- The "Sub-Screen Boundaries" option in the "View" menu has been updated to display the screen number and sub-screen label for each section, which may make placing entrances a bit easier.

- A few minor modifications have been made to the MWL format to accommodate the new exit information that has to be stored. Older versions of LM can still read the newer MWL files, but they will import the main and secondary entrances incorrectly. LM continues to remain backwards compatible with MWL files created with older versions of the program.

- It's now possible to resize objects using the mouse. Just move the mouse cursor along the right and/or bottom edge of an object's tiles until the cursor changes.

- You can now export and import custom level palette files (MW3) directly through the file menu.

- An "Export Multiple Levels to Files" command has been added to the file menu.

- The "Insert at the original address of the level" option of the "Save Level to ROM" dialog has been removed. It's an old function that should have been taken out long ago.

- The "Enhanced Level Management" option in the save level dialog was removed, as recent changes have made turning that off inadvisable under any circumstances.

Version 1.31 October 1, 2001

- Fixed a nasty bug from 1.30 that could potentially erase non-level data such as BG Map16 data, overworld data, etc when saving a level. Thanks goes out to SomeGuy for submitting the patch that revealed this problem.

- Fixed a crash bug (!) that would occur in 1.30 if you left-clicked on the currently selected 8x8 tile in the 16x16 tile editor window without opening the 8x8 editor at least once beforehand.

- In the 16x16 properties dialog of the 16x16 editor window, the upper right and bottom left 8x8 tile numbers were switched around. Thanks to Sendy for pointing this out.

- Fixed a bug where FG tiles >= 0x400 in the 16x16 properties dialog of the 16x16 editor window incorrectly reported tile gameplay settings of 0x130 in the 16x16 editor window, even when they had been changed.

- Sprite 18 (a surface jumping fish) is not "unused" as a tool tip in 1.30 indicated. It has now been added to the sprite list, and the tool tip has been updated. A thank you goes out to Iggy for pointing this out.

- Sprite A4 (a spike ball that floats on water) is slow or fast based on (X&1), so Lunar Magic's description of it has been updated.

Version 1.30 September 24, 2001 (1st year Anniversary of Lunar Magic!)

- Lunar Magic now supports the Japanese version of SMW!

- The overworld editor is no longer hidden, as just about everyone knows how to get into it by now anyway. Only minor updates have been done to it since the last version.

- Added a dialog to the overworld editor that can switch the music selection of the submaps around. Thanks goes out to ßouché for finding the data to support this.

- Added tool tips for objects and sprites in the main editor window. If you have a really old version of comctl32.dll though (earlier than 4.70), this feature will be automatically disabled.

- The palette editor is now a modeless dialog, meaning you can now keep it open while you edit the level. The main editor window is updated immediately whenever changes to the palette are made.

- You can now use a control-left click to copy an existing color in the palette editor, and a right click to paste it back in.

- Reversed the byte order of the SNES color values reported by the palette editor, since it was inconsistent with the order of the PC color values.

- Fixed a bug that appeared in version 1.20 of Lunar Magic where saving a level without a custom palette overtop of a level with a custom palette resulted in the level using a custom palette of "random" colors.

- Fixed another bug from version 1.20 where the "File", "Exit" command was closing the MDI child window instead of the parent window.

- Fixed a bug that caused a Yoshi egg placed at (X&3)==3 to appear yellow instead of blue.
- Fixed a couple bugs in rendering Extended Objects 91-96 at certain screen coordinates. A thank you goes out to Iggy for sending me the level that allowed me to track this one down.
- Updated the BG ASM hack to support up to 16 pages (0x100 tiles/page) of Map16 data, a significant increase over Nintendo's limit of 2 pages. You can change the active page used by the background by using the Page Up/Down keys in the Background editor window. Any ROM with the old BG ASM enhancement will be automatically detected and updated when a modified background is saved.
- Included a new ASM hack to support 16 pages of FG Map16 data, up from Nintendo's limit of 2 pages. The first extended page has been made tile set specific, while the rest will remain fixed for all levels. The new tiles can be accessed through the "Direct Map16 Access" menu of the "Add Objects" window.
- Added the ability to set a FG Map16 tile to "mimic" the gameplay properties of another tile. For example, if you copy a coin tile into one of the new FG pages, the tile will not only look like a coin, it'll actually act like one too! To only copy a tile's appearance as in previous versions of Lunar Magic, just hold down the control key while pasting.
- Added the ability to export/import the currently loaded FG and BG Map16 data in the 16x16 Tile Map Editor Window using keyboard shortcuts (F2,F3,F5,F6,F7,F8). Also added confirmation dialogs for those keys plus F9.

- Corrected a minor bug in the Background Editor Window, where it was using the temporary data of the currently selected tile in the 16x16 Tile Map Editor Window when placing a tile into the Background Editor Window...it _should_ have only been using the tile number. This created the illusion of being able to take a 16x16 tile, modifying it and then pasting it directly into the background without actually saving it somewhere in the Map16 data first.
- Added several minor enhancements and buttons to the 8x8 editor window.
- Added a music and time limit bypass dialog to override the settings in the level header.
- Improved Drag/Drop logic. The program isn't as strict regarding valid mouse positions when moving things near the edge of the screen or an invalid area.
- Repaired a bug where decreasing the Z-order of an object on layer 2 beyond zero would result in the object jumping to layer 1 (I'm surprised this has never been noticed before...)
- Added 32 Mbit (4 meg) ROM expansion support.

Version 1.20 May 20, 2001

- Added a Windows Help file to the program. The online documentation on the website will be taken down, since the help file is more easily maintained and generally more helpful.

- Added Cut/Copy/Paste commands for objects/sprites to the "Edit" menu. This allows copying compatible objects or sprites between levels. In fact, since it was implemented using the standard Windows Clipboard, you can even copy between multiple instances of Lunar Magic!

- Implemented basic SNES layering and color addition support, so that level modes 1E and 1F will now show up in the editor as they would in the game. Although technically none of Nintendo's levels appear to have used those modes in the first place, it's still rather interesting to be able to make a translucent level. ^^

- Rewrote the animated tile support; Lunar Magic will now read the data from the ROM rather than having it hard coded. Additionally, the “On/Off” block and the “always-turning” turn block will now show up as they'd appear in the game.

- Included a minor ROM modification that will enable support for the animated water surface tile in tile sets 4,5,7,9 & D (in other words, all the tile sets where the water surface would show up as a blue question block). The modification will be inserted automatically the first time you save a level to the ROM.

- Added an ASM hack to allow appending a selectable ExGFX or GFX file to the end of the animated tile area, for use in animated tiles.

- Added Flip X/Y buttons to the 16x16 editor window.

- Made a few minor modifications to the placement of certain window elements. Probably the most noticeable change is that the status bar is now anchored to the bottom of the main window, which is really where I should have put it in the first place.

- Fixed a minor bug in the 8x8 tile editor, where it wouldn't update the colors of the current palette it was using if the window was open while a different level was loaded. There was also an annoying screen flicker when that happened, which has also been fixed.

- The "Add Sprites" window will now display a small line of text at the top left of the preview screen to suggest which GFX files could be loaded into SP3/SP4 to view the sprite properly.

- Sprite 8C, the smoke and fire generator for Yoshi's house that also allows Mario to walk off the edge of the screen without passing the level, will apparently not display any smoke or fire when (x&1) is true. Lunar Magic has been updated to do the same.

- Added a third category list to the “Add Sprites” window. The new category will contain the special sprite commands (like layer 2 scrolling, for example) and the sprite generators.

- Object 12 apparently does not detect and adjust for the tiles it is overwriting for the 4 upside-down slopes. This has been corrected in Lunar Magic. A thank you goes out to Jonathan for noticing this one.

- Lunar Magic will now identify and display in the status bar the contents of a block tile that has been placed using the Direct Map16 Access feature.

- Put in a warning message to notify the user when the exe file has been modified/corrupted by something (like a virus, for example). Previously the program would just silently terminate itself on startup when that happened, which wasn't exactly informative.

- Various other minor updates.
- omitted.

Version 1.11 February 9, 2001

- fixed a bug in the "Bypass Standard SP GFX List" menu that affected vertical levels. It caused Lunar Magic to incorrectly save the list index number to the ROM/MWL file and consequently made the game load the wrong sprite graphics.
- sprite 63 apparently depends on the value of (x&1), so I fixed it.
- included the "Open Level from Address" file menu command from the debug version. It's useful for viewing some things that aren't in the main level pointer table. Try address 0x30338 for the boss monster test room. ^^
- implemented a dialog to manually enter the four 8x8 tiles used to make up a 16x16 tile in the 16x16 Tile Map Editor, which is accessible through the previously unused "Edit 16x16 Attributes" button. Although standard mouse clicks for copy/paste already do this when the 8x8 Tile Editor Window is open, the former is more intuitive for beginners.

- invalid areas of the main screen are now painted black, rather than tiling the background image behind everything. This is to make it easier to see the valid workspace area of the level.
- layer 2 levels are now painted with the default back area color rather than a blue gradient, for the sake of correctness.
- omitted. ^^

Version 1.10 December 25, 2000

- added an ASM hack to allow up to 0x80 additional GFX files to be stored in the ROM for level use, called ExGFX files. Also added a FG/BG/SP GFX tileset list bypass hack so a level can access these files. Basically, this creates a new table of 0xFF list entries that you have direct control over, allowing you to select exactly which 8 GFX files are loaded by a level. Check the documentation for details.
- added an optional ASM hack to allow the use of 4bpp tiles in SMW instead of 3bpp tiles. In other words, most 8x8 tiles will have access to all 16 colors of a palette instead of only 8. This increases the amount of raw GFX data to compress by ~25% though.
- modified the GFX insertion code so that if you run out of room inserting at 0x40200-0x60200, it will insert the rest of the GFX to the expanded portion of the ROM instead of just giving an error.

- modified the GFX insertion code to use the level manager whenever graphics are stored in the expanded ROM space (meaning it will erase the old GFX and scan to ensure it doesn't overwrite things).

- added the "Classic Piranha Plant" and "Death Bat Ceiling" to the sprite list, since they can be safely implemented now with ExGFX.

- determined that "Sprite Display 1" in the sprite header selects the sprite memory and the maximum number of sprites per screen.

- added a couple menu items that allow you to extract and insert the shared SMW palettes, useful for backing up and restoring colors.

- located and implemented the last missing level palette colors!

- editing the colors of the palette loaded by a level is now supported.

- added another ASM hack to give any level the option of having its own custom palette instead of using the standard shared palettes.

- registered another file type (MW3) for storing custom Mario World level palettes.

- the title screen colors are now loaded into the main level palette when viewing intro level 0xC7, unless it has a custom palette.

- Lunar Magic now inserts a minor hack into the ROM when saving level 0xC7 (the intro level) to prevent the game menu from changing the back area color to other hard-coded values. If you don't want the hack inserted for some reason, hit F8 before saving this level.

- a basic background tilemap editor is finally in place, as well as a new ASM hack to support storing BG tilemaps in the extended ROM area.

- a new view menu item called "Special World Passed" was added, to allow viewing level differences caused by completing Special World.

- I've removed the "Custom Collections of Objects" category from the "Add Objects" window... this is only usable in the debug version anyway. Consequently, the "custom.bin" and "custom.txt" files are no longer included in the zip file distribution.

- the options in the "Options" menu will now be saved to the Windows registry.

- fixed a potential bug in the level manager that could cause Lunar Magic to crash when trying to erase a corrupt level in the ROM.

- fixed a bug where the level manager wasn't correctly scanning some sprite pointers. This means that when it was removing a level there was a small chance that some data would only be partially erased.

- various other minor changes.

Version 1.03 October 28, 2000

- fixed a bug that was caused by an accidental change I made in version 1.01. It seems the "Sprite Display 1" value in the "Level/Change Sprite Header" menu was no longer being saved, making it impossible to change the value in the GUI. Thanks goes out to DarKnight13 for pointing this out.

Version 1.02 October 10, 2000

- fixed a bug that I suspect was causing an overlapping level problem for many people. Apparently when saving a level to the ROM, Lunar Magic wasn't storing the last address used to the ROM until after you did another ROM operation. Which means if you went and saved a level, immediately exited LM (or reloaded the ROM), then later came back and saved a different level without changing the address that showed up, you'd often end up over-writing the first level. I can't believe I never noticed this bug until now... O_O
- implemented Enhanced Level Management. Enabling this option will cause Lunar Magic to erase the level being replaced (if it's in the expanded ROM area), then scan and adjust the destination address to avoid overwriting any other data. For best results, don't use this on old test ROM's that will likely have multiple undeleted and unreferenced levels from older versions of Lunar Magic.

Version 1.01 October 2, 2000

- fixed a minor display bug involving sprites in vertical levels.
- included an option for using joined GFX files instead of split ones.
- included Layer 3 options in the "Change Other Properties" Menu.
- documented the unknown bit in the "Change Properties in Header" Menu, which controlled Layer 3 layering. (thanks goes out to Mr. 207 for noticing what this bit was doing)
- documented and replaced the second value of the "Change Properties in Sprite Header" with options for sprite buoyancy.

Version 1.00 September 24, 2000

- First Release.

Contact Information

FuSoYa, Defender of Relm

Website: http://fusoya.eludevisibility.org

Legal Notice

Main Window Mouse Behavior

In general, a right click is used to paste and a left click is used to select. This tends to hold true for all the windows, so once you know that it doesn't take long to figure out the rest of the program. However, the main level editor window has a few extra things you can do with the mouse when holding down keys such as “shift”, “control”, and “alt”, so you may want to refer to the table below as a quick reference.

The editor also supports Opera-like mouse gestures for moving between levels.

Double Left Click on Tile	Select tile in 16x16 editor.
Left Click on Unselected Object/Sprite	Deselect any currently selected objects or sprites, and select the new object/sprite.
Left Click on Nothing	Deselect any currently selected object/sprite(s).
Left Click on Nothing and Drag	Deselect any currently selected objects or sprites, create a rectangular selection area, and select all objects/sprites that have a single tile or more within it.

	Note: Any object whose tiles are all hidden by other objects in the rectangle will not be selected. This quirk does not exist when working with sprites.
Left Click on Object or Sprite(s) and Drag	Move objects/sprites. Lunar Magic will first bring all the selected objects/sprites to the foreground, preserving the Z order only among the selected objects/sprites.
Control + Left Click on Object/Sprite	Select or deselect object/sprite from currently selected group of objects/sprites.
Hold Shift + Left Click on Object/Sprite(s) and Drag	Restricts object/sprite dragging to current "screen" only, allowing Lunar Magic to preserve the Z order of all objects/sprites on that screen.
Left Click on edge of a selected object tile and Drag	Resize the selected object(s), if the size mouse cursor is being displayed. Not all objects can be resized, and some only in certain directions. Sprites cannot be resized.
Right Click with Object/Sprite(s) Selected	Paste a copy of currently selected object/sprite(s) at mouse position.
Right Click with no Object/Sprite(s)	Paste object/sprite(s) from the Object/Sprite Editor Window at mouse position. Whether

selected and Object/Sprite Editor Window is Open	you paste a sprite or object depends on the Edit/Mode setting.
Control + Right Click and Object/Sprite Editor Window is Open	Paste object/sprite(s) from the Object/Sprite Editor Window at mouse position regardless of whether or not you already have objects/sprites selected.
Control + Shift + Right Click and Object Editor Window is Open and in Direct Map16 Access List	Fill with FG tile(s) from the Object Editor Window at mouse position if not in sprite editing mode.
Right Click with no Object(s) selected and Map16 Editor Window is Open while Object Editor Window is Closed	Paste FG tile(s) from the Map16 Editor Window at mouse position if not in sprite editing mode.
Control + Right Click and Map16 Editor Window is Open while Object Editor Window is Closed	Paste FG tile(s) from the Map16 Editor Window at mouse position if not in sprite editing mode, regardless of whether or not you already have objects selected.
Control + Alt + Right Click and Map16 Editor Window is Open	Paste FG tile(s) from the Map16 Editor Window at mouse position if not in sprite editing mode, regardless of whether or not you already have objects selected.

Control + Shift + Alt + Right Click and Map16 Editor Window is Open	Fill with FG tile(s) from the Map16 Editor Window at mouse position if not in sprite editing mode.
Alt + Right Click on Object/Sprite	Edit single object/sprite manually.
Alt + Left Click on Game View Screen	Drag Game View Screen (only when view option is on).
Middle Click	Switch between sprite editing mode and the last used layer editing mode.
Alt + Middle Click	Edit screen exit for the screen you clicked on.
Hold Right Mouse Button down, move left/right, release and Object/Sprite Editor Window is Closed	Mouse Gesture - Go back/forward in level number history.
Hold Shift and Right Mouse Button down, move left/right, release.	Mouse Gesture - Go back/forward in level number history regardless of whether or not the Object/Sprite Editor window is closed.
Hold Shift + Alt and Right Mouse Button down, move left/right, release.	Mouse Gesture - Go back/forward one level.
Hold Alt and Right Mouse Button	Mouse Gesture - Go to exit level destination of screen

down, move right, release.	that you clicked on, if it has one.
Mouse Scroll Wheel while object/sprite(s) selected	Bring Forward/Send Backward.
Control + Alt + Mouse Scroll Wheel while object/sprite(s) selected	Bring to Front/Send to Back.
Control + Shift + Mouse Scroll Wheel	Scroll vertically.
Shift + Mouse Scroll Wheel	Scroll horizontally.
Control + Mouse Scroll Wheel	Zoom in/out.

File Menu : Open ROM

This will open the Mario World ROM you wish to use. You cannot edit levels without first opening up a valid ROM, because the program requires information in the ROM to render certain objects and levels.

Currently both the American and Japanese version 1.0 Super Mario World SNES ROMs are supported. The SMW portion of the American version 1.0 Mario All Stars + World SNES ROM is also supported. Any SNES ROM you use must have the standard 0x200 byte file header. If your ROM does not have one, the program can add one for you.

If the program warns you of a corrupt checksum when you open a ROM for the first time, it means the ROM is corrupt or has been previously modified by another program. It's strongly recommend that you locate a pure copy to replace it before you start to edit levels, or you could end up with serious problems later.

File Menu : Open Level Number

This will open up one of the Mario World levels, discarding any unsaved changes in the current level. You can open any level in the ROM from 0-0x1FF. Most of these levels are empty however, so you'll often just get a screen with the word "test" on it. The levels are numbered according to their pointers, not according to the order they are encountered in the game or on the overworld map.

File Menu : Open Level Address

This menu command will allow you to open a level from the exact ROM offset you specify. This should only be used for loading a level that does not have a pointer in the main level pointer table. There aren't very many of these however, so it's unlikely you'll use this menu much. No sprite, entrance, or background data will be loaded when you use this command. Also, Lunar Magic will report your current level number as the last level number you were on, which means if you attempt to save the level it will be inserted as one of the standard 0x200 levels. Since Lunar Magic won't know where the actual pointer for the original level data is (assuming it even has one), you may have to manually find and repair this pointer yourself if you want the changed level to be used in place of the original.

Following is a list of all known locations of level data that is not referenced in the main pointer table, along with a couple addresses that may contain the pointers for them. This list is from the source code of a program that was made to scan the entire level area of the ROM and match the data with the pointers to determine gaps where an unknown level may be located.

```
//cancel out the areas we already know about (24)
CancelOut(0x30263);
CancelOut(0x302BD);
CancelOut(0x30338); //Boss monster Test Room!
CancelOut(0x30464);
CancelOut(0x304EB);
CancelOut(0x3059C);
CancelOut(0x305AB); //duplicate level
CancelOut(0x30701); //duplicate layer 2
CancelOut(0x3073D);
CancelOut(0x30875); //duplicate ghost house end
//check out 0x2DD0F for 2 byte pointers!
CancelOut(0x36BFB); //the coin/time levels
CancelOut(0x36CB0);
CancelOut(0x36D72);
CancelOut(0x36DBE);//I recognize this!
CancelOut(0x36E7E);
//check out 0x2D966 for 3 byte pointers!
CancelOut(0x38200); //Ghost house entrance!
CancelOut(0x38218); //blank (header only)
CancelOut(0x3821E); //Castle entrance!
CancelOut(0x3824E); //no Yoshi star entrance
CancelOut(0x38260); //Ghost house exit!
CancelOut(0x38281); //3 bushes, no ground (garbage?)
CancelOut(0x38290); //Another Castle Entrance
CancelOut(0x3829F); //no Yoshi again
CancelOut(0x382B1); //no Yoshi again
```

File Menu : Save Level to ROM

This does pretty much the same thing as the Save Level To ROM As menu, but it doesn't prompt you for any options.

File Menu : Save Level to ROM As

This will allow you to insert the current level back into the ROM. All relevant pointers for the level are updated automatically, and the old data is erased if it resides within the expanded portion of the ROM.

The **PC address to insert** is the ROM offset that you wish to insert the level to. Changing this isn't really necessary, as the program can figure out for itself where it's safe to store things. The address will automatically increment and jump around a bit as you save more data in the ROM. It will generally only save to the address you provide if: it actually exists in the current ROM, it's in the expanded area of the ROM (0x80200 or higher), there's enough space there, and no other override takes priority (like the "prefer saving in 2MB+ ROM area" option).

The **Level number to save as** is to input which of the 0x200 levels you wish to save the current level into. The default value is the current level number you're on.

The **Expand the ROM to 1 Meg** option will expand the original 512 K ROM to 1 Meg, which will give you more room to insert data. If Lunar Magic runs out of space, it will just prompt you for permission to expand the ROM further if necessary.

The **Transfer Secondary Entrances** option instructs Lunar Magic to scan the secondary entrance table for entries where the destination level number matches the current level number, and switch it to the new level number that you're saving this level as. If the level number you're saving as is identical to the current level number, then this option will have no effect.

File Menu : Open Next Level in ROM

This will open the next level number (+1) in the ROM, and discard any unsaved changes in the current level. Normally you'll use the keyboard shortcut for this to quickly flip through levels.

File Menu : Open Previous Level in ROM

This will open the previous level number (-1) in the ROM, and discard any unsaved changes in the current level. Normally you'll use the keyboard shortcut for this to quickly flip through levels.

File Menu : Open Level From File

This will allow you to open up a previously saved MWL (Mario World Level) file. An MWL file is a self-contained binary file that contains the level layout, background tile map, sprite data, palette, secondary entrances, and a few other things required for the level. These provide an easy way to export and transfer basic SMW levels between ROMs, friends, etc.

However, these files do not contain any actual graphics. Check the Extract GFX Files menu description on how to get those. They also don't include any other level-independent bits of data (Map16 data, shared palettes, and so forth).

Versions of Lunar Magic prior to 1.60 used a different MWL format that split the data into multiple files (MWL, MW0, MW1, MW2, and MW3). Lunar Magic can still open this older format, but you can only save in the new format.

File Menu : Save Level to File

This will let you export a SMW level to a MWL file. Check the Open Level From File menu description for more information on MWL files.

File Menu : Save Level to File As

This does the same thing as the Save Level to File menu command, except it lets you specify a new filename.

File Menu : Setup Tile Editor

From here you can set up the path to the external tile editor you wish to use when you hit the "edit" buttons for one of the slots in the various graphics dialogs of Lunar Magic. This will launch the tile editor to edit the selected file in the Graphics or ExGraphics folder. Remember you must first extract the files from the ROM, and you must still reinsert them to the ROM after editing them.

The **Use Above Custom Command Line Arguments** option is used to enable setting custom **Command Line Arguments** for running the tile editor. While not required for most programs, it can sometimes be useful for setting certain options on the command line. The default setting is "%1", which is just the path and filename of the current graphics file enclosed in quotes.

The **Replace yychr.pal file with current palette** option will cause Lunar Magic to replace the default yychr.pal file that the YY-CHR tile editor uses with one that has the current palette for the level or overworld submap you're working with. Just make sure that the tile editor's folder is in a writable location for this option to work.

The **Set transparent colors to blue** option will cause Lunar Magic to replace color 0 for all palettes with blue in the yychr.pal file. This can often make it easier to distinguish from other colors. Otherwise the transparent color is normally set to whatever the back area color is (or mostly left alone for layer 3 graphics). Note that if the option to replace the yychr.pal file has been disabled, then this option will have no effect.

The settings in this dialog are saved to the registry.

File Menu : Extract GFX from ROM

All the graphics in the Super Mario World ROM are compressed. To edit the graphics, you must first use this menu command to decompress and save all the graphics in the ROM to separate files (or to a single file called "AllGFX.bin", if the Use Joined GFX Files option is enabled). A folder called "Graphics" will be created in the directory containing the ROM, and 52 numbered files (roughly around 219 K total) will be placed into it.

To edit the contents of these files, you can just use your favorite SNES tile editor of choice like Tile Layer, etc. Lunar Magic also has a small built-in 8x8 tile editor you can use to modify some of these files (see the 8x8 Tile Editor Window description). You may find the information in the GFX File Information somewhat useful when editing these files.

Just remember to later re-insert the graphics into the ROM using the Insert GFX to ROM menu command.

File Menu : Insert GFX to ROM

This recompresses and inserts into the ROM all the GFX files that you previously obtained from the Extract GFX from ROM menu command, and the old data is erased if it resides within the expanded portion of the ROM. Lunar Magic compresses the GFX fairly quickly, and even achieves a slightly better compression ratio than Nintendo did.

Most of the modified graphics will not show up in Lunar Magic until you switch or reload the level.

The **PC address to insert** is the ROM offset that you wish to start saving the graphics at. It's usually best to insert at the default address of 0x40200 since this is the starting address of the original GFX. If the program runs out of room inserting here, it will simply insert the rest of the graphics to the expanded portion of the ROM.

The **Expand the ROM to 1 Meg** option will expand the original 512 K ROM to 1 Meg, which will give you more room to insert data. If Lunar Magic runs out of space, it will just prompt you for permission to expand the ROM further if necessary.

The **Modify the ROM with ASM to use 4bpp Tiles instead of 3bpp Tiles** option will insert an ASM hack that allows most tiles to access all 16 colors of a palette instead of the standard 8 that Nintendo stuck with. This will increase the amount of GFX data to insert into the ROM by about 25%. Note that the program will not uninstall the ASM hack once it is in place, so turning this option off after you've already inserted graphics to the ROM with the option on will have no effect. If you have previously inserted ExGFX as 3bpp tiles and you decide to use this option, you must later re-insert the ExGFX so they will be inserted as 4bpp tiles. Otherwise, any existing ExGFX in the ROM will appear as garbled graphics when exporting or viewing them.

Technical Note: GFX files 32 and 33 must begin within the same bank... these are the largest two files, so they are inserted first. You should thus choose an insertion address at (or close to) the start of a LoROM SNES bank, or you may get an error message when one of them cross a bank boundary.

File Menu : Extract ExGFX from ROM

This will decompress and extract all the ExGFX from the ROM that have been previously inserted with the Insert ExGFX to ROM menu command. They will be placed in a folder called “ExGraphics” in the directory that the ROM resides in.

File Menu : Insert ExGFX to ROM

This will recompress and insert all your custom ExGFX files to the ROM, and the old data is erased if it resides within the expanded portion of the ROM. ExGFX are "Extra Graphics" that are not part of the original Mario World game. Lunar Magic inserts an ASM hack to allow levels to access these additional files through the bypass FG/BG/SP GFX list level menu commands.

Since these graphics files are not in the original game, it's up to you to create the files that you want to insert. All you have to do is create a folder called "ExGraphics" in the ROM's directory, place your files into it and rename each one as "ExGFX#.bin", where "#" is a hex number from 80-FFF (do not insert a leading zero for numbers less than 100). Each file should usually be 4 KB (0x1000) bytes, and contain standard SNES 4bpp tiles. What you will likely end up doing most of the time though is simply taking one of the multiple original GFX files from the "Graphics" folder, then renaming and moving it to the ExGFX folder to modify however you like.

The only way to use these extra graphics in a level is through something like the Super GFX Bypass dialog.

Most of the modified graphics will not show up in Lunar Magic until you switch or reload the level.

The **PC address to insert** is the ROM offset that you wish to start saving the graphics at. It's usually best to insert at the default address of 0x100200 since this is the starting address of the 2 meg expansion.

The **Expand the ROM to 2 Megs** option will expand the ROM to 2 megs, which will give you more room to insert data. If Lunar Magic runs out of space, it will just prompt you for permission to expand the ROM further if necessary.

The **Modify the ROM with ASM to use 4bpp Tiles instead of 3bpp Tiles** option will insert an ASM hack that allows most tiles to access all 16 colors of a palette instead of the standard 8 that Nintendo stuck with. This will increase the amount of ExGFX data to insert into the ROM by about 25%. Note that the program will not uninstall the ASM hack once it is in place, so turning this option off after you've already inserted graphics to the ROM with the option on will have no effect. If you have previously inserted standard GFX as 3bpp tiles and you decide to use this option, you must re-insert the GFX so they will be inserted as 4bpp tiles. Otherwise, any existing GFX in the ROM will appear as garbled graphics when exporting or viewing them.

Note: If you are not using the 4bpp ASM hack, the ExGFX range of E00-FFF has been set aside for files to be inserted/extracted "as is" so they can be excluded from 3bpp/4bpp conversions. This range should then be used only for layer 3 GFX (which must remain 2bpp), layer 3 tilemap files (which shouldn't be converted), and for files intended for slot AN2 (which must remain 4bpp) in the level editor.

Note 2: The "Use Joined GFX Files" option will not affect ExGFX. They are always stored as separate files, as their sizes can vary and it also makes it easier for a user to add or remove any number of ExGFX files at will.

File Menu : Extract Old Bypass List from ROM

This will extract the bypass FG/BG/SP GFX list from the ROM to a file called “Bypass.lst” by default. This is useful for backing it up or transferring it to a different ROM. Check the documentation of the older bypass menu commands for what this list is used for.

File Menu : Insert Old Bypass List to ROM

This will reinsert the bypass list to the ROM from a file. This list will have previously been obtained from the Extract Bypass List from ROM menu command.

File Menu : Export Level Palette to File

This will export the palette of the currently loaded level as a custom palette file. You can save the palette in PAL format for YY-CHR, Tile Layer Pro TPL format for SNES, or the custom MW3 format.

If you want to use the palette file in TLP, ensure that TLP is set to load 256 color entries.

File Menu : Import Level Palette from File

This will import a custom palette into the currently loaded level. You can import a file using the PAL format for YY-CHR, the Tile Layer Pro TPL format for SNES, the custom MW3 format, or from a ZSNES save state.

If you save the palette in TLP, make sure that it's set to 256 entries using the SNES palette format.

Note that if your level does not already use a custom palette, the custom palette setting will be automatically enabled for you.

File Menu : Extract Shared Palettes from ROM

This will extract all the shared level palettes in the ROM to a file called smw.pal by default. It's useful for backing them up or transferring them to other ROMs. Note that this file will not include custom level palettes, since those are stored along with the level itself.

File Menu : Insert Shared Palettes to ROM

This inserts the shared palettes to the ROM from a file. This file will have previously been obtained from the Extract Palettes from ROM menu command.

File Menu : Export Multiple Levels to Files

This will export multiple levels from the ROM to MWL (Mario World Level) files in a directory you select. The file name you enter will be used along with the level number to create the file names used for each level.

The "**Only export modified levels from the ROM**" option allows you to only export levels that lie within the expanded area of the ROM. Turning it off will cause all 0x200 levels to be exported, which will create 512 small files. So don't turn it off unless you really want to. ;) If you want to stop before it's finished, hit the Escape key.

File Menu : Import Multiple Levels from Files

This will insert all the MWL (Mario World Level) files in a directory to the ROM one after another without user input. Just be careful that each MWL file has been saved with the level number you want it to insert it as, because the program won't ask you for those. If you want Lunar Magic to skip inserting an MWL file in the target directory, just mark the file with the hidden attribute before inserting...Lunar Magic will notify you afterwards how many files were skipped.

If you want to stop before it's finished, hit the Escape key.

The **Clear all existing secondary entrances in ROM before importing** option will erase all entries in the secondary entrance table of the ROM before importing your levels. This is useful if you're transferring all your levels to a new ROM and you don't want the entrances from the original game to clutter up your table.

The **Auto-Set Number of Screens** option will automatically set the number of screens for each level. This overrides the same option found in the General Options dialog. There may be a few times where you will want to turn this off, as a couple of the levels in the original game were a bit lazy about not putting things past the last screen of the level (like the bonus game in level 0 for example).

See the Save Level To ROM As menu description for information on the rest of the options you'll find in this dialog.

File Menu : Analyze Resources in Levels

This will scan through your levels to analyze certain resources used by them, then generate a text file report that will be saved in the same folder as your ROM. It can currently examine the following things:

- Which Map16 tiles are used in which levels. This does not include tiles covered by other tiles or tiles that have conditions for appearing that haven't been met yet. There is also an option to report only on tiles that are defined (they don't have the default appearance of the empty blue blocks) yet not used.
- Which GFX/ExGFX files are loaded in which levels. There is also an option to report only files that are inserted but not loaded.
- Which sprites are used in which levels. The extra bits are shown as the upper part of the sprite number, as custom sprites often use these for implementing completely different sprites.
- Which music tracks are used in which levels.

Note that Map16 tiles and graphics will only be analyzed in levels that Lunar Magic can render (so not boss fight levels).

If you want to stop before it's finished, hit the Escape key.

File Menu : Clear Original Level Data Area

This will cause all levels other than the test level that have not yet been modified to be resaved in the expanded portion of the ROM, then the original level data area in the ROM will be cleared so it can be reused. The usable area is from 0x30480 to 0x3E95F, but there are a few extra sections within this that are still not available for use (mostly for castle/ghost house entrances, Chocolate Island 2, and some areas that weren't used in the original game but are used for LM ASM hacks). These are marked off with RATs so you can avoid them.

Other than the marked off areas, LM does not currently save anything else in this area. So the only reason you'd want to use this menu command is to free the area up for 3rd party ASM hacks.

If you want to stop before it's finished, hit the Escape key.

File Menu : Export Multiple Levels to Image Files

This will export multiple levels from the ROM to image files, either as PNG files or 24-bit bitmap files. The file name you enter will be used along with the level number to create the file names used for each level. The levels will appear as they look in Lunar Magic without zoom, so you should turn off anything you don't want in the view menu beforehand.

The "**Only export modified levels from the ROM**" option allows you to only export levels that are within the expanded area of the ROM. Turning it off will cause all 0x200 levels to be exported, if possible.

The "**Auto-Set Number of Screens**" option will automatically set the number of screens for each level before exporting (the change is not saved to the ROM). Normally you would leave it off to use the existing number of screens already set for each level.

Levels that cannot be rendered will not be exported.

Note that if you choose to save as bitmap files, the resulting files will be somewhere between 300 KB to 11 MB in size.

If you want to stop before it's finished, hit the Escape key.

File Menu : Export Level to Image File

This will export the current level as an image, to either a PNG file or a 24-bit bitmap file. The level will appear as it currently looks in Lunar Magic without zoom, so you should turn off anything you don't want in the view menu beforehand. The program will automatically set the level's number of screens and use it to determine the image dimensions. If you want to override this, you can turn off the "Auto-Set Number of Screens" option and specify the screen count manually in the "Change Properties in Header" dialog.

Levels that cannot be rendered will not be exported.

Note that if you choose to save as a bitmap file, the resulting file will be somewhere between 300 KB to 11 MB in size.

File Menu : Run ROM in Emulator

This menu item will launch your SNES emulator and run the ROM currently loaded in Lunar Magic. Make sure you first save any changes if you want to test them in the ROM.

If you have not yet set up the path to your emulator, you will first get the emulator setup dialog.

File Menu : Setup Emulator

From here you can set up the path to the emulator you wish to use for testing the currently loaded ROM in Lunar Magic. Once you have set this up, you can run the ROM at any time from the Run ROM in Emulator menu.

The **Use Above Custom Command Line Arguments** option is used to enable setting custom **Command Line Arguments** for running the emulator. While not required for most emulators, it can sometimes be useful for setting certain options on the command line. The default setting is "%1", which is just the path and filename of the current ROM enclosed in quotes.

The **Use 8.3 DOS File Name for ROM** option will cause the ROM file name and path supplied to the emulator to use the short 8.3 DOS naming conventions. This should only be used when running ZSNES for DOS (which you probably aren't using unless you're on an older OS like Windows 95/98/ME).

The settings in this dialog are saved to the registry.

File Menu : Run/Stop Internal Emulator

This menu item will allow you to simulate playing the currently loaded level directly in Lunar Magic using a DLL provided by Alcaro (downloaded separately - the DLL must be 32 bit if using 32 bit version of LM, or 64 bit if using 64 bit LM). You must make sure to place the DLL and all related files within the same folder as Lunar Magic's exe before you'll be able to use it.

For best results, you should change Lunar Magic's Animation Rate setting to 60 fps. Loading and running the emulator may require a fair amount of RAM and processing power, so if you find your computer can't handle it then you may be better off using a normal external emulator instead.

It's possible to edit the objects in the level as you play it. For sprite editing mode, the emulator will be paused while you edit the sprites... once you leave this mode the level's sprites will be reloaded. If you make changes to the level other than object and sprite placement, you will generally have to save the level then restart the emulator for the changes to take effect.

Note that while the emulator is running, its keyboard shortcuts take precedence over LM's own keyboard shortcuts. Also, the level will scroll by itself to keep Mario on the screen... if you need to scroll elsewhere to edit, you can pause the emulator to permit user scrolling.

File Menu : Pause Internal Emulator

This will pause or unpause the emulator in Lunar Magic. You may find this more useful than the game's built-in pause function, as pausing the emulator will allow you to scroll to other locations in the level since normally scrolling will occur automatically to keep Mario visible.

File Menu : Mute Internal Emulator

This will allow you to mute or unmute sound for the emulator.

File Menu : Swap F4/Ctrl+F4

Normally F4 is used to run an external emulator, while Ctrl+F4 is used to run the internal emulator. When this menu option is on, it will switch the function of these two keyboard shortcuts.

File Menu : Expand ROM

This will allow you to expand your ROM to 2, 3, or 4 MB. However, the program already expands the ROM when needed, so you don't have to use these menu commands unless you want to expand the ROM for other data you intend to insert yourself.

It will also allow you to expand a SA-1 ROM to 6 or 8 MB. See here for more information on large SA-1 ROMs.

File Menu : Create a Full Restore Point

This will allow you to manually create a full restore point with any description you specify.

See Restore ROM to previous state for more information about restore points.

File Menu : Restore ROM to Previous State

Lunar Magic contains a restore system that tracks changes made to the ROM and records them. This allows a user to revert to previous versions of the ROM in this dialog in case something goes wrong, they change their mind about a recent change, or they want to retrieve something that only existed in a prior version of their hack.

To do this, Lunar Magic creates a "sysLMRestore" folder in the same directory as the ROM which contains a copy of the original unmodified ROM to compare against. This is also where it keeps your restore points, stored in the file ROMFileName.lrp.

Note that even if your ROM is no longer possible to open in Lunar Magic, you can still restore to a previous state. You can access the restore list by either replacing your current ROM with any other SMW ROM that can be opened and giving it the same filename, or by opening some other SMW ROM and hitting the **Browse** button to locate and open the ROMFileName.lrp file that corresponds with your hack.

Incremental Restore Points contain only changes done by Lunar Magic for a particular save (which take up less space in the restore file), while *Full Restore Points* contain all changes done to the ROM. No matter which point you restore to however, you'll end up with a copy of your ROM as it existed at that point in time, with the changes in the description applied.

Just remember, **restoring to a previous version of your ROM will overwrite the ROM that is currently open in Lunar Magic!!** (and the auxiliary files if selected) If you want to keep your current ROM, either make a backup of it or create a restore point for it.

See the Restore options to control how restore points are created.

File Menu : Create IPS File for this ROM

This allows you to create an IPS patch for your ROM, which you can use to distribute your changes to the game. Lunar Magic uses the restore system's copy of the original unmodified ROM to compare against to create the patch.

For another person to use your patch, they can apply it to a copy of the original unmodified ROM using either Lunar Magic's "Apply IPS" menu (which will check if the ROM has been modified or not), or download a separate IPS patching program like Lunar IPS (which you can obtain from the author's site).

File Menu : Apply IPS File to this ROM

You can use this to apply an IPS patch to your ROM. Note that if the patch to be applied is intended to be a full game hack, it must be used on an **unmodified** copy of the original ROM.

File Menu : Scan ROM

This will scan the user area of the ROM and give you various bits of information, such as how much free space is left, whether there are conflicted RATs, and the last version of Lunar Magic that was used on the ROM.

The **RAT Protected Space** is all the used space in the user area of the ROM that is currently protected by RATs. This will contain your levels, overworld, Map16, ASM, and all kinds of other data.

The **Unprotected Map16** value should normally be 0 in any modern hack. In very old versions of Lunar Magic (1.63 and earlier), Map16 was not protected by RATs. While LM was always able to keep track of where it was and automatically avoid it, 3rd party tools often didn't. So if you intend to work on an old hack that has unprotected Map16 using newer tools, you should first resave the Map16 in a newer version of LM to avoid issues.

The **Unprotected Used Space** value is all the space with non-zero bytes in the user area that is not protected by RATs. This will normally be 0 in a well maintained hack. While LM tends to avoid overwriting unprotected used space, 3rd party tools might not, and it's generally just safer to have data protected by RATs.

The **Unusable Space** value keeps track of how much space is technically free but can't really be used at the moment as it's in an area too small to hold the minimum size of a RAT plus 1 byte. In other words, it records how many bytes belong to free space gaps of 8 or fewer bytes. This will normally either be 0 or a fairly small number, and can change over time as data gets moved around.

The **Free Space** value is all the empty space in the user area that hasn't been used yet and is currently available to hold new data. If this value starts to get low, it means you have less space left for adding things and the ROM may need to be expanded soon (if possible).

The **Total User Space** value is all the space, used or empty, that is scanned through to find space to store things. This will usually be the size of the expanded ROM minus the size of the original ROM.

The **Conflicting RATs** value records how many nested RATs were found, which is what happens when you have more than one RAT trying to protect the same area. This should normally be 0 as nested RATs are not allowed. If one or more shows up, this is often an indication that something has gone wrong with a 3rd party tool. You should consider restoring your ROM to a previous state from before the nested RAT appeared to avoid potential issues.

If conflicted RATs are detected in the scan, their positions will be recorded in the RATS.log file.

The **Conflicting Space** value is just the area covered by the conflicted RATs value above.

The **RAT Structures** value is the number of RATs that were detected in the user area. This number will tend to grow over time depending on the number of new things you save to your ROM.

The **Largest Free 32KB Bank** value is the largest free area detected within a 32KB SNES bank that can be protected with a RAT (the size of the RAT itself is already subtracted from this value). So if it shows 8000, this means a full 32KB bank is available for data that can have a RAT stored just outside the bank to protect it.

The **Largest Free Area** is just the largest area that is free when ignoring bank boundaries. Most data in SMW cannot cross LoROM bank boundaries, but compressed graphics can.

The **Last version of Lunar Magic used** is exactly what it says. This will report the version number of Lunar Magic that was last used to modify the ROM. For older hacks this can be useful to know in case you want to use the same version of Lunar Magic that was used to create the hack (which may be needed in some cases if they used 3rd party ASM hacks that are incompatible with newer versions of LM).

File Menu : Recent Files

This contains a list of the last 10 ROM or MWL files you've used. If you open an MWL file without first opening a ROM, the program will attempt to open the most recently used ROM and then open the MWL file with it.

File Menu : Exit

This will close the program and all open editor windows, including the overworld editor. All unsaved changes will be lost.

Edit Menu : Edit Level Layer 1 Mode

This is the default mode that Lunar Magic starts in when you first load the program. This mode allows you to select and edit the layout of objects on the main level layer.

Edit Menu : Edit Level Layer 2 Mode

This mode allows you to select and edit the layout of objects on layer 2. Only certain types of levels actually have a second level layer in place of a BG image, which depends on the " SNES Registers and Level Modes" menu setting.

Edit Menu : Edit Sprites Mode

In this mode, you can select and edit the layout of sprites in the level instead of the objects.

Edit Menu : Undo

This will undo the last object/sprite change. To change the max number of undo operations allowed, check the General Options dialog.

Edit Menu : Redo

This will redo the last object/sprite change. To change the max number of redo operations allowed, check the General Options dialog.

Edit Menu : Cut

This is a clipboard function that will put the selected objects or sprites onto the clipboard, then delete them from the level.

Edit Menu : Copy

This is a clipboard function that will copy the selected objects or sprites onto the clipboard.

Edit Menu : Paste

This is a clipboard function that will paste the objects or sprites on the clipboard into the current level. Since the clipboard functions use the standard Windows Clipboard, it's possible to copy and paste objects or sprites between multiple instances of Lunar Magic.

Edit Menu : Delete

This will delete any selected objects or sprites from the level.

Edit Menu : Delete All

This will delete all objects, sprites, screen exits and secondary entrances from the level.

Edit Menu : Select All

This will select all sprites or objects of the layer you're editing, depending on which mode you're in.

Edit Menu : Insert Manual

This will let you insert a single object or sprite into the level by specifying the exact object/sprite code and data required. But since you can find all the valid objects in the "Add Objects Window" and all the commonly used sprites in the "Add Sprites Window", you'll probably only use this for inserting custom sprites.

You'll get a different dialog for this menu depending on the editing mode you're currently in.

Layer 1 or 2 Object Mode

The **command** specifies which object number you want. The **screen number** determines which screen the object should be placed on. The **XY Position** byte determines the X and Y coordinates for the object (the upper four bits are for Y, and the lower four are for X). The **size/type/ext** byte will change the object size, type, or extended object number (what this byte does tends to be object-specific).

There's also a field called **Extension**, which is used by objects that can have additional bytes. You can enter multiple 2-digit values in this field, but it will only be used if the data size for that particular object allows for it. In practice it's only used for object 2D, which was set aside as a 5 byte object for user defined purposes (do not use this object unless you've inserted code into the game for it, or in-game level parsing errors will occur)

Sprite Mode

This requires mostly identical information to the object data, but there is another field called **Extra Bits**. This should normally be left as 0 since only a couple sprites in the original game actually use this field for something (the secret goal point tape, for example). It's more often used in custom sprites inserted by 3rd party tools.

The **Extension** field is used by sprites that can have additional bytes. You can enter multiple 2-digit values in this field, but it will only be used if the data size for that particular sprite allows for it. Generally only custom sprites inserted by 3rd party tools can use this.

Edit Menu : Edit Manual

This will let you edit a single selected object/sprite manually. See the "Insert Manual" section for more information, as the fields here are pretty much the same.

Edit Menu : Bring to Front

This will bring all currently selected objects or sprites above any other objects or sprites visible at that location, by increasing their Z order if possible.

You can also do this by moving the mouse wheel towards you while holding shift.

Note that the Z order referred to here is really the order that objects or sprites are created in within the level. It's not a reference to actual SNES layers.

Edit Menu : Send to Back

This will send all currently selected objects or sprites below any other objects or sprites visible at that location, by decreasing their Z order if possible.

You can also do this by moving the mouse wheel away from you while holding shift.

Edit Menu : Bring Forward

This will increase the Z order of all currently selected objects or sprites for that screen by at least one, if possible. Increasing the Z order of something will make it appear “above” other items.

You can also do this by moving the mouse wheel towards you.

Edit Menu : Send Backward

This will decrease the Z order of all currently selected objects or sprites for that screen by at least one, if possible. Decreasing the Z order of something will make it appear “below” other items.

You can also do this by moving the mouse wheel away from you.

Edit Menu : Increase/Decrease Object Size X/Y

This section is for all the object size menu commands.

Since most objects have X/Y components, it's possible to use these menu commands to increase or decrease the size of selected objects. The term “size” is used rather loosely here though…for example, when you increase the “size” of X for a question block from one to two, you'll just get 2 question blocks side-by-side. (As a side note, the only difference between having a question block object of size X=2 as opposed to having two question block objects each of size X=1 is that having two objects takes up twice as many bytes in the level data).

Due to the way Nintendo coded a few objects, a size of 0 is sometimes misinterpreted as a size of 256. Lunar Magic emulates this behavior, but you should be aware that any object whose tiles extend way past the first or last level screen could potentially overwrite critical RAM locations.

These menu commands are not applicable to sprites, since sprites do not have a size X/Y component to modify. Note that you can use the mouse to perform the same size actions as these menu commands.

Edit Menu : Conditional Direct Map16

This will bring up a dialog allowing you to change the conditional attributes of all selected Direct Map16 Access objects.

The **Flag number to check** is the flag number from a RAM table of bits at $7FC060-F (bits are ordered from low to high). The values in this table are not initialized in any way... it's up to you to insert the necessary ASM code to initialize and manipulate the bits. If the specified flag in the table is not enabled, the object will not be rendered when the level loads.

However, if the **Always show objects** option is checked, the object will always show up and will just add 0x100 to all Map16 tiles in the object if the flag is enabled. This effectively allows showing 2 different states instead of just having the object be displayed or not.

If the **Remove flag check** option is checked, all selected Direct Map16 Access objects will be converted back to normal Direct Map16 objects (meaning they will no longer be conditional and will always be displayed).

Edit Menu : Remap Direct Map16

This will bring up a dialog allowing you to remap Direct Map16 Access objects that have already been placed into the level based on certain criteria. This can come in useful if you've moved some of your Map16 tiles from their original locations but your level still refers to the tiles in the original spots.

You remap the tiles by entering a list of paired comma separated values in the large text edit in this dialog. The first value can be a single value or a range indicating the source of what you want to change, while the second value describes what to change it into. Each pair of values can be on their own line, or you can just separate them with more commas. There are several examples given in the dialog for the syntax you can use. The simplest is changing one tile into another: 100,25 would change tile 100 into tile 25. Or you can do a range:100-101,25 would change both tiles 100 and 101 into tile 25.

To add or subtract an offset, you must specify the sign to use. 100-101,+25 would change 100 to 125 and 101 to 126. Or 100-101,-25 would change 100 to DB and 101 to DC. But since the intent of adding/subtracting offsets is usually to move a range of tiles to another area, it may be easier to just use 'M' and the starting destination: 100-101,M125 would also change 100 to 125 and 101 to 126. It's also common to move a rectangle of tiles, so you can specify the top left and lower right points of a rectangle using a command like R100-111,M25 to change tiles 100 to 25, 101 to 26, 110 to 35 and 111 to 36.

Keep in mind that for groups of tiles pasted as a single DM16 object, pattern matching will be based on the upper left tile number and will affect all tiles in the group. So for example if you had a DM16 object that consisted of tiles 133 and 134 together and you entered 133,+25, both tiles 133 and 134 would have 25 added to them for that object. But 134,+25 would not affect the object at all.

When entering a list of remap pairs, it's important to remember that the source values always refer to the tiles as they currently exist *before* any remapping occurs, not as they would exist after processing the pairs that come before it. For example, you might think the sequence 100,25,25,100 would be of little use, but it will really cause tiles 100 and 25 to be swapped. This also means in a sequence like 100,25,100,30 the first pair is effectively useless as it's superseded by the second pair.

View Menu : Layer 1 (Level)

This is the main layer that most level objects (ledges, berries, pipes, bushes, etc) are usually placed onto.

View Menu : Layer 2 (Level/Image)

This is a second layer that can be used either as a background image or have level objects placed on it like with layer 1. This layer can also scroll independently of the first layer with certain commands placed into the sprite data.

View Menu : Layer 3 (Image)

This is a third layer that can have an image on it. On the SNES the tiles on this layer can't use as many colors as the first 2. But it can sometimes be useful for simple backgrounds when you want to use layer 2 as a level instead of an image. Or it can be used for effects like translucent mist or small goldfish slowly going by.

Normally Mario can't interact with this layer, however the layer 3 tides are an exception. But for that it reuses the level map in RAM for layer 2, which means you can't interact with both layer 2 and 3 in a level.

In SMW part of layer 3 is also used for the status bar at the top of the screen.

View Menu : Sprites

All the sprites are displayed above all other layers (this behavior is only in Lunar Magic, not in the game). Any sprite which is not implemented or doesn't have any visual appearance will appear as a large "X", unless of course you modified the "X" in GFX file 0, in which case it'll look like whatever you put.

View Menu : Sprite Data (Hex code)

This will display a small "x" on top of every sprite, followed by a one-digit number indicating the value of the extra info field of the sprite command (usually 0). Underneath that will be a two digit hex number, which represents the sprite command number itself. Generally you'd only use this to find out the command number of a particular sprite on the screen.

This option is off by default, and it will also not display anything regardless of the setting if the sprite viewing option is off.

View Menu : Screen Exits

This allows you to view the exit destination of any particular "screen", which will be used for any door or exit pipe within that screen. Each screen will be labeled with the screen number, in addition to specifying the level number of the destination. Any screen that has an outline but isn't highlighted has no exit destination set and leads to an undefined level.

Note that screen exits are not always the same type of screen referenced by objects/sprites.

View Menu : Sub Screen Boundaries

This creates an outline around each subscreen, along with a label indicating the object/sprite screen number and subscreen section. Sometimes objects break apart or otherwise don't behave well when placed on a subscreen boundary, so it can be useful to know where these are.

View Menu : Game View Screen

This creates a draggable SNES-sized screen that you can use when designing levels to figure out what will be visible on the screen. It also optionally has highlighted areas for locations Mario will be during scrolling in different directions, to aid in determining where the screen will be during gameplay.

The screen is draggable on an 8x8 grid (Alt-Left Click), and the highlighted bands (Alt-F3) are the regions Mario will generally be during scrolling. The idea here is that if you want to know what you'll be seeing if Mario is at a certain location, just drag the screen over it and align the location with the desired band (the four band intersection points in the middle would be the most commonly used if you have vertical scroll at will). Red is scrolling right, green is scrolling left, blue is climbing up, purple is climbing down. The 2 extra red and green bands on the far sides are for if the player has used the L/R buttons to look further ahead.

It can't be pixel perfect as there's some variance depending on what's happening, but it can give a fair idea of what will be on screen.

View Menu : Exit Enabled Tiles

This will show red outlines for tiles that are exit enabled so you can more easily figure out where there are tiles that use screen exits. The outlines will appear above sprites and other layers.

View Menu : Tile Surface Outlines

This will show outlines for tiles surfaces so you can more easily see slopes and which tiles are solid. A green line can be walked on, green squares are solid from all sides, blue is for slopes with water, red hurts or kills Mario, and yellow only affects sprites other than Mario.

View Menu : Line Guide Outlines

This will show outlines for line guide tiles. This is meant for cases where you want to use line guided tiles yet don't have the right graphics loaded to see them. Note that you need to use a separate patch if you plan to have another tile act as a line guided tile.

View Menu : Tile Grid

This creates an grid allowing you to more easily see how tiles are placed. If you want to change the grid color, hit control+alt+F8.

View Menu : Zoom

This will cause the level editor window to display the level zoomed at the specified percent.

Note that some zoom levels and the filtering option may not be available if DirectX is not being used.

View Menu : Level Entrances

This will place a picture of Mario and a label at the main, midway, and secondary entrances to the level. The actual picture of Mario used depends on the action setting for the entrance.

If the midway entrance is at the same location as the main entrance, the main entrance label will simply display a leading ">" character.

A level can only have one main and one midway entrance. Not all levels make use of the midway entrance.

View Menu : Block Contents

Enabling this will allow you to see the contents of question blocks, etc in the editor. However this doesn't affect sprite based blocks (the sprite ones already show their contents in a different way).

This view option is saved to the registry.

View Menu : Animation

Using this menu command will cause Lunar Magic to do continuous animation to simulate the animated tiles in the game. It will also do the palette animation of Yoshi's coin.

This may cause a performance hit depending on how fast your computer is, and how high the Animation Rate has been set. The larger you make Lunar Magic's window and the more editor windows you open, the more CPU power it takes to animate them all. If you notice that it takes a long time for small message boxes to pop up, you may want to keep this off.

This view option is saved to the registry.

View Menu : Next Animated Frame

Using this menu command will advance the animated frames displayed in the editor by 1.

View Menu : Reset Animations

Using this menu command will cause Lunar Magic to reset the animations of the level to their initial state for viewing.

While animations are already reset whenever you edit the animated frames or edit the level's animation settings, there are a few one-shot animation triggers that you may want to be able to reset the effects of at will for viewing purposes. This way you don't have to keep clicking through a dialog to do it.

View Menu : ExAnimation Triggers

This submenu allows you to toggle various triggers added through ExAnimation to preview their effects in Lunar Magic. This includes the "Have Star", "Timer Below 100" and "Have 5 Yoshi Coins" triggers, along with all the Custom, One Shot, and Manual triggers. There are a fair number of the latter 3, so the keyboard shortcuts have been set up so that you first have to select the trigger you want then use another key to activate it.

Remember that a One Shot trigger is designed so that it will automatically turn itself off once the animation using it has finished going through all its frames. Also note that if an animation is set to be initialized to a certain state on level load, this state will be restored every time animations get reset for viewing in Lunar Magic.

View Menu : Silver POW

This menu command can simulate what happens when you hit a Silver POW. A silver POW turns a black piranha plant into a coin, and most sprites into silver coins.

View Menu : POW

This menu command can simulate what happens when you hit a blue POW. A blue POW turns coins into brown blocks, and vice versa. It can also reveal hidden doors, coins, and some question blocks.

View Menu : Special World Passed

When a player passes special world, GFX file 0x31 is loaded over part of the standard sprite graphics. Also, green shells turn yellow and red shells turn blue. Enabling this option will allow you to see what the level will look like in the game when this happens.

View Menu : Invisible POW Objects

In Mario World, some coins, doors, and blocks are only visible when Mario jumps on a POW (P-block). If you enable this option, all POW objects that are normally invisible will appear as translucent objects in the editor. When disabled, all the objects remain invisible (although you can still select and manipulate them).

View Menu : Other Invisible Objects

Enabling this will allow you to see other normally invisible objects, such as question blocks, 1-Up points, etc. They will all appear semi-transparent in the editor.

View Menu : On/Off Switch On

In levels with line tracks that control the movement of platforms, toggling this will act just like the on/off switch block does in the game so you can view a different line track. Note that you can still select and manipulate a track in the editor regardless of whether it's visible or not.

View Menu : Conditional Direct Map16 On

This will simulate having all flags enabled or disabled for conditional Direct Map16 objects.

View Menu : 512 Height Background

This will enable backgrounds in vertical levels to be viewed in the editor so that they tile vertically with a pixel height of 512 instead of the old 432 height that all Nintendo's backgrounds use. This setting only affects the appearance in the editor... in the game, the background will always use the full 512 pixels.

This view option is saved to the registry.

View Menu : Translucent Text and Outlines

This causes the text and outlines used for marking entrances and exits to appear translucent in the editor, which may make it easier to see what you're doing while editing. It does take a bit more processing time though, and may make overlapped text hard to read.

This view option is saved to the registry.

View Menu : Green/Yellow/Red/Blue Switch Blocks

Whenever the viewing option for one of these blocks is turned off, you will only see the outline of the blocks instead of a solid block, to see how it would look in the game when the switch for that color hasn't been activated yet.

Note that this setting does not affect the visibility of the blocks in the actual game. All options you find in the View menu are for design purposes only.

Editors Menu : Add Objects Window

The Add Objects Window is a floating window that contains a compiled list of all the valid objects in the game. The Window has a preview area, along with a drop-down list of object categories and a list box containing the actual object entries. To paste one of these objects into a level, all you have to do is first select the object you want, then right-click somewhere in Lunar Magic's main level editor window while this window is open. Just make sure you don't have some other objects already selected in the level, or you'll end up with a copy of those objects instead (or you can just hold down the control key when you right click to avoid that). And being in Sprite Editing mode while trying to paste objects doesn't help, either.

You can do a text search in the current category by simply starting to type. F3 will find the next match, Ctrl+F3 will find the previous match, and escape will clear the current search. For the Direct Map16 list, this can instead be used to jump to either the page you want or the tile to select (if it doesn't also match a page number).

The **show preview icons in list** button lets you see a small preview of the object beside each list entry. Next is the **"hide objects"** button... this button will hide objects that do not have the right GFX files in the last 2 FG/BG GFX slots or the right FG/BG GFX index set. If you are using ExGFX files, you may want to turn this off so you can see all available objects. The **vertical layout** button gives you an alternate layout of the items in the window that's more suitable for resizing vertically. Beside it is the **zoom** button, which allows viewing the preview area at different sizes. Last is the **view preview area** button, which can turn on or off the preview area for viewing the current selection.

Now, on to the object categories. "**All Objects**" contains the objects from all the other lists put together, excluding the Direct Map16 category. "**Standard Objects**" are available in all tile sets, and usually have size X/Y components. These objects can easily be copied from one level to another, assuming the correct graphics are loaded. The entries present in the "**Tile set Specific Objects**" category will change depending on which tile set the current level has loaded, so be wary of using these if you plan to switch the tile set at some point. Since these objects do not exist in certain other sets, copying them from one level to another may not work. They do often have an X/Y component though. "**Extended Objects**" are available to all levels, but since they don't have an X/Y component they're not quite as flexible as standard objects.

The "**Direct Map16 Access**" category is an ASM enhancement added to Mario World by Lunar Magic that allows you to select any of the 16x16 (Map16) tiles in the current tile set, paste them into the level, and manipulate them like an object. If you select and paste a group of tiles then resize the object, it will repeat the tile pattern of the group you originally pasted. To save on ROM space, it's best to only use this to access tiles that aren't available in Nintendo's collection of pre-made objects. You should also paste groups of tiles where possible and/or use the resizing feature rather than placing tiles one by one. You can also fill an enclosed area of the level with Direct Map16 tiles with control + shift + right click.

Be aware that some objects, although they may technically be available in the list, will look like garbage if the tile set doesn't have the particular GFX tiles or Map16 tile data in question to create the object.

In the description of the objects, the number in front of each one is the command number used to create the object. A blank spot in place of the number means that the object uses the same object number as the one above it, usually made possible by using part of the X/Y size bit fields as a type indicator. The X/Y/SS size indicators let you know if the object has an X component, a Y component, or SS component (SS means X and Y control the Same Size component). Extended Objects will never have any X/Y/SS indicators.

A phrase you may see now and then is "exit enabled", with reference to pipes. Some pipes cannot be used as exits, but the ones that can will be labeled as "exit enabled".

The term "Star 2" is used to indicate that a block normally has a coin in it unless Mario hits it while he already has a star on, in which case it becomes a star instead of a coin.

Another bit of information you may see is "X/2", "X/3", "X/4" and so on, along with a list of potential items that the block may contain (for example, a coin/1-UP/vine inside). The "X/2" part indicates that what the block has in it depends on the X position of the object MOD 2 for that particular screen. However you don't have to figure this out manually, just wave the mouse pointer over the block in your level and either look at the bottom left status bar to find out what's in it at that location or check its tooltip. Or you can just turn on viewing block contents in the view menu. If it's not what you want, just move the object horizontally a space or two till you get it right.

Editors Menu : Add Sprites Window

Since the controls and operation of this window is mostly the same as the "Add Objects Window", I'll skip repeating it here.

Now, "**All Sprites**" contains the sprites from all the other lists put together. The "**Standard Sprites**" category has the sprites that are available in all sprite tile sets (because in all sprite tile sets, SP1=0 and SP2=1). The "**Tile set Specific Sprites**" depend on the GFX loaded by SP3 and SP4. Unlike "Tile set Specific Objects" however, the contents of this list won't change so these sprites are always technically available, but most of them will still appear corrupt without the correct tile set loaded. The **“Special Commands and Generators”** category contains commands that can control things like the scrolling of layer 2, as well as sprite generators. The **“Custom Collections of Sprites”** section is mainly intended to hold custom sprites (see the Custom Collections of Sprites topic for more information).

To help in finding which graphics a sprite uses, a small line of text will sometimes appear at the top of the preview window to suggest which GFX files could be loaded into SP3 and/or SP4 to view the sprite correctly.

In the sprite descriptions, you may sometimes come across sprites that have lists such as "Goomba/bomb/fish/mushroom", ending with something like "(X&3)". This means that the sprite behavior or appearance will alter depending on the position of X in the current "screen" bitwise-and 3. In these cases, the editor will often either display the sprite differently for each position or give more information in the sprite's tooltip.

I've taken the liberty of marking sprite "generators" with a small star along with a picture of which sprite the generator produces. They activate once Mario reaches the screen number the generator is on, and typically continue until Mario enters a screen with a "Turn off Generators" command (0xD9). However, the generators that appear after this command in the list cannot be turned off (such as the Ghost Ceiling, for example).

A few of the sprite commands will sometimes display a warning that the sprite “may glitch”. This is because most of the layer 2 commands must be placed at the top of a subscreen (subscreen coordinate Y=0 or 1), otherwise they could crash the ROM or create unexpected results. If the sprite command is placed in the correct position on the screen, the glitch warning will be replaced with a description of what the command does at that location. Several also require that layer 2 (BG) vertical and/or horizontal scrolling be set to “none”, and the “auto-scrolling” type commands may also require that layer 1 (FG) scrolling be set to “none”.

As you probably already know, Mario World only allows a certain number of sprites on the screen at once. Exceeding this number will result in the additional sprites not being rendered until after Mario has killed the existing sprites, left the screen, and come back. This number is determined by the sprite memory setting in the sprite header menu. Be especially careful around a Goal Point sprite...you don't want THAT to not show up because of too many sprites on the screen at once!

You may encounter a problem at some point where sprites look just fine in Lunar Magic, but look mangled or (in rare cases) won't even show up in the actual game. Chances are that either you have sprites that are incompatible with one another, or it has something to do with the settings in the sprite header. My advice would be to look up the sprite memory setting in a level that uses the sprite in question and try to use that. If that doesn't work, then it's a sprite incompatibility problem and you'll have to isolate and remove the offending sprite(s) from the level.

As an example of sprite incompatibility, in level 12C sprite 62 (the brown line-guided platform) cannot appear on the screen at the same time as sprite 65 (the chainsaw). If a chainsaw happens to be on the screen when Mario reaches a brown platform, the platform won't show up, and vice versa.

And an example of an incorrect sprite memory setting would be going to level 105 and selecting any other memory setting other than 4 or D, which are designed for use with sprite 9F (Banzai Bill).

Of particular note is that Yoshi himself is not compatible with a few special sprites (like the vertical fireballs in castles, and baby Yoshi) because Nintendo used the same memory location to store their graphics. This is probably why Nintendo never allowed Yoshi inside castles, and why Yoshi eggs hatch into a 1-UP if Yoshi is already present.

You may notice that sprites in the main Lunar Magic Window near the bottom/top of a "screen" will bleed off into the top/bottom of the next/previous screen. While this behavior is correct for objects, it is not correct for sprites, and should be ignored.

Do not use a key and keyhole or a Secret Exit Goal Point sprite in a level that you have not set up a secret path for on the overworld. That usually just results in passing an event that was already assigned to a different level and possibly enabling a direction from the current level tile that you didn't set up a walking path for. Setting these up is the job of the overworld map editor.

Editors Menu : Palette Editor Window

This is where you can actually edit the colors used by a level. The SNES uses a 256 color table, and the colors themselves are 15 bit. In the mode that SMW usually uses for levels, this table is further divided into 16 palettes with 16 colors each. The first half of the table (palettes 0-7) is used for FG/BG tile graphics like objects and background images, and the second half (palettes 8-15) is used by sprites. The first color in each palette is used as the transparent index, so there's really 15 colors you can use per tile.

Each palette in this editor is displayed as a horizontal row of colored squares. The colors can be edited simply by **left-clicking** on the color you wish to change. You can then use the standard Windows color editing dialog to select or enter a PC 24-bit RGB value, and Lunar Magic will automatically convert it to the nearest valid SNES 15-bit color that matches it. However, if you just want to quickly copy an existing color without going through the color dialog, you can hold down the **control key and left click** on the color you want to copy (which also places it on the windows clipboard as text). Then just **control-right click** wherever you want to paste the color. To copy or paste an entire row, hold down **alt** as well.

You can also create a gradient, simply by holding down **alt and right-clicking** on the first color and then the last color to use. All colors between them will then be changed to create the gradient. You cannot use color 0 in any of the palettes for this, as that's the transparent index (the editor will automatically choose color 1 if you do that).

If you want to copy the color of any pixel on your computer screen to the windows clipboard, move your mouse over it and hit **F3**. This can be useful for copying colors being displayed in other programs. Note however that the palette editor must have the window focus, and it may not work on all windows or protected media.

The **Back Area Color** is an extra color that is shown behind all other scenery, and can be edited the same way as the other colors.

Note that changes caused by palette animation are temporary and will not affect the level's original palette. In fact palette animation changes may be reverted as you edit colors. If you find this distracting, you can always turn of animation while you work.

Most of the buttons in this editor are fairly self-explanatory. The first one discards all changes made to the palette and palette settings since the level has been loaded (applies to both shared and custom palettes). The next two are standard undo and redo buttons (for RGB changes only).

After that is the **Palette Mask Edit Mode** button. Palette masks provide a method for palette files to specify which colors in the palette should be imported into your level. In this mode, you can mark which colors should be enabled or disabled for export with a **Left Click** (add **Alt** for an entire row). An X means the color is disabled, otherwise the color is enabled. When you export the palette to file in any format, all the colors are technically still exported regardless of the settings... but if any colors were disabled the settings are saved to a separate .palmask file that goes with the palette file of the same name. When you later import the palette file, the program will check for the presence of the .palmask file. If it exists, it will load the settings in it and only import the colors that were enabled into your current level's palette. If it doesn't exist, all the colors are imported.

Note that palette masks are automatically cleared whenever a new level is loaded. And the only way to save them is to export a palette. You can use the **Palette Mask Menu** button to quickly enable or disable all colors for export.

The next two buttons allow you to export or import the current level's palette to file (if import is used, it will automatically enable the custom palette setting).

The last 4 buttons deal exclusively with shared palettes. The first two buttons export or import all shared palettes in the game to file (palette mask files do not apply to these). The next one discards all changes made to the shared palettes since they've been saved, and the last one saves the shared palettes to ROM. For more information on shared palettes, see below.

The **Auto-Enable custom palette on edit** option will automatically enable the custom palette option for the level when you edit any colors.

Shared Palettes (custom palette disabled)

If the Custom Palette setting is disabled, the level is using the standard shared palettes of the original ROM. This means that if you change a color value, any other level or overworld submap that uses that particular color will also use the new value. There are also a few colors that are hard-coded by ASM, and Lunar Magic will not allow you to alter them (these are mainly the black and white colors). Whenever you alter an RGB value, 2 buttons at the top will be enabled which allow you to discard the changes or save them to the ROM. The modifications you make to the RGB values will be used by Lunar Magic immediately, however they will not be saved to the ROM until you hit the save button in the palette editor. Unsaved RGB changes to shared palettes will not be lost by switching levels. This can be fairly useful, since you can flip between levels to view how they look before saving the changes. The changes will only be lost if you reload the ROM, close Lunar Magic, or hit a button to discard the changes.

With the Custom Palette setting disabled, you also have the option of using the drop-down lists to select different colors for certain sections of the palette. This doesn't change any actual color data in the ROM, it just modifies part of the level header to select which shared palettes to load up. The available settings are for changing the **Back Area color**, **BG palette**, **FG palette**, and **Sprite palette**. The **Mario palette** setting is only provided to give you access to the player colors so that they can be modified...the setting itself doesn't actually affect anything in the game or level.

Warning: The shared palettes can be edited in both the level editor and overworld editor simultaneously. This means the save button in the level palette editor saves the changes in the overworld palette editor and vice versa. But undo/redo and discarding unsaved changes when editing shared palettes will also affect both, sometimes in ways you may not anticipate, so be careful.

Custom Palettes (custom palette enabled)

If the Custom Palette setting is enabled it means the level will use its own custom palette, completely separate from all the other palettes in the ROM, which is made possible by another ASM modification that Lunar Magic installs. Changing a color in the palette will have no effect on any other level in the game.

Although you can now technically edit any color of any palette, a few colors are overwritten by the game at run-time so changing them doesn't serve any purpose. **Palette 6 Color 4** is used by Yoshi coins for palette animation (unless you turn it off in the animation settings), and **Palette 8 Colors 6-15** are overwritten with the current player colors (to modify the player colors, temporarily switch to any level that uses shared palettes). Other than those, you can modify the colors to your heart's content.

The drop-down lists for the header settings will be disabled as they no longer affect the level, and the palette is now saved along with the level itself. Just remember that this means the RGB changes will be lost if you switch levels before saving, unlike with shared palettes.

If you have a custom palette for the intro level (0xC7), you should be aware that the game overwrites **Palette 0 Colors 8-15** and **Palette 1 Colors 8-15** with the colors used in the Mario World title screen. But Lunar Magic only loads these colors for editing if level 0xC7 is not using a custom palette! To edit the title screen colors after you've already switched to a custom palette for this level, a possible work-around is to save a copy of the original level from another ROM to a MWL file. Then go back to your current ROM and load the MWL file to temporarily access those colors. When you've finished editing the colors, simply save the RGBs without saving the MWL file you loaded.

General Palette Information

At the end of a level when Mario hits a goal tape, most colors are dimmed to create a fading effect. Colors 9,A,B of palette 0 and colors 9,A,B,D,E,F of palette 1 are not affected, as they're used in the status bar.

In general, this is how Nintendo organized the colors and palettes in their levels, which may be useful to know:

Color 0 of all palettes is a transparent index. While changing this color in a level should not have any effect on the level, the color may not be overwritten properly when the overworld reloads, which has been known to cause a strange flashing effect during path revealing. So it's best to leave these as black.

Colors 1-7 of palettes 0 and 1 are used in the backgrounds.
Colors 9-B and D-F of palettes 0 and 1 are used for the time, lives, score etc status screen at the top of the layer 3 screen.
Colors 8 and C of palettes 0 and 1 are not used (technically they're for the transparent index of the 2bpp layer 3 tiles, so they can be reused for other things).
Colors 1-7 of palettes 2 and 3 are used for foreground tiles, so this will change from level to level for different FG tile sets.
Color 8 of palettes 2, 3, and 4 are not used.
Colors 9-F of palettes 2, 3, and 4 are used for the FG berries that Yoshi can eat.
Colors 1-7 of palettes 5, 6, and 7 tend to be used for static FG objects such as pipes and ice blocks.
Colors 8-F of palettes 5, 6, and 7 are not used. The colors in this location are left there from the overworld, I believe.

Palette 8 is used for the current player and power up colors (Mario/Luigi/Fire/etc). Colors **1-5** also get used for certain gold colored sprites, while **colors 6-F** get overwritten with the player colors even if using a custom palette.
Colors 1-7 of palette 9 are used for gray sprites.
Color 8 of palettes 9, A, and B are not used.

Colors 9-F of palettes 9, A, and B are used for the sprite berries that Yoshi eats. Basically, when he eats a berry the FG tile actually gets replaced with a sprite berry which is used in the eating animation. Thus it needs 2 copies of the berry colors as one has to exist in the FG palettes and the other in the sprite palettes.
Colors 1-7 of palettes A, B, C, and D are used for the Koopa sprites as well as several others.
Colors 1-7 of palettes E and F are used for tile set specific sprites, so this will change from level to level.
Colors 8-F of palettes C, D, E, and F are not used.

Editors Menu : 8x8 Tile Editor Window

This is a fairly basic tile editor, allowing you to modify the raw GFX currently loaded by the level. This editor can be a handy tool for quick modifications or just to see which graphics are actually loaded by the level. For extensive graphics modifications however, I highly recommend using one of the many fully featured SNES tile editor programs available on the net to edit the *.bin GFX file(s) extracted by Lunar Magic.

Before you start editing graphics with this, make **sure** that you've used the “Extract GFX from ROM” menu command first! Otherwise, you might accidentally erase your GFX work when you're forced to use this menu command later.

Most of the functions in this editor are keyboard and mouse driven. The **Arrow Up/Down** keys control scrolling. A **right click** will paste the currently selected tile to the mouse position, while a **left click** will select a tile. A **control left click** will copy a tile to the windows clipboard, and a **control right click** will paste it. The **Page Up/Down** keys will change the current palette number being used to view the GFX. The palettes are taken directly from the palettes loaded by the current level.

You can use the **shift key** along with the **arrow keys** to shift the currently selected 8x8 tile up/down/left/right by one line of pixels.

The first three pages of graphics are for the 6 FG/BG graphic slots (each “slot” has 0x80 8x8 tiles, and is typically stored as a separate file when you extract the graphics from the ROM). These are ordered according to how they're stored in SNES VRAM, and are referred to as FG1, FG2, BG1, FG3, BG2 and BG3 in other dialogs of the program. BG2 and BG3 do not exist in the original game and are added by an ASM hack.

The fourth page is normally blank and not available for use.

The next two pages are for sprite graphics. These 4 slots are referred to as SP1, SP2, SP3, and SP4.

When you select an 8x8 tile, it will be displayed below the current page at 16x the normal size (the other tiles in the page are displayed at 2x their normal size). Left or right clicking on one of the “pixels” of the 8x8 tile will replace the pixel with the currently selected FG or BG color. A color bar will be shown below the tile with the 16 colors of the current palette on it, and the FG and BG colors currently selected will be displayed above the left side of the color bar. Left or right clicking on one of the colors in the color bar will change which FG or BG color is currently selected. You can also use a **control left or right click** on a pixel in the tile itself to select a color.

The actual colors used for the 8x8 tile will depend on the palette colors loaded by the level and the palette number specified for the 8x8 tile within a 16x16 tile. But if you've worked with SNES graphics before, you probably already know this.

To the left of the selected 8x8 tile are several buttons that are already possible to access through other keyboard shortcuts. To the right of the tile are buttons for **Flip X/Y** and **Rotate 90**°. The next two buttons of **Map colors** and **Do Map** are for creating and using a color number filter map on the tile, which can be useful for adjusting new or imported graphics to use the color numbers you want in a palette. To access the dialog to create a map, hit the "Map colors" button. In this dialog, you can select a filter number from 0-15 to use (the filters are all saved to the registry). By default, the colors in all the filters are mapped to themselves. So color 0 becomes color 0, color 1 becomes color 1, etc. You can reset to these default values at any time by using the **reset button**. To change the color that a particular color is mapped to, select the original color in the leftmost list box and select the color to map it to in the list box on the right. When you're finished creating the filter, just hit OK. Now when you hit the "Do Map" button, the changes you selected for the filter will be applied to the currently selected tile.

The **F9** key will save the currently loaded graphics to files in the /graphics folder (or the /ExGraphics folder depending on what it is you're editing) to be inserted later with the “Insert GFX to ROM” menu command. Although the modifications you make to the graphics in this editor will show up in the main Lunar Magic level editor window immediately when the display is refreshed (hit **F1** in the 8x8 editor to force an update), unsaved changes will be lost once you switch levels. Also, since Lunar Magic uses the graphics from the actual ROM to render levels, the changes you save to the graphics files on your hard drive will not be reloaded by Lunar Magic for level rendering once you switch levels. You must insert the GFX files to the ROM if you want Lunar Magic (and the actual game) to use them on a permanent basis.

One of the limitations of this tile editor is that you cannot paste over the animated tile area (tiles 0x40-0x81, 0x90, 0x91, 0xDA-0xDD, 0xEA-0xED), because these tiles are actually being copied in from other GFX areas. And if the "Special World Passed" view menu item is enabled, you cannot edit or save sprite tiles 0x480-0x4FF.

Editors Menu : 16x16 Tile Map Editor Window

Levels in SMW are made up of objects, but these objects are built out of 16x16 tiles. And these 16x16 tiles are themselves built out of four 8x8 GFX tiles along with information specifying the desired palette, x/y flipping and layer priority for each 8x8. This editor will allow you to modify the data responsible for mapping this out.

Must like the rest of the program, a **right click** will paste the currently selected tiles to the mouse position, while **left click** can be used to select multiple tiles or move selected tiles. A **double left click** selects an 8x8 tile into the 8x8 editor and selector windows, and a **control + right click** will paste tiles from the 8x8 tile selector window. Standard windows clipboard keyboard shortcuts are supported (**Control-C** to copy, **Control-X** to cut, **Control-V** to paste). You can use **Control-Z** to undo, and **Control-Y** to redo. The **ESC** key deselects any selected tiles, and the **Delete** key will delete selected tiles. **F9** will save the Map16 data directly back to the ROM. **F1** will toggle the page view. **F2** exports selected tiles to file, and **F3** loads tile from file. **F8** will toggle the grid view, while **Control+F8** will toggle the grid color.

It's also possible to paste FG tiles from this editor directly into the level as Direct Map16 tiles by using a **control + alt + right click** in the level editor. Add the **shift** key to do a fill operation instead.

Map16 data is shared between levels. It is not part of any specific level and thus isn't stored with MWL files, nor is it automatically saved when you save a level to the ROM. You must save the Map16 in this editor if you want to keep changes to it. If you switch levels without saving, changes made to tileset-specific tiles (pages 0-1, and page 2 if set to tileset specific) will be lost but the rest will remain intact.

To save ROM space LM only saves Map16 up to the last used (non-default blue) block within each group of 0x10 pages. This means it's best to use up the earlier pages of Map16 data first. Although if you want to reserve enough space for all the pages ahead of time for some reason, just store a non-blank tile at the end of every 0x10th page, which would be every 0xFFF tile (so at tile 0xFFF, 0x1FFF, 0x2FFF, etc) and save. Storage of 0x10 pages of Map16 data takes up an entire LoROM bank (0x8000 bytes).

The first 0x80 pages of Map16 data are for the FG (level objects), and the next 0x80 pages are for the BG (background images).

FG Map16 (tiles 0-0x7FFF, pages 0-0x7F)

The first 2 pages (tiles 0-1FF) are from the original game. This area is made up of both static tiles whose Map16 data never changes, and dynamic Map16 tiles that are replaced with other tiles for certain tile sets. For example, the question block tiles remain static throughout the game, while the ghost house tile set will replace a fair number of Map16 tiles with its own tiles for crossbeams, spider webs, etc. This means if you modify the appearance of a question block the change will appear in all levels, yet if you modify a spider web tile the change will only be present in ghost houses and will not affect other tile sets (excluding the "Act as" setting, which is not tileset specific).

The next 0x7E pages (tiles 200-7FFF) were created using an ASM extension that Lunar Magic inserts into the ROM. Since these tiles are not used by the objects in the original game, the only way to place them into a level is to use the "Direct Map16 Access" section of the Add Objects editor window.

Page 2 (tiles 200-2FF) can optionally be set to be tileset specific, but this is an older feature that is off by default in new ROMs. You can change this setting using **Control + F1** (just remember the "Act as" settings remain the same across all tilesets!).

BG Map16 (tiles 0x8000-0xFFFF, pages 0x80-0xFF)

The first two pages (tiles 8000-81FF) are for the original data that Nintendo used, and the remaining 0x7E pages (tiles 8200-FFFF) are from an expansion created by Lunar Magic. These can be placed into the background of a level by selecting them and using the Background Tile Map Editor Window to paste them. The "Act As" setting does not apply to background tiles.

The editor's buttons all have tooltips, which makes them fairly easy to figure out. The reload button discards all changes and loads from the ROM. The next two are standard undo and redo buttons.

After that are the **8x8 editing mode** and the **8x8 tile selector** buttons. While normally you can just edit the 16x16 tiles by manually typing in the 8x8 tile numbers to use, it can be much faster to drop down into 8x8 editing mode so you can copy and paste these tiles directly. And that's where the 8x8 tile selector comes in: you can select multiple tiles from this window and just paste them into the Map16 editor using a Control-right click in 8x8 editing mode. Or you can use the standard windows clipboard keyboard shortcuts (which also places the tiles on the windows clipboard as text). If you want to change the palette used in the 8x8 tile selector, you can use Page Up/Down. In 8x8 mode, the "Act as" settings for the 16x16 tiles are not affected.

Next is the **zoom** button, which allows viewing the tiles at different sizes. Beside it is the **pipe** button. SMW has an unusual way of controlling pipe colors... the main vertical pipes starting at tile 0x133 have 4 different sets of Map16 tiles that can be swapped out, and which set is used depends on the screen number the tile occupies in the level. You can use this pipe button to swap out which ones are currently displayed in the Map16 editor so that you can edit them all.

After this is the Select Tiles button, which can automatically select all FG or BG tiles for you (which is only really useful if you plan to export tiles in the older .bin format used in prior versions of Lunar Magic). The next button with a red X is the **Auto-Deselect on Editor Select** button. If this option is enabled, whenever you make a new selection in the 8x8 Tile Selector window, anything selected in this window will be deselected. This can make it easier for pasting, since then you don't have to always remember to hold down the control key. The red outline button after it is to enable or disable importing Map16 tiles from files to the same location they were imported from.

The **Export Current Selection button** will let you export all selected Map16 tiles to file, while the **Import Map16 button** will let you import them back in. To save time when importing, you can actually select multiple files at once. By default the tiles will be imported to the same location they were exported from, although this setting can be turned off using the red outline button mentioned above.

If you want to export and import all the FG and BG Map16 tiles at once (such as when you're moving all your resources to a different ROM), then it's better to use the next 2 blue export and import buttons. These will export and import all the tiles, even the tileset specific ones that are not visible at the time. Note that when exporting, the first 2 pages are taken directly from the ROM instead of the currently loaded data. And when importing, the imported data is immediately saved to the ROM. If you just want to preview most of the data within a full Map16 export file without saving, you can technically load it using the regular import button... however this does not load the tileset specific information in the file, so do not use this in place of the regular blue import button.

The default format to use for exporting is the new .map16 format, which was introduced in version 1.90 of Lunar Magic and makes things considerably easier. You can chose to export in the older raw generic .bin format instead, however this does not record the selection's dimension information or the position they were exported from. For maximum compatibility with older versions of LM using .bin files, you would usually select either a whole page of tiles or all the FG/BG tiles and name the file based on what it is. If a whole page, name the file Map16Page.bin (a second file called Map16PageG.bin will be created automatically). If all the FG tiles, name the file Map16FG.bin (a second file called Map16FGG.bin will be created automatically). If all the BG tiles, name the file Map16BG.bin (a second file called Map16BGG.bin will be created but can be discarded).

The second row of buttons starts with the **Display Page Numbers** button, which will display an outline for each page so you can more easily see which page a tile is in. It also has the effect of turning on the GFX slot outlines for the 8x8 tile selector, displaying which GFX file is loaded for each slot. The **grid** button displays a tile grid, both in the Map16 editor and the 8x8 tile selector. And the **gradient** button beside it can toggle either viewing a gradient as the background in the editor or the current level's back area color.

Then we come to the tile editing controls. When selecting tiles, these will change automatically to reflect the settings of all the tiles. If the settings for some of the tiles are not the same, the control will indicate this by changing to "--".

The first 2 buttons are for **flip x** and **flip y**. Next is a drop down box for the **palettes**. While there are technically 16 palettes in the SNES palette table, FG/BG tiles can only use palettes 0-7. Sprites use the remaining palettes of 8-F.

The **priority** drop down box affects tile layering. In general this can be used to make a tile appear above sprites or another layer. The actual effect will depend on the layer setting of the particular sprite, or whether the tile layers are both on the SNES main/sub screen for that level. For more information on how SNES layering works, you may want to consult an SNES hardware document.

Next are the 4 text edit controls that contain the 8x8 GFX tile numbers that make up the full 16x16 tile. You can use either the 8x8 editor or tile selector as a reference for getting the tile numbers you want to use. Just keep in mind that although it will let you enter values as high as 0x3FF since this is the max that the SNES tile format allows for, only up to 0x2FF is normally available.

The last tile editing control is the **"Act as"** setting, which is sometimes also called the **"Gameplay"** setting. While levels may be constructed of objects, the game actually checks for individual 16x16 tile numbers to determine how Mario and sprites should interact with a level. But since FG tiles above 0x1FF do not exist in the original Mario World game, Lunar Magic has implemented a lookup system where you can set a tile to act like another to avoid having to code your own block behavior. While this means you must choose a gameplay setting that resolves to one of the original tiles from 0-0x1FF, the settings can be chained together. For example, if tile 0x201 is set to tile 0x200, and 0x200 is set to 0x2B, the game will use the properties of tile 0x2B for tiles 0x200 and 0x201. You must not however chain them together in a way that would cause an infinite loop (setting 0x200 to 0x201 and 0x201 to 0x200 for example). Lunar Magic will check and correct for such infinite loops when saving. And while you can technically change the "Act As" settings of the original 0-1FF tiles, normally these should just be left to act like themselves.

The 2 most common "Act As" settings used when creating new tiles are 0x25 for a non-solid tile and 0x130 for a solid tile. If you want custom block behavior that was not in the original game, you must either code them yourself using the ASM jump points you can find in the Map16 Gameplay Programming section, or use one of the 3rd party block inserter tools that exist.

Near the bottom you'll find the **Remap** button. This brings up a dialog that allows you to remap individual 8x8 tiles selected based on certain criteria. The most common use would be to add an offset to all selected tiles, useful in cases where you may have imported a GFX file to a different slot than it was originally made for, causing the Map16 tiles you also imported to refer to the wrong 8x8s. The first field in this dialog lets you enter either a value to do exactly this.

For fine tuning or much greater control over which selected tiles are changed, you can instead enter a list of paired comma separated values in the large text edit in this dialog. The first value can be a single value or a range indicating the source of what you want to change, while the second value describes what to change it into. Each pair of values can be on their own line, or you can just separate them with more commas. There are several examples given in the dialog for the syntax you can use. The simplest is changing one tile into another: 100,25 would change tile 100 into tile 25. Or you can do a range:100-101,25 would change both tiles 100 and 101 into tile 25.

To add or subtract an offset, you must specify the sign to use. 100-101,+25 would change 100 to 125 and 101 to 126. Or 100-101,-25 would change 100 to DB and 101 to DC. But since the intent of adding/subtracting offsets is usually to move a range of tiles to another area, it may be easier to just use 'M' and the starting destination: 100-101,M125 would also change 100 to 125 and 101 to 126. It's also common to move a rectangle of tiles, so you can specify the top left and lower right points of a rectangle using a command like R100-111,M25 to change tiles 100 to 25, 101 to 26, 110 to 35 and 111 to 36.

You can also use this list to change which palettes are used for each 8x8 tile. P5-6,7 would cause all the selected tiles that use palettes 5 or 6 to use palette 7 instead. You can even change the palettes used by specific 8x8 tile numbers: 100-101,P7 would cause tiles 100 and 101 to use palette 7.

The gameplay settings can be remapped in this dialog as well, however you must be in 16x16 editing mode as that's a 16x16 setting. G100-101,+25 would add 25 to the gameplay setting of any selected 16x16 tile whose existing gameplay setting was 100 or 101 to make it 125 or 126. Or G100-101,M125 would have the same effect. There is also a special command to set a rectangle of Map16 tiles (by giving the top left and lower right points of a rectangle) to use a rectangle of gameplay settings with the specified base: R200-211,S25 would change Map16 tile 200 to use a gameplay setting of 25, 201 to use 26, 210 to use 35, and 211 to use 36.

When entering a list of remap pairs, it's important to remember that the source values always refer to the tiles as they currently exist *before* any remapping occurs, not as they would exist after processing the pairs that come before it. For example, you might think the sequence 100,25,25,100 would be of little use, but it will really cause tiles 100 and 25 to be swapped. This also means in a sequence like 100,25,100,30 the first pair is effectively useless as it's superseded by the second pair.

The last button in this editor is the **Page** button, which just scrolls the view to the page listed on the right. Typing the desired page into the text edit on the right also causes the view to scroll to it automatically.

Editors Menu : Background Layer 2 Editor Window

This particular editor will allow you to modify the 16x16 tile map that makes up the background image you see on layer 2 in some levels. However if a level is set to have level data on layer 2 instead of an image, there will be nothing to edit here... to edit layer 2 on those levels check the “Edit Level Layer 2 Mode” item in the edit menu. If you want to edit a layer 3 tile map instead (which is sometimes used for simple backgrounds), check the “Load Layer 3 of Level” command in the overworld's file menu.

The original game uses backgrounds that were 432 pixels in height, but LM expands this to 512 height BGs (5 extra rows of 16x16 tiles). The extra height is mostly only useful in vertical levels or resized horizontal levels... in the original game's horizontal levels, the lower 5 rows normally aren't shown as long as the BG init position and/or BG Height are set correctly for the level entrance used.

The editor behaves much like the main Lunar Magic window. A **left click** selects multiple tiles, and a **right click** pastes a copy of the tiles that are selected. If no tiles are selected, it pastes tiles from the Map16 editor (you can force this by holding down the **control key** at the same time or using the **insert key** instead of the mouse). Use **Shift + Right Click** to fill an enclosed area with a pattern of tiles from the currently selected area. If no tiles are selected, the pattern will be from the Map16 editor (you can force this by adding the **control key**). You can also resize a selected area as a pattern of tiles by using the mouse on resize borders. A **double left click** on any tile in the background automatically selects it in the Map16 editor. The **delete key** deletes any selected tiles, and the **escape key** deselects any selected tiles. Use the **Page Up** and **Page Down** keys to change the current Map16 bank that the background is using to render the tile map. **F9** can be used to add a value to the 16x16 tile number of all selected tiles. Hit **F1** in this editor to refresh the main Lunar Magic window.

This editor also has Windows clipboard support, which is accessible through the standard clipboard keyboard shortcuts (**ctrl-x** for cut, **ctrl-c** for copy, and **ctrl-v** for paste). You can use **Control-Z** to undo, and **Control-Y** to redo.

The background is saved along with the level itself (but not changes to the Map16 data, which must be saved separately in the Map16 editor). All unsaved changes will be lost if you switch levels.

The background tile map data simply consists of references to the BG Map16 tiles you can find in the "16x16 Tile Map Editor Window". Therefore, you'll probably want to have the 16x16 Tile Map editor window open so you can see which other tiles are available. Each background can only access the tiles within a single bank of tiles within the 16x16 editor (1 bank = 0x10 pages or 0x1000 tiles). However, you can change the current bank the background is using with the keyboard shortcuts mentioned above. Attempting to paste a 16x16 tile that the background cannot access with the current settings may cause the equivalent tile number of the current bank to be pasted instead.

By default Lunar Magic links to the shared backgrounds of the original game to save space, but it will save the current level's background as a custom background as soon as you modify it.

Note that while you cannot access FG Map16 in backgrounds or BG Map16 in level objects, they do have access to the same 8x8 tiles. This means FG Map16 tiles can be copied into the BG Map16 tile area and vice versa, and they'll still appear using the correct graphics. For example, you could take the coin Map16 tile in the 16x16 Tile Map Editor Window, paste it into one of the blank spots in the background Map16 data, then select that tile and stick it into your background. It might look like a real animated coin in the game and even turn into a block when you hit a POW, but it's part of the background image so the player can't interact with it.

The editor's buttons all have tooltips, which makes them fairly easy to figure out. The first two are standard undo and redo buttons.

The 16x16 Tile Map Editor Window is the same editor you can open from the main Lunar Magic window, as mentioned above. The Palette Editor Window is also the same one from the level editor.

Next is the **zoom** button, which allows viewing the tiles at different sizes. The next button with a red X is the **Auto-Deselect on Editor Select** button. If this option is enabled, whenever you make a new selection in the 16x16 Editor Window, anything selected in this window will be deselected. This can make it easier for pasting, since then you don't have to always remember to hold down the control key.

Beside it is the Change Background Map16 Bank button, the Copy Background Image from other Level button, and the Remap button, which have their own help file sections.

Editors Menu : Overworld Editor Window

For information on this editor, please read the Overworld Editor Window documentation.

Level Menu : Super GFX Bypass

Opening this dialog will cause Lunar Magic to immediately install an ASM hack into the ROM, allowing you to bypass the standard GFX files loaded by the level. This is the standard way to access your ExGFX files in a level.

The checkbox at the top of the dialog will determine whether or not this level will bypass the standard GFX and use the custom settings in this dialog. If not enabled, all the other controls will be disabled.

In this dialog are several drop down boxes that allow you to select which GFX files to load for the FG/BG, the sprites (SP), or the Extended Animated area. Values 0-31 are for accessing the original GFX files of the ROM. Values 32-7E are invalid...DO NOT USE THEM! Value 7F should only be used to skip decompressing a file for a slot you aren't using. Values 80-FFF are for any ExGFX you've inserted. Each index value has a number in parenthesis that indicates the current ROM address of the GFX file in question, assuming it's inserted.

The GFX files contain 0x80 4bpp tiles, which means they should be 0x1000 bytes (4KB). The AN2 file is an exception... it can have up to 0xD0 4bpp tiles and be 0x1A00 bytes. Even if you aren't using the 4bpp option the requirements for these files are the same, as the program will internally auto-convert the files for 3bpp on insertion if needed.

The "**Edit**" buttons beside each slot will launch an external tile editor you can use on the graphics files once they have been extracted from the ROM. If the file doesn't yet exist in the ROM or extracted folders it may offer to create a new blank one for you. Just remember to reinsert any edited files to the ROM if you want your changes to show up. You can configure the tile editor to use from the "Setup Tile Editor" menu.

It's important to remember that although you're bypassing the standard list to specify exactly which GFX files are to be loaded by the level, the game still thinks that the tile set selected in the "Change index of GFX in Header" menu have been loaded. That means the original setting will still control some of the Map16 data loaded for the level, the animated tiles used for the level and the objects available in the "Tileset Specific Objects" menu.

This dialog is intended to replace the older ExGFX bypass dialogs, which can only access ExGFX files up to 0xFF. Enabling the settings in this dialog will cause the settings in the older dialogs to be ignored.

Level Menu : Layer 3 GFX/Tilemap Bypass

In the original game, layer 3 was used for things like the status bar, message box text, the credits text, the title screen, tides, the layer 3 smash command in Iggy's castle, and sometimes simple foregrounds/backgrounds (mist, goldfish, etc).

Opening this dialog will cause Lunar Magic to immediately install an ASM hack into the ROM, allowing you to bypass the standard GFX files used by the level.

As with the "Super GFX Bypass", in this dialog are several drop down boxes that allow you to select which GFX files to load for layer 3. In the original game, the layer 3 GFX were only decompressed once at the start of the game and mostly left alone. But once you save at least one level or submap in Lunar Magic that bypasses the layer 3 GFX, the program will enable bypassing layer 3 GFX on a per level and per submap basis. If the bypass isn't enabled for a particular area, it will simply use the original files (0x28-0x2B).

Layer 3 GFX files contain 0x80 2bpp tiles, which means they should be half the size of regular 4bpp GFX files. So the files should be 0x800 bytes (2KB) instead of 0x1000 bytes (4KB). Remember that 2bpp tiles can only use 4 colors (1 transparent + 3 regular colors). These colors come from the main palette table, however the palettes are assigned by using the first 4 colors for the first palette, the next 4 colors for the second palette, and so on. So only the first 16 colors in the main palette table can be accessed (the top 2 rows in the palette editor window), and only 3 out of every 4 of those colors can actually be shown.

The "**Edit**" buttons beside each slot will launch an external tile editor you can use on the graphics files once they have been extracted from the ROM. If the file doesn't yet exist in the ROM or extracted folders it may offer to create a new blank one for you. Just remember to reinsert any edited files to the ROM if you want your changes to show up. You can configure the tile editor to use from the "Setup Tile Editor" menu.

This dialog also lets you **bypass the layer 3 tilemap** by choosing a GFX file to load that actually contains tilemap data. Since layer 3 is shared with the status bar (it occupies 32x5 tiles on the top left half of layer 3), you must choose the **destination** for the file. "Under Status Bar" will avoid copying over the game's original status bar tilemap. "Start of Layer 3" on the other hand would overwrite the status bar area... you would normally not use this unless the status bar has been turned off or you're using a third party patch like the sprite status bar (which replaces the status bar with sprite tiles so full layer 3 images can be used). "Last Line of Status Bar" will copy over the last line of the original game's status bar... you normally wouldn't use this either unless you've possibly shortened your status bar from the original. "Bottom Half of Layer 3" would copy only over the bottom half of the tilemap starting on the left side.

The **file size** of the tilemap file must be set in advance so the game and editor know what to expect. 0x2000 bytes (8 KB) would be for a full 512x512 image, and is probably what you'll be using most often. 0x1000 bytes (4 KB) would be for 512x256, effectively half of layer 3 (top or bottom half). And 0x800 bytes (2 KB) would be for 256x256, which is only enough to cover a single screen.

Do not adjust the size of the tilemap file to avoid the layer 3 status bar! This is already taken into account when you set the appropriate destination for the file. It will skip over the status bar area of the tilemap when copying your file into VRAM.

To edit the layer 3 tilemap file, see the Load Layer 3 of Level command in the overworld file menu.

For Patch Authors or Advanced Users with Custom Layer 3 Status Bars: If you have adjusted the height of your layer 3 status bar tilemap, you will need to make a couple minor edits to tables in the ROM inserted by LM so that using the "Under Status Bar" destination setting will still work as intended. 0x800BC (PC) holds the 2 byte layer 3 VRAM destination for that setting, and 0x800C4 (PC) holds the 2 byte size of the layer 3 status bar tilemap in bytes.

Level Menu : Bypass Standard FG/BG GFX List

Note: This is an older feature that has been replaced by the Super GFX Bypass dialog. It's still safe to use, but you can only access up to the old ExGFX file max of 0xFF.

Opening this dialog will cause Lunar Magic to immediately install an ASM hack into the ROM, allowing you to bypass the standard list that determines which GFX files are loaded by the level.

The checkbox at the top of the dialog will determine whether or not this level will bypass the standard list and use the custom settings in this dialog. If not enabled, all the other controls will be disabled.

This dialog has a drop down list (referred to as the “**Custom List Index Number to Use**”) of 0xFF entries that you can use to store your GFX lists. To keep things organized, I suggest using 00-7F for FG/BG lists, and 0x80-0xFE for SP lists. Obviously you can share a single list between levels just like with the standard lists. Note that these lists aren't stored with your level...they're kept in a table that's inserted to the ROM immediately when you hit “ok”. Only the list index number to use for your level is saved along with the level itself. Check the file menu for commands that will let you extract and insert this list table, if you ever want to backup or transfer the list to another ROM.

Also in this dialog are the four drop down boxes (referred to as the “**GFX Files to Use**”) that allow you to select which four GFX files to load for the table index selected in the main drop down box. Values 0-31 are for accessing the original GFX files of the ROM. Values 32-7F are invalid...DO NOT USE THEM! Values 80-FF are for any ExGFX you've inserted. Each index value has a number in parenthesis that indicates the current ROM address of the GFX file in question, assuming it's inserted. Note that if you change the list index number before hitting “ok”, the four GFX file settings for that index are not saved.

It's important to remember that although you're bypassing the standard list to specify exactly which GFX files are to be loaded by the level, the game still thinks that the tile set selected in the "Change index of GFX in Header" menu have been loaded. That means the original setting will still control some of the Map16 data loaded for the level, the animated tiles used for the level and the objects available in the "Tileset Specific Objects" menu.

Level Menu : Bypass Standard SP GFX List

Note: This is an older feature that has been replaced by the Super GFX Bypass dialog. It's still safe to use, but you can only access up to the old ExGFX file max of 0xFF.

This menu pretty much does the same thing as the "Bypass Standard FG/BG GFX List" menu command, except it's for sprites instead. So check the documentation for that menu command.

Level Menu : Extend Animated Tile GFX

Note: This is an older feature that has been replaced by the Super GFX Bypass dialog. It's still safe to use, but you can only access up to the old ExGFX file max of 0xFE.

Opening this dialog will cause Lunar Magic to immediately install the “Bypass Standard FG/BG GFX List” ASM hack, in addition to another ASM hack that lets you append a GFX file to the end of the animated tile area. You must insert the GFX file used as 4bpp, or it will not show up correctly.

The checkbox at the top of the dialog will determine whether or not this level will extend the animated tile area or not. If enabled, the drop-down box will let you select the GFX or ExGFX file you wish to use to extend the area.

Making use of the extended animated tile area to change the game's original animations will require manually hacking the animated tile frame references in the ROM. An extensive explanation of this is beyond the scope of this help file, but to get started, you can find the table at 0x2BB99 in the ROM. Each animated tile uses 4 frames, and each frame reference takes up 2 bytes. These 2 bytes are actually the lower 16 bits of an SNES RAM location to get the source tile from. A 2 byte reference of 0xAD00 is the starting address of the extended animated tile area.

Level Menu : GFX Index in header

This dialog will let you change the current graphics tile set index for the FG/BG and Sprites, as well as inform you of the actual GFX file numbers that are used for any given index. For example, FG1=14 means that GFX14.bin is one of the GFX files that are loaded by the level. Lunar Magic loads the GFX directly from the ROM when rendering levels however, so if you want the ROM or Lunar Magic to use the GFX files that you've extracted and then modified, you'll have to use the "Insert GFX to ROM" menu command first.

It should be noted that the FG/BG index will affect the "Tile Set Specific" objects available. The "T" value in the FG/BG list indicates which group of tileset specific objects will be available. The FG/BG index also alters which animated tiles are used in the level, and even change some of the Map16 tile data loaded. So be aware that you can make your level look pretty messed up when you change this.

The FG/BG index also determines if a "No Yoshi" intro will be displayed. If you wish to disable the intro for a level, check the "Change Other Properties" Menu.

There is a special case for levels set to use FG/BG index 3 that use a level on layer 2... level tiles on layer 2 that have a palette number less than 4 will then have 4 added to their palette number.

If you're wondering how the graphics in Mario World are handled, they're actually compressed and split up into multiple segments, which Lunar Magic can decompress and save to standard SNES tile GFX files (See the "Extract GFX from ROM" Menu). Each level loads 8 selectable GFX Files, in addition to GFX files 32, 33, and a few others I forget at the moment that are almost always resident in VRAM. The first 4 selectable files are for the FG/BG, while the other 4 are for sprites. There is a table set up for both of these that determines which 4 files to load for any given index value, and this index is stored in the level header. And these are the two indexes you can change in this dialog.

It's possible to bypass this list and explicitly tell the ROM which files to load for the level. It's also possible to insert additional GFX files into the ROM, which can be accessed using the bypass hack.

Level Menu : Edit Level ExAnimated Frames

As you probably already know, Mario World uses animated tiles to create some effects, such as spinning question blocks, coins, bouncing note blocks, etc. This dialog will allow you to set up new custom animations of your own on a per-level basis.

The first drop down box is for which **slot** you want to use for a particular animation. There are 0x20 entries available. However, using multiple entries with large tile amounts can have a negative impact on the game (read the section about technical information and performance for more information).

The next two slots are for the animation **type** and **trigger**, which are discussed in detail below. The "**Colors**" value is used to set the number of colors to animate in palette animations. And the "**frames**" value near it determines the number of frames the animation should use.

The "destination" value is the starting tile number of where to put the animated frames of the tile. Open up the 8x8 tile editor to see where you can put things. Tiles from 0-0x2FF (layer 1/2 graphics), 0x400-0x5FF (sprite graphics) and 0x1C00-0x1DFF (layer 3 graphics) are valid destinations. Advanced users can specify **direct VRAM offsets** instead by putting a '1' and the 4 digit VRAM offset to use.

The "**Use alternate ExGFX file for source**" option is a per-slot setting that enables you to use one of a few non-compressed ExGFX files for source frame data. These files can be up to 0x8000 bytes in size, but keep in mind that they're not compressed and thus will take up a fair amount of space in the ROM. The "**Alt ExGFX**" drop-down box determines which ExGFX file will be used, but the setting is NOT slot specific! You can only chose one file per level (although the global animations can use a separate one).

The large group of "**Frames**" boxes determine where to get the source data for each of the frames in the animation. You can have a max of 0x100 frames, but only about 0x40 of these are shown in the dialog at once. You can use the **<** and **>** buttons to switch between which group of 0x40 frames to edit.

The frame source values are normally set to tile numbers from the 8x8 tile editor. Open up the 8x8 tile editor and hit "Control + Shift + Page-Down" to view animated tile pages, and move the mouse over the tiles to get their tile number. Tiles 0x600-0x77F are for the original game's animated tiles, and 0x780-0x857 are for tiles used to extend the animated tile area with the ExGFX file you've selected in the Super GFX Bypass dialog. You can also technically use Mario's graphics with tiles 0x900-0xBE7 (this is where the game gets the animated berry frames from).

If a particular slot is using the "Use alternate ExGFX file for source" option, you can only use tile numbers from a specific range depending on which ExGFX file is being used. File 0x60 uses tiles from 0xC00-0xFFF, 0x61 from 0x1000-0x13FF, 0x62 from 0x1400-0x17FF, and 0x63 from 0x1800-0x1BFF.

It's also possible to work with **direct offsets** for source frame data instead of using tile numbers from Lunar Magic. This can be useful for advanced users, or when you're using palette animation with more than one color. Simply put a '1' and the 4 digit offset to use. The offset in this case is effectively the lower 16 bits of the RAM address in bank 7E on the SNES. For reference, Mario's graphics start at

0x2000, the original game's animated tiles (AN1) start at 0x7D00, and the extended animated tile area (AN2, which is what you usually use for your custom animated tiles) starts at 0xAD00. If using the "alternate ExGFX" file option, the 4 digit offset is simply the offset of the data in the ExGFX file when viewed in a hex editor, which means it starts at 0. Remember when calculating offsets that each 8x8 4bpp tile takes up 0x20 bytes, or if doing a palette each color takes up 2 bytes.

For Layer 3 Graphics: The source frame tile numbers will still refer to 4bpp tiles even though layer 3 uses 2bpp tiles. This means that when looking at your 2bpp tiles in a tile editor, you should start your tiles for a particular frame on an even tile number, then divide the tile number by 2 and add it to the starting tile number of the source slot to get the tile number to enter.

Below the frames are 5 buttons that let you copy and paste data to the standard Windows clipboard. The **“Copy Slot”** and **“Paste Slot”** buttons will let you copy from or paste to the current entry. The **“Copy All Slots”** button will copy all animated tiles in your list to the clipboard, while the **“Paste All Slots”** button will paste all the non-blank slots you copied using the “copy all slots” button to the same slots they came from. The **“Insert All Slots”** button works almost like the "Paste all Slots" button, except it will insert the copied slots to unused slots in your animated tile data list, starting at the currently selected slot. If there are not enough free slots to insert all the data, the program will notify you of this.

There are also 2 more buttons (**“Clear Slot”** and **“Clear all Slots”**) that can either clear the current slot, or clear all slots for the level.

The **“Trigger Init”** button opens up another dialog that lets you initialize "custom triggers" and "manual frames" entries. These are applied when the level is loaded. Level initialization values will override global ones. Check the triggers section below for more information about these two trigger types.

Animation Types

Not Used

This means the slot itself is not used. No animation will occur.

1 8x8, 2 8x8s: line, etc... up to 0x20 8x8s: line

This transfers the number of tiles indicated to the destination. Since Mario World uses 16x16 blocks, the most common one you'll be using in this group is the "4 8x8s: line" type (and ones that have a multiple of 4), as the 4 tiles can be rearranged into a 16x16 block in the Map16 editor.

Newbie Warning: A common mistake by beginners is to animate a 16x16 tile by doing each 8x8 individually in separate slots. Not only is this slower and a waste of slots, the individual 8x8s will not be animated in sync with one another (unless placed in specific slots, see technical info below). The best way to animate a 16x16 is to use the "4 8x8s: line" or "4 8x8s: 16x16" types.

For Layer 3 Graphics: As Layer 3 graphics are 2bpp instead of 4bpp, these will each transfer twice as many tiles as listed.

1 8x8 2bpp

This transfers a single 2bpp tile. However, most SMW graphics are 4bpp... 2bpp is just used for layer 3 graphics.

2 8x8s: stacked, 4 8x8s: 16x16, 8 8x8s: 32x16

These types arrange the tiles into different formations for the destination (note that despite this, the **source** tiles must still be in a line!). 2 8x8s stacked is for 2 8x8s, one on top of the other. 4 8x8s: 16x16 is for 4 tiles arranged into a 16x16 formation (top 2 tiles first then bottom 2 tiles). 8 8x8s: 32x16 is like the 16x16 one, but with the 4 top tiles first then the bottom 4 tiles.

While it might be tempting to use the 4 8x8s: 16x16 instead of the 4 8x8s: line type because it's a bit easier for viewing the result in the 8x8 tile editor, the former requires a little bit more processing for the SNES to do. So it makes more sense to use the "line" type whenever possible. Not to mention it makes it easier if you want to combine multiple tiles in one entry later.

For Layer 3 Graphics: These are only intended for 4bpp tiles not 2bpp, so this won't work for layer 3... the second row of tiles will skip a row due to the graphics format difference.

Palette

This type allows you to change the colors in the palette table for the level. Since palette animation affects all tiles that use that color and colors take up less space than tiles (a single 16x16 tile is equivalent to 0x40 colors storage wise), it can be a very effective way to do large scale animations on the SNES.

The "**Colors**" value should be set to the number of colors you want to change (1-0x80). If you're only animating one color, the source frame values are used as the actual SNES RGB color values for the animation. To figure out what the values should be for a color, you can edit colors in the palette editor and copy the hex SNES RGB values displayed (but it's faster to copy the color to the windows clipboard in the palette editor, as you can paste the SNES RGB value as text).

If you're animating more than one color, the source frame values should be *direct offsets* to SNES RGB values stored in the ExGFX file. You'll likely want to open up your source ExGFX file in a hex editor to modify color values and figure out the offset to use. SNES RGB values are 2 bytes each, and should be stored in little endian format (so 7FFF for white would be FF 7F in the hex editor). For example, if you wanted to do a 3 frame palette animation for 2 colors (SNES RGB colors per frame: Frame 1 - 7398, 6ACE Frame 2 - 6B55, 5208 Frame 3 6312, 3942), and you stored them at the start of an ExGFX file for the AN2 slot, you'd put 98 73 CE 6A 55 6B 08 52 12 63 42 39 in the hex editor at file offset 0, then use 1AD00, 1AD04, 1AD08 for the frame values in LM.

The "**Destination**" value should be set to the first color number in the palette table to animate... for example color 5 of palette 8 would be color 85 in the palette table.

Just like with the palette animation in the Yoshi coins, this animation is not compatible with the fade in/out that occurs at the end of a level (the effect will look rather ugly in fact).

Palette + Working

This is just like regular palette animation, with 1 difference: it also copies the palette modifications to the game's 2 working copies of the palette in RAM for you. While this means the SNES has to spend more time processing, there are a few situations where this type should be used instead of the regular palette animation. For example, if you're using a One-Shot trigger this will make sure the fade in/out at the end of the level doesn't cause colors to revert back to the way they were before. The same goes for the Manual Frame triggers (assuming they're not doing continuous animation). Also the palette rotate animation types use the colors in the working copies of the palette for their source data, so if you plan to use palette rotation on the same colors modified by another type of palette animation, you pretty much need to use this.

Palette + Working, Stop on Fade

This one is like "Palette + Working", but at the end of the level when colors are supposed to fade in/out, the animation will simply stop. This will allow the fade in/out to operate correctly.

Palette Back Area Color

This type animates only 1 color: the back area color. Therefore the "Colors" and "Destination" values are ignored.

Minor technical note: unlike all the other animation types, this one does not require any extra processing during vblank as the game already updates this color every frame.

Palette Back Area Color, Stop on Fade

This one is like "Palette Back Area Color", but at the end of the level when colors are supposed to fade in/out, the animation will simply stop. This will allow the fade in/out to operate correctly.

Palette Rotate Right

This animation causes a strip of colors in the palette to move one space to the right per frame, with the last color of the strip wrapping around to the first color. The "**Colors**" value should be set to the number of colors to rotate (2-0x80, but you likely won't be doing more than 0xF in an entry since you really shouldn't rotate the transparent color in each palette). The "**Destination**" value should be set to the first color number in the strip to rotate.

As the number of frames is already determined by the number of colors to rotate, the "**Frames**" value for this type is used as a **frame delay** value. A value of 1 causes no delay. A value of 2 will cause the animation to occur every 2nd frame, effectively making the animation go at half speed. A value of 3 would be for every 3rd frame, etc. The frame counter in RAM for this slot is also used as a frame delay counter (mentioned just in case any ASM hackers are puzzled as to why the frame counter isn't doing what you'd normally think it would).

As this animation uses the game's 2 working copies of the palette, there is no frame source data to set up. Any values entered there are ignored.

Unlike regular palette animation, this type is fully compatible with fade in/out at the end of a level. It also treats triggers differently... if the trigger has not been activated, the animation simply won't take place.

Palette Rotate Right, Reverse on Trigger

This works just like "Palette Rotate Right", but when triggered it will reverse the direction of the rotation.

Palette Rotate Left

This one is like "Palette Rotate Right", it just goes in the opposite direction.

Palette Rotate Left, Reverse on Trigger

This works just like "Palette Rotate Left", but when triggered it will reverse the direction of the rotation.

Triggers

For triggers that cause "x2" to appear beside the number of frames box, it means you really have twice the number of frames you set (and also means the max value you can enter is halved to 0x80). The first half of the frames will be used for the non-triggered animation, and the second half for the triggered animation.

Most animations are continuous, meaning that they're always running and uploading frames. However, some triggers change this. For example, "One Shot" triggers do not animate at all until triggered. And "Manual Frame" triggers only update when the frame number has been changed. Since this can potentially reduce the amount of processing in vblank required, you should chose carefully what type of trigger you want to use.

Note: The maximum number of frames you can have for "One Shot" triggers is 0xFF, not 0x100.

None

This means the animation has no trigger, so it's always running.

POW, Silver POW

Standard triggers, pretty much self-explanatory.

ON/OFF

For some reason, Nintendo reversed the meaning of ON/OFF as it relates to the actual RAM value used. The first half of the frames are used for ON, and the second half for OFF.

Have Star

Standard trigger.

Timer < 100

Once the timer goes less than 100, it'll trigger. If the timer should go back above 100 for some reason, the trigger will turn back off as you'd expect.

Timer < 100 One Shot

If the timer goes less than 100, this will trigger the animation, but it will only run once.

>= 5 Yoshi Coins

If the player collects 5 or more yoshi coins in the level, the trigger will activate. Remember that the game keeps track of whether a player has collected all the yoshi coins in a level, so if the player comes back the yoshi coins will be gone. If this is a problem, you can place the yoshi coins using direct Map16 so they'll still respawn.

>= 5 Yoshi Coins One Shot

Is just like the regular Yoshi coin trigger, but will only run once.

Do Not Use

These are reserved for future use.

Precision Timer Palette Rotate

This isn't a trigger in the normal sense. It causes any animation that uses it to share the frame counter of the first slot in every group of 8 (so slots 0-7 will use the counter from slot 0, 8-F will use the counter of slot 8, 10-17 will use slot 10, and 18-1F will use slot 18). Its intended use is with palette rotation animation types in combination with faster animation speed tricks to allow many more variable speeds for this animation through the use of the frame delay value.

Manual 0-F

These are manual triggers that can be used to directly control which frames of an animation to display through the use of ASM in custom blocks. While most animations automatically proceed from one frame to the next, these will not. They will stay on the particular frame number they're set to and won't update VRAM (and hence won't use up any vblank time) until you change their frame number in your code. This can be useful for showing states... for example, the original game's ON/OFF block would have been an ideal use for this.

The 0x10 bytes starting at $7FC070 in SNES RAM control the frame values for the 0x10 Manual frame triggers. So for example, if you want all animations using "Manual 4" as a trigger to show their 3rd frame, you'd store a byte value of 2 to $7FC074.

Just remember that the Manual frame values are not automatically initialized on level load unless you specify it (hit the "**Trigger Init**" button in this dialog).

Warning: you cannot initialize to frame FF... the ExAnimation system uses this value as the default frame to start all animations on level load, so it'll think it's already displaying this frame.

Custom 0-F

These are custom triggers that can be used to control an animation through the use of ASM in custom blocks. These work just like a regular trigger, but instead of relying on the original game's predefined events like a POW being hit, there are bit flags set aside in RAM you can use for your own custom events.

The 2 bytes at $7FC0FC in SNES RAM contain the flags for all of these. If a bit is on it means the trigger is enabled. So for example, if you wanted to turn on the "Custom 9" trigger, you'd do LDA $7FC0FC ORA #$0200 STA $7FC0FC. To disable it you'd mask the bit out with AND #$FDFF instead.

Just remember that the Custom triggers are not automatically initialized on level load unless you specify it (hit the "**Trigger Init**" button in this dialog).

One Shot 00-1F

These are one shot triggers that can be used to control an animation through the use of ASM in custom blocks. These work much like a regular trigger, except when triggered the animation will only go through its animation sequence once then stop. The triggers use custom bit flags set aside in RAM that can be set whenever you like.

The 4 bytes at $7FC0F8 in SNES RAM contain the flags for all of these. If a bit is on it means the trigger is enabled. So for example, if you wanted to turn on the "One Shot 01" trigger, you'd do LDA $7FC0F8 ORA #$0002 STA $7FC0F8. To disable it you'd mask the bit out with AND #$FFFD instead.

Once triggered, the animation will proceed through its frames exactly once, then it will stop on the last frame and turn off the flag responsible for turning it on. If you like, you can trigger the animation again by turning the flag back on. Note that you should not assign the same one shot trigger number to more than one animation slot in a level, as the first slot to reach the end of its animation would turn off the others.

One Shot triggers are always initialized to off on level load.

Technical Information and Performance

There are two performance measures to consider when making animations: game engine **slowdown** and **vblank** time. The more animation entries you make, the more the game has to process and the more likely you'll encounter slowdown... but this will vary depending on the level type (horizontal levels with an image on layer 2 are fairly forgiving, while horizontal layer 2 levels are not), what you have enabled (like sprite buoyancy), the number of sprites on the screen at once, and any 3rd party ASM hacks you've installed that run during a frame.

Similarly, with vblank the more animation entries you make (and perhaps more importantly, the *size* of the data to transfer), the more likely you'll exceed vblank time, which on the actual SNES and some emulators will get you into trouble. On the SNES, the only time you can access certain video registers for updating is when the screen isn't busy drawing anything. Such as in vblank, which occurs for a limited amount of time between each frame. Exceeding vblank in SMW tends to cause flickering at the top of the screen.

To give you some idea of what can be done: in testing a typical horizontal level with an image on layer 2, and filling both the Level and Global ExAnimations completely (all 0x40 entries) with continuous animations that do 4 8x8s: line, vblank time was never exceeded (there were a couple scanlines worth of capacity left), and the game engine was still able to handle a fair amount of sprites on screen at once (although there wasn't much capacity left for processing anything else you might want to throw in). When trying the same test on a horizontal layer 2 level, vblank time wasn't exceeded (barely... I wouldn't suggest doing this, you usually want to leave a bit of capacity for fade in/out at the end of a level), but the game did start to slow down more easily with multiple sprites on the screen at once.

So, how do you deal with these? Well, you can turn off the original game's tile animation to save about 5 scanlines or so of processing (3 of which are in vblank). You could use FastROM to make the game code run a bit faster. You can use animations that don't animate continuously (triggers like Manual, One Shot). You can also use less entries by combining some entries together... for example, instead of 2 slots using 4 8x8s:line, use one slot that uses 8 8x8s instead. However, you have to be careful not to try and transfer too many tiles within a single frame when you do that. But this requires knowing which slots animate on which game frame.

Which means now we get into technical details. ;) Mario World's tile animations run at 7.5 fps, while the yoshi coin palette animation runs at 15 fps. Since the actual game runs at 60 fps, SMW splits up the tile animations it has to do among 8 different frames. This results in the tile animations effectively running at 60/8=7.5 fps, but also spreads the entries across those frames so you don't end up trying to process all the entries in a single frame (meaning that for example, instead of doing 8 different animations at once every 8th frame, you do a single different animated tile on each of those 8 frames).

Lunar Magic does the same thing. The first 8 ExAnimation slots all actually run on different frames. However, the next 8 slots all run in the exact same frames as the first 8 slots. And so on. This means slots 0,8,10,18 animate in the same game frame, as do 1,9,11,19, then 2,A,12,1A, etc. The global animation slots run in the same frames as the level ones. This effectively means that in any particular game frame, a total of 8 ExAnimation slots are being run: 4 global ones, and 4 level ones. So once you fill enough ExAnimation slots to go past 8 slots, it's worth thinking about whether you want to start combining some entries together. With this in mind, you can optimize your slots according to how much data they transfer to try and spread things around. Or if your concern is for keeping certain animations in sync, you can either place them within the same group of slots that do animate on the same frame, or transfer more tiles in those same slots.

For reference: if you want to create a block that changes its behavior depending on the frame of an animation being displayed, you can find the frame values in SNES RAM at $7FC080, one byte per slot, with level slots first then global slots 0x20 bytes later. In previous 1.6x versions of Lunar Magic, $7FC004 was used as the frame counter for all ExAnimations, and they all used a fixed number of frames (0x10). While the counter is being maintained for backwards compatibility with certain custom blocks, it's only truly accurate now for slots 0,8,10 and 18 (although if you have a block that hurts mario, it seems the next 2 slots after those are still close enough).

Advanced Animation Tricks

If you know how Lunar Magic implements animation (see technical info above), it's possible to use that knowledge to do some interesting things.

You can do animations that run twice as fast (15 fps), by simply using 2 slots that are 4 slots apart within a group of 8 (like for example, 0 and 4) and setting the same destination in each. If you want to take it further, you can go at 30 fps by using slots 0,2,4,6. And if you want to go at 60 fps, you'd use all of slots 0-7 to do it (and if you give yourself a seizure animating the palette at that ridiculous rate, don't blame me).

You can overwrite global animations with level ones by placing the animation within the same set of 4 slots that animate during that game frame. It's also possible to do the same thing to the game's original animations, but this requires knowing which group of 4 slots you can put your animation in for whichever particular animation you want to overwrite. Slots 0,8,10,18 can do animation for tile 0x60 - question block, 0x64 - music block, tile 0x68 - turn block. Slots 1,9,11,19 can do tile 0x74 - midway tiles, tile 0xEA - always turning turn block, tile 0x80 - berry. Slots 2,A,12,1A can do tile 0x50 - POW door, tile 0x54 - POW coin, tile 0x58 - brown block. Slots 3,B,13,1B can do tile 0x5C - black piranha plant, tile 0x78 - POW question block, tile 0x7C - line guide. Slots 4,C,14,1C can do tile 0xDA - ON/OFF block, tile 0x6C - coin, tile 0x70 - animated water. Slots 5,D,15,1D can do tile 0x4C - line guide end/lantern/twinkle/stars/lava, tile 0x44 - weed/up conveyer, tile 0x48 - down conveyer/lava. Slots 6,E,16,1E can do tile 0x40 - lava/candle background.

Note that Lunar Magic may not display some of these tricks accurately if its Animation Rate setting is too low, but this will not affect how it shows up in the game itself.

Something else you can do is sacrifice Lunar Magic's global slots in a level to turn them into local ones, effectively giving you twice as many local slots. Hit **Ctrl+Alt+F7** in the main editor to unlock this, then turn off Lunar Magic's global animations for the level in the animation settings. But be very cautious about using this feature, since you need to keep vblank limitations in mind. And remember that it's still more efficient to transfer more tiles per slot than to use more slots.

Source Frames

Tile Range	Direct Offset	Type	Description
0x600-0x77F	0x17D00-0x1ACFF	RAM	Original game's animated tiles AN1, GFX file 0x33.
0x780-0x857	0x1AD00-0x1C7FF	RAM	Extended animated tile area AN2.
0x900-0xBE7	0x12000-0x17CFF	RAM	Original game's frames for Mario, GFX file 0x32.
	0x10000-0x17FFF	ROM	Non-compressed

0xC00-0xFFF*			ExGFX file 0x60.
0x1000-0x13FF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x61.
0x1400-0x17FF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x62.
0x1800-0x1BFF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x63.

*Requires animation slot to be set to use alternate ExGFX source, and for Alt ExGFX file to be set to appropriate file.
Note: If using direct offsets, remember that a single 8x8 4bpp tile takes up 0x20 bytes and a single SNES color takes up 2 bytes.

Destinations

Tile Range	Direct Offset	Type	Description
0-0x2FF	0x10000-0x12FFF	VRAM	FG/BG graphics FG1, FG2, BG1, FG3, BG2, BG3 (in order).
0x400-0x5FF	0x16000-0x17FFF	VRAM	SP graphics SP1, SP2, SP3, SP4 (in order).
0x1C00-0x1DFF	0x14000-0x14FFF	VRAM	Layer 3 graphics (2bpp tiles) LG1, LG2, LG3, LG4 (in order).

Note: If using direct VRAM offsets, remember that the SNES uses word (16 bit) addressing for VRAM. So if you're working with byte offsets, divide by 2 before adding 0x10000 for the direct offset to enter. This means single 8x8 4bpp tiles would really be 0x10 away from each other in VRAM addressing, not 0x20.

RAM Locations

RAM Address	Size	Description
$7FC070	0x10 bytes	Manual Frames 0-F , 1 byte each.
$7FC080	0x20 bytes	Frame Counters for level slots 0-1F, 1 byte each.
$7FC0A0	0x20 bytes	Frame Counters for global slots 0-1F, 1 byte each.
$7FC0F8	0x4 bytes	One Shot Triggers 0-1F, 1 bit each.
$7FC0FC	0x2 bytes	Custom Triggers 0-F, 1 bit each.

Level Menu : Edit Global ExAnimated Frames

This dialog is pretty much the same as the Edit Level ExAnimated Frames one, so you should read that section for more information.

Just remember that animations added here will run in all levels, potentially consuming processing and vblank time. So think carefully about what you decide to put here.

Level Menu : Edit Animation Settings

Since there may be levels where you will want to save on VBlank time, processing time, or tile/palette space, this dialog will allow you to set whether or not certain animations are enabled in a particular level.

You can disable the game's original animated tiles, the original palette animation (for color 64, used in the Yoshi Coins), Lunar Magic's global animations, or Lunar Magic's level animations. Note that even if LM's animations are disabled, any trigger initializations you've set up will still activate on level load.

Lunar Magic's animations are disabled by default in level 104 to prevent it from interfering with the end game sequence, but you can chose to re-enable them if you want.

Level Menu : Modify Screen Exits

In Super Mario World, every "screen" in a level can have an exit destination, which is used by any exit-enabled pipe or door within that screen. If no exit is set, it will lead to an undefined level, which is usually (but isn't guaranteed to be) the bonus game in level 0 or 0x100. Which is why it's important to remember to use exit-enabled pipes and doors only on screens that have an exit set up.

In the level editor you can use the "View Screen Exits" menu command to view the area covered by all the screen exits in the level (or just hit F1). Be careful with placing an exit-enabled object right beside a screen boundary, because there's a chance the next screen's exit may be used instead depending on the player's coordinates when they enter.

There are two types of exits you can use. The first type (standard exit) will allow you to specify the level destination and will simply lead to the destination level's main or midway entrance. This is the same entrance used when entering a level from the overworld. The second type (secondary exit) allows you to specify a Secondary Exit Number. This number is used as an index into a table of secondary exits stored in the ROM which are shared between all levels. While there is only one main entrance and one midway entrance to a level, you can have multiple secondary entrances to the same level.

The first drop-down box in this dialog indicates the **screen number** of the screen exit you want to examine or modify. If you want to clear the exit, then hit the **Clear Screen button**. Or if you want to clear the exits for all screens in the level, hit the **Clear All Screens button**. Just remember that this doesn't remove exit-enabled pipes or doors on that screen... you still have to remove those yourself. If you instead want to create a new screen exit or change the destination level of one that already exists, then just fill in the **Level Destination or Secondary Exit Number**.

If the **Use the above Value as a Secondary Exit Number** option is enabled, the exit for this screen is treated as a Secondary Exit (See the "Modify Secondary Entrances" menu for Secondary Entrance/Exit settings). If it isn't enabled then this is a standard exit, and you can find the settings for it in the "Modify Main and Midway Entrance" menu when you load the destination level in Lunar Magic for editing.

If this exit is a secondary exit, you can also select whether to **make the destination level a water level** or not. This is an older option that's still here for backwards compatibility. The secondary exits now have their own option to set this.

If the **Go to midway entrance** option is enabled, the exit for this screen will go to the target level's midway entrance. Note that this option won't work if the target level has not enabled the "Use separate settings for midway entrance" option or has redirected the midway entrance to another level.

Note: There used to be an old rule in SMW where levels from 0XX were not able to exit to levels from 1XX and vice versa. This also applied to Secondary Exit Numbers. While Lunar Magic can modify the ROM to eliminate this limitation, the new behavior is dependent upon Lunar Magic converting the exits in a level from the old style to the new style; the old style exits in unmodified levels still follow the old rule. This means that if you use a new exit to go from one level block to another, and then use an old style exit in that level, the old style exit will unexpectedly divert you back to the original level block. But this can easily be solved simply by resaving the level that contains the old style exit so LM can automatically convert all the old style exits in the level to use the new enhancement.

Level Menu : Modify Main and Midway Entrance

This menu dialog will let you change the position of the main and midway entrance. While you can just drag the Mario sprite in the level editor's sprite editing mode to move an entrance, there are still some additional options in here that you'll want to make use of.

There can only be one main level entrance, and one midway entrance. In the original game the two are tied together, due to the fact that other than the screen number, they share the same settings for X/Y Screen Position, Mario Action, etc. However Lunar Magic adds an option to use separate settings for the midway entrance, for more flexibility.

Note that you don't have to make use of the midway point entrance. If you don't want the player to be able to start at the midway point, just don't put a midway point tape in the level. While you can use the midway point tape object in any level, when you re-enter from the overworld it will normally use the midway entrance of the level number you're entering, which isn't necessarily the level number that you obtained the midway point in. If you want to change that, in the level you enter from the overworld you can enable the **Redirect Midway Entrance to other level** option, and fill in the level number of the midway entrance you want to use. Just be careful not to cause an infinite loop with this option, as it's possible to chain the redirects.

The **Screen Number** of the entrance can be any screen (0-0x1F). The **X/Y position** refers to the position Mario will enter the level. These are relative to the screen number (or the sub-screen of the screen number if not using method 2). The **Additional Pixels (P)** value is always zero (unless you hack the table itself in the ROM), so you can ignore it. The **Sub-Screen (S) value** refers to which half of the screen the X/Y coordinates are relative to. For vertical levels a 0 indicates the top, and a 1 indicates the bottom. For horizontal levels, a 0 indicates the left and a 1 indicates the right. To view the Sub-Screen Boundaries, just hit F2 in the main editor.

There are two methods for setting the X/Y position...the first one has table-based coordinates and is used by the original SMW levels. When using this method, keep in mind that the S value of X is used for vertical levels, while the S value of Y is used for horizontal levels. **Method 2** does not use table-based coordinates, and is an enhancement inserted by Lunar Magic. The second method is the more flexible of the two, and is practically required for reaching most areas in horizontal levels that are taller than the original game.

The **Layer 1, 2 (FG,BG) initial positions** set up what the initial FG and BG vertical position offsets will be when Mario enters the level. These should be set depending on your Y position. A value of 0xC0 is used for positions near the bottom of the screen, 0x60 for midway positions, and 0 for the top.

The **Set FG/BG relative to player** option is a new method added by Lunar Magic, and is nearly required for horizontal levels that are taller than the original game. Instead of using fixed positions, the FG position will be set relative to the player and an offset that you can specify. The BG position will be set relative to the FG position, scroll settings, level height, and BG height. The BG is basically set so that when the player reaches the bottom of the level, it will show the bottom of the BG image.

In horizontal levels, Lunar Magic displays the background tiled in the level editor if it determines that the background will loop vertically based on the level's layer 2 scroll settings and BG height. You can adjust both of these in the Change Other Properties dialog. Either use a taller BG or use a slower vertical scroll setting if you want the background to be able to cover the whole level vertically without looping.

In levels where layer 2 is set up to have objects, you should normally have the FG and BG set to the same position so that the two layers are aligned with one another. If you're using the relative FG/BG option, this is done by setting the BG to +00 in the Change Other Properties dialog. If you're using a layer 2 scroll sprite that requires the BG position to be set to 0 and you're using the relative FG/BG option, this is done by setting the BG to "Absolute 0" in that same dialog.

Keep in mind that it's possible for the player to die instantly when they enter the level if you start the FG initial position too far away from the player X/Y position. And of course, the player will also die instantly if you put the X/Y start position right in the middle of a solid brick wall, or put the start position on a screen number that's beyond the current level's maximum number of screens setting.

The **Mario Action** drop-down list allows you to determine what Mario will do when he enters the level. This could be nothing, or it could make it appear that he's coming out of a pipe. With the "nothing" options, Mario will not be able to carry an item in his hands from another level into this one. The options below this list allow you to set the level to be **slippery**, whether Mario will be forced to swim through the level as though it were filled with **water**, and if Mario should face the **left direction**. If you use the "Shoot from Slanted

Pipe Right" option combined with the "Face left" option, the player will shoot left instead.

You may notice that the last "Mario Action" in the list is a bit different for midway entrances compared to the main entrance. This one will do nothing but still make it a water level if used as an actual midway entrance, but will do "Vertical Pipe Exit Down (water level)" if accessed via a screen exit. To avoid confusion, it may be better to instead use the "Vertical Pipe Exit Down" action with the separate "Make this a Water Level" setting.

Level Menu : Modify Secondary Entrances

The dialog of this menu will let you edit all the entries in the Secondary Entrance/Exit table in the ROM. This table is shared between all levels.

The drop-down list at the top (referred to as the “**Secondary Entrance/Exit Number**”) is where you'll find the index number for each entry. A Secondary Entrance and a Secondary Exit are essentially the same thing and refer to the same table, it just depends on whether you're looking at it from the destination level's perspective or the source level's perspective.

In the drop-down list, any entrance that is already used will have the words "to level XXX" or "to Overworld [XXX]" beside the entrance number, where "XXX" is the level destination (although for the overworld ones, the level destination is only used to determine which level the entry will be exported with when saving an MWL file). A secondary entrance can be referenced multiple times by multiple levels... meaning that all exits that use that secondary entrance number will go to the same destination. The “**Clear Slot**” button will clear the settings for the current entry, while the “**Clear All Slots**” button will clear the settings for all entries in the table.

Any changes you make to entries in the table will be lost if you switch levels in the main editor without saving the current level to the ROM first. Or you can save the level to an MWL file, and it will export any secondary entrance to the MWL file whose destination level number matches up with the level you're saving. When you open up the MWL file the secondary entrances are imported back into the table, but will not be saved to the ROM until the level itself is saved to the ROM.

The **Overworld Action** section is an addition by Lunar Magic to the game that lets you use a pipe or door to exit to the overworld instead of a level. To use it, you must first enable the **Exit to overworld** checkbox. In the drop-down list you can then choose to "Do Nothing" (which would just make you go back to the overworld at the same place you were at before), pass the level via one of the specified exits, or to end your turn by switching to the second player in 2 player games.

You can also combine one of the above actions with the **Teleport to location index** option, and specify the location index to use. Make sure the overworld has been saved at least once in Lunar Magic for these indexes to work. You can change where on the overworld these indexes go in the overworld editor's Layer 1 16x16 mode using the "Set Location Teleport" menu. If you intend to combine this with passing a level, there are two things you should be aware of: 1) The direction to enable for walking will come from the overworld tile the player is standing on *after* teleporting, and 2) The base event used to figure out which event to pass will come from the tile the player is standing on *before* teleporting... unless you enable the **Use a different base event** option and specify a different base event to use.

The remainder of the items you'll find in this dialog are pretty much identical to the ones you'd see for a main/midway entrance, so check the "Modify Main and Midway Entrances" menu for information on the rest.

Level Menu : Scan for Undefined Exits

One of the more common problems that people have when editing Mario World is that when they use exit-enabled pipes or doors, they often forget to set up an exit destination for the screen number the object is on. If a player then enters that pipe or door, they usually end up going to level 0 or 0x100 and getting trapped in an endless bonus game.

Thus this menu item will scan the current level for any exit-enabled pipes or doors that lead to levels 0 or 0x100, and give you a list of the screen numbers the objects are on. You should then either delete or replace the exit-enabled objects in question, or set up an exit destination for that screen.

The keyboard shortcut for this menu item is ctrl+shift+A.

There is also the Scan Exits on Save to ROM option to do this scan automatically whenever you save a level to the ROM.

Level Menu : Change Properties in header

In the dialog for this menu, you can change various other header properties that didn't fit in any other menu. This includes the Layer 1 (FG) vertical scrolling, number of level screens, item memory index, and the SNES register/level modes.

An explanation of most elements in the "**SNES Registers and Level Modes**" setting is beyond the scope of this document. If you really want to know about the registers, look them up in an SNES hardware document. As for the Level Modes, these determine everything from whether the level will be Vertical or Horizontal to the type of screen level memory map used.

If you're going to play with this setting, I suggest you save your level first. A quick summary of the level modes is as follows:

- The Non-Layer 2 levels are modes 0,C,E,11,1E (horizontal levels), and modes A,D (vertical levels).
- The Layer 2 levels are modes 1,2,F,1F (horizontal levels) and modes 7,8 (vertical levels).
- Modes 3,4,5,6 are peculiar mixed memory map modes, which Nintendo apparently thought better of because they're never used AFAIK. Lunar Magic partially implements them, but layer 2 will be in the wrong position. Using these is not recommended.
- Modes 9,B,10 are for Boss battles, which will bypass the standard object level data. Lunar Magic will not show anything except a red gradient background. These have NULL memory maps, so Lunar Magic won't let you place any objects.

- Modes 12-1D are unused and have NULL Memory Maps. Lunar Magic will not allow you to select these, since crashing emulators are detrimental to your computer's health.

In modes 1, 7, and F Mario and the other sprites cannot "touch" or otherwise interact with the objects on layer 2. And modes 1E and 1F have translucent layers, which were never used in the original game.

The **Horizontal Level Mode** list lets you change how high a horizontal level will be in 16x16 tiles. This tends to change both of the level's dimensions, as the higher you make the level vertically the fewer screens you'll have horizontally. Remember that if you have layer 3 tides or a layer 2 level, it will cut the number of horizontal screens you can actually use in half.

The **Allow viewing full bottom row of tiles** option for horizontal levels will let the game scroll the full bottom row of tiles into view. In the original game, only the top line of pixels for this row was visible. It will also do the same for the background image.

The **Layer 1 (FG) vertical scrolling** drop-down list gives you several options to control when/if vertical scrolling should occur. The last option also affects horizontal scrolling.

The **number of screens** value will determine how many level screens to allow scrolling of before stopping, although this may be overridden by other scroll settings. The Auto-Set Number of Screens option sets this for you automatically.

The **item memory index** value controls which area of RAM the SNES should store a temporary record of the coins and other objects that a player has gotten in the current level. So the general idea here is to use different index numbers for other levels that your main level has exits to. Typically, you'd use 0 for the main level and use 1 or 2 for other levels accessed by pipes or doors. You can technically share the same number between two levels, but you run the risk of having the game mistakenly thinking that certain coins and such are already taken, so they may end up being mysteriously gone when the player enters the room. You can still safely get away with it in some cases though. Index 3 was available but unimplemented in the original game, however an ASM hack by BMF inserted by LM allows you to use this index for levels where you don't want the items collected to be tracked (which means they'll always respawn).

Level Menu : Change Properties in Sprite header

The dialog you get from this menu will let you change the options in the sprite data header.

The drop-down list referred to as the **Sprite Memory** list controls how memory is allocated for the sprites. This list doesn't show the actual memory requirements of each sprite and the memory allocation setup of each list index, due to the complexity involved. If you have problems where the sprites in Lunar Magic look fine, but in the ROM portions of sprites are becoming invisible or "morph" completely into other sprites on the screen, it means you need to change this setting. There's actually only 0x13 possible settings, so just play with it a bit.

What this drop-down list does show however is the memory index range of each selection... and from this it can determine the maximum number of allowed sprites that the ROM will render on the screen at once (this is the **M value**). The **SP value** lists 2 sprites that are given "special" treatment... these are usually sprites that take up a larger than normal amount of memory like a Big Boo, Banzai Bill, etc. A value of FF for one of the SP values means the value isn't used. The **M1 and M2 values** describe the range and maximum number of allowed sprites per screen for the two "special" SP sprites. These max values are usually separate from the M1 normal sprite max, unless of course the ranges overlap.

For an example of how to read this data, take a look at Level 105, which uses sprite index 04. With the index used in this setup, a maximum of 7 sprites are allowed on the screen at once, in addition to one "special" sprite which can be either sprite 60 (...?) or sprite 9F (Banzai Bill).

The **Sprite Vertical Spawn Range for Horizontal levels** list lets you control the height of the spawn/despawn range for sprites. In other words, this lets you set just how far beyond the top and bottom of the screen that sprites should be allowed to spawn/despawn. A smaller range can reduce slowdown and means you're less likely to reach the maximum number of allowed sprites on screen. The default is "Horizontal Level", which is intended for compatibility with the original game's horizontal levels and allows approx 12-13 tiles above the screen. The "Vertical Level" setting is much shorter (3-4 tiles above), which can be useful for sprite dense levels but might look slightly odd for a few bigger sprites that are taller than the spawn range as they may look like they're popping into view. The "Enhanced Vertical Level" setting is a compromise between the two previous ones (8-9 tiles above). The "Infinity" setting can span the height of the level, which you most likely won't want to use except in very special circumstances.

The **Enable Smart Spawn** option will basically prevent spawning more sprites from the sprite list unless you're scrolling the screen, which can also improve performance.

The **Enable sprite buoyancy 1** option determines whether sprites other than Mario will be allowed to interact with water and lava (but cave lava tiles 0x159-0x15B don't seem to need it). For some sprites, this might be rather minor like causing a shell to splash when it hits water/lava or slowing down a falling spike. But for other sprites like the dolphins and fish, it will determine whether or not they are allowed to float and swim. And an extreme case would be the vertical fireball, which causes the game to freeze unless buoyancy is enabled! Unfortunately, enabling this option will increase the amount of processing the game has to do, which will decrease the number of sprites you can have on the screen at once before slow down becomes noticeable during game play.

The **Enable sprite buoyancy 2** option works much like the first, except it also disables all other sprite interaction with layer 2/3 to somewhat reduce the amount of processing the game has to do (assuming you have a level on layer 2 or tides). In other words, sprites will walk right through solid blocks placed on layer 2, but they'll still swim in water placed on any layer. Mario's ability to interact with layer 2 is not affected by this setting… that's determined solely by the Level Mode setting.

The **Make sprites beyond level boundaries interact with air instead of water** setting was added mainly for tides. In the original game sprites beyond the level boundaries would always interact with water, probably to prevent water sprites from falling away into the void when near or past the edge of the level. But tides can already extend beyond the level boundaries (unless you're in a vertical level or a 1 screen wide horizontal level), so it's not as necessary there. Making it air can help prevent some sprites that float on water surfaces from levitating when they reach the level edge. Plus in a tide level that allows vertical scrolling, it can aid in scrolling and prevent sprites below the screen from behaving as though they're in water (alternatively you could change the sprite vertical spawn range to "Vertical Level (short)" to minimize that effect... but if you have a lava tide that can kill sprites, it won't be enough).

Level Menu : Change Other Properties

This will let you change various other properties of the level. This includes the Layer 2 (BG) scrolling rate and initial position for the relative method, "vertical level positioning" option, and the "No Yoshi level intro" option.

The **Layer 2 (BG) scrolling rate** let's you change the rate that the background is scrolled, both vertically and horizontally. The speeds are relative to the layer 1 (FG) scrolling rate. “None” means the BG will not scroll at all and will remain fixed in place. “Constant” is 1:1 which means it will move at the exact same speed as the FG (this is usually used when you have level objects on layer 2). “Variable” is 1:2, so the BG will scroll half as fast as the FG. “Variable 2” is 1:4, “Variable 3” is 1:8, “Variable 4” is 1:16, “Slow” is 1:32, and “Slow 2” is 1:64.

The **BG Height in Tiles** setting is used when determining the **Layer 2 (BG) Initial Position** for entrances that set the FG/BG relative to the player. It doesn't affect the actual height of the background... rather it helps set the scroll position so that when the player reaches the bottom of the level, it will show the bottom of the BG image.

The **Set BG relative to FG only** setting causes entrances that set the FG/BG relative to the player to *only* take the FG position into account along with adding the offset provided in the dropdown box below it. Typically you'd use this option for levels that have level data on layer 2 instead of an image.

See the Modify Main and Midway Entrance (FG/BG relative option) section for more information on how to use the layer 2 settings here.

Vertical Entrance Positioning should be enabled in a vertical level and disabled in a horizontal level, or your level entrances could end up somewhere in la-la land, instantly killing any player that tries to enter through them.

The **Unknown Vertical Level** option is typically enabled when the vertical entrance positioning option is also enabled, but it doesn't seem to get used for anything. Most likely it was just meant to signal when layer 2 is vertical while the first option would be used for layer 1. But Nintendo ended up not using the mixed level modes, so it has no real use.

The **Disable 'No Yoshi' Level Intro** option will prevent certain FG/BG index settings (like for ghost houses and castles) from showing a short intro where Mario is forced to jump off Yoshi before entering a level from the overworld. The intro is activated by FG/BG index settings of 1, 2, 5, 6, and 8. Also note that if you allow the player to obtain a Yoshi in a level which displays a no-Yoshi intro, the Yoshi the player is riding on will vanish whenever the player exits to another level.

Level Menu : Change Layer 3 Settings

The **Layer 3 options** allow you to set which of the original game's layer 3 tilemaps should be loaded. You have the option of a blank tilemap, water tides, or a tileset specific image.

Note that the actual tilemap displayed can be bypassed from the Layer 3 GFX/Tilemap bypass dialog. But the behavior and scrolling of the original setting will remain unless you enable the advanced bypass settings.

The tides cannot be used on a layer 2 level because it uses the same RAM that layer 2 uses. And just like with layer 2 levels, the maximum number of screens available will be cut in half. Be careful not to let objects place tiles beyond the right side of the level or this will interfere will how the tide acts. To use tides in a vertical level, you must enable the advanced bypass settings (minimum level height of 2 screens required). To use tides in a horizontal level that's only 1-2 screens wide (not supported for HD version), you must enable the advanced bypass settings.

For water/lava tides (excluding cave lava), make sure to turn on the second sprite buoyancy option in the Change Properties in Sprite Header dialog. And for most tide levels you should usually turn on the option to make sprites beyond level boundaries interact with air instead of water in the same dialog.

The tileset specific option is normally only used on certain FG/BG tilesets. The possible tilesets are 1 (windows), 3 (rocks), 5 or D (mist), 9 (goldfish), or E (scrolling rocks). All other tilesets will give you the beta cage tilemap, which uses graphics that are not in the released game.

The **Make tides act as** setting will let you change which tile the tides will act like. This can be water (default), lava (surface tile kills sprites while tile under it kills Mario, compatible with floating lava platforms), Mario lava (kills Mario only, is water to sprites), cave lava (a special type of lava in caves, requires tileset 3 or E to work), solid tile, or tiles 0x200-0x20A (for custom behavior).

The **Force layer 3** flag will show layer 3 tiles that have the priority bit enabled above all other layers and sprites. This will not work if layer 3 is only enabled on the subscreen (TS) while the layer you want it to appear above is on the main screen (TM). In the original game the tile priority bits were only enabled for some layer 3 tilemaps (tides, mist, and goldfish).

The **Enable advanced bypass settings for layer 3** checkbox will let you change certain advanced characteristics of how layer 3 scrolls or is drawn.

The **CGADSUB for layer 3** checkbox will determine if layer 3 will appear translucent against layers on the subscreen. This of course assumes that layer 3 is on the main screen (which it normally is), and that there's any layers on the subscreen to appear translucent against with the current level's settings.

The **Move layer 3 to Subscreen** checkbox can allow layer 3 to appear behind layer 2, as layer 2 is often also on the subscreen. But this can also cause the layer 3 status bar to appear below layer 1, so the option is mainly for people using the third party sprite status bar patch. Although it'd also be possible to fix by creating a custom patch to put layer 3 on the main screen during NMI and restoring the layering registers during the status bar IRQ.

The **Fix layer 3 scroll sync issue caused by status bar** checkbox is to fix a minor glitch with the original game. Basically, layer 3 scrolling is not in sync with the other layers. This is a side effect of the layer 3 status bar IRQ, which causes the scroll values used to be one frame ahead of the other layers. It's hard to notice though unless you have sharp eyes or layer 3 is moving at or nearly the same rate as another layer, so most of the time you can just leave it off. If you're using a custom patch to move your status bar to the bottom of the screen, or you don't have a layer 3 status bar at all such as when you use a sprite status bar, then there is no sync issue to fix and the option should be left off (in fact turning it on in this case would *cause* a sync issue).

The **Vertical Scroll** and **Horizontal Scroll** drop down boxes have various options for how to scroll layer 3. "None" means layer 3 won't move. "Constant" means it will scroll in place with layer 1. "Variable" is 1/2 the position of layer 1, "Variable 2" is 1/4, and "Slow" is 1/32. "Fast" is 1.2, which means it goes faster than layer 1 (this is intended for the type of foregrounds you can see in SMW2:Yoshi's Island). There are also several auto-scroll options available. The slow speed is the same one used for fog/goldfish, while the fast one is the same speed used in tides.

The **Initial X/Y Position/Offset** lets you adjust which part of layer 3 you want to be shown. These are in units of 16x16 tiles. The Y setting can be from -400 to 3FF. To make things a bit easier the Y position for horizontal levels automatically changes when you change the vertical scroll setting to match how the game does layer 2. Note that for the "None" and the auto-scroll settings (except in tide levels with vertical auto-scroll), these values are actual positions. For the other settings, they're used as offsets added to layer 1's position.

If using tides with a constant vertical scroll setting, most of the time you'll use a negative Y offset value to set the height of the tides in the level. In a horizontal level if you intend to use a positive Y offset of 1-5 or higher, a minimum level height of 0x1C-0x20 tiles is required.

Remember that the layer 3 tilemap is shared with the layer 3 status bar (it occupies 32x5 tiles on the top left half of layer 3). So unless you're using something like the third party sprite status bar patch, take care not to use vertical scroll and Y offset settings that could result in a second copy of your status bar being scrolled into view. If you need to use a vertical auto-scroll setting, you can get around the issue by setting the horizontal scroll to none and the X Position to 0x10. This would cause only the right half of the layer 3 tilemap to scroll by.

Level Menu : Change Music & Time Limit Settings

The **music setting** in the level header doesn't list all the music tracks available for levels to use. So if you don't find what you want in the first list, you can enable the option to **bypass the music header setting** and specify any of the valid level music tracks to play. Be aware that anything labeled with (SFX) does not loop. And be careful not to use the ones listed as "Not used" unless you've replaced them with your own custom music track, since there's a chance the game will freeze up.

The **time limit setting** also only offers limited choices in the level header, so you can choose to enable the option to **bypass the time limit header setting** if you want to specify a **custom time limit** value. You can use a value from 0-FFF for the time limit, although I don't think people would normally expect to see a time limit with hexadecimal digits. Note that 0 is actually infinite, but if Mario dies in the level you'll get a “time-up” message from the game (to use 0, you have to use the level header setting rather than the bypass unless you're using the force option). The **force time limit reset** setting makes the time limit reset to this value whenever the player enters the level…not just from the overworld map, but from other levels as well!

Options Menu : General Options

Program Options

Mouse Gestures: This enables Opera-like mouse gestures using the right mouse button. Read the mouse behavior section for more information on how to use the mouse gestures.

Auto-Save on Mouse Gestures: This will cause the current level to be saved if it has been modified whenever you switch levels using a mouse gesture.

Remember Window Size: Enabling this option will cause Lunar Magic to save the size of the main and overworld editor windows to the registry when you close the program, so that the windows will be the same size when you reload the program again later.

Disabling it will cause the default window size to be used the next time you run the program.

Show Old/Obsolete Menu Items: This can toggle display of older or obsolete menu items, some of which haven't been in use since the early years of SMW hacking (such as list based GFX bypassing). Requires program restart to take effect.

Show ID in Add Object/Sprite Editors: This can toggle display of the sprite/object ID and object size info in the Add Object/Sprite windows. If the Add Object or Sprite window is already open when you change the setting, you may need to change the group that you're viewing to refresh the list.

Standard GFX Bypass Dialogs: Turning this option off will cause Lunar Magic to use alternate GFX bypass dialogs that use regular edit fields instead of lists. A single list will still be included in the dialog to allow checking on the insertion status of each graphics file. This option is considered obsolete as you can also now type in the list boxes in the regular dialog.

Auto-Deselect on Editor Select: If this option is enabled, whenever you make a new selection in the Add Object/Sprite windows or the Map16 editor window, anything selected in the main level editing window will be deselected. This can make it easier to paste into the main window, since then you don't have to always remember to hold down the control key.

Allow Control + Mouse Wheel to Zoom: This option will enable or disable the ability to zoom by holding down the control key and using the mouse scroll wheel.

Wrap Toolbars on Main Windows: This option will cause toolbar buttons of the level editor window and the overworld editor window to wrap if there isn't enough room to display them on one row. May require program restart to take effect.

Open Levels at Main Entrance: When you open a level, this option will cause the program to scroll to the screen of the main entrance.

System DPI Aware: This option will cause the program to notify the OS that it's system DPI aware, to avoid fuzzy scaling sometimes done by Windows Vista and later when possible (scaling is still done if the window is dragged to a monitor with a different DPI). If disabled, it's left up to the OS how it wants to handle scaling. Requires program restart to take effect.

Use DirectX for Main Windows: This option will make the level editor window, background layer 2 editor window, and the overworld editor window use DirectX 9 if available, which allows using zoom filters. It uses far more RAM however, and may be more taxing on very old or weak hardware.

Pause Animation When Inactive: If the program loses window focus, this will cause animations to be paused to save on CPU usage.

ROM File Name in Main Window Title Bar: This option causes the program to show the ROM file name in the title bar of the main editor window.

Show Other Tiles in 8x8/16x16 Editors: Enabling this option will cause the 8x8 and 16x16 editors to display other tiles such as animated source tiles, layer 3 tiles, external graphics tiles, and some internal tiles used only for display in the program such as Sprite Map16. May require program restart to take effect if disabling.

Max Undo: This controls the maximum number of undo operations that will be allowed. Keep in mind that the higher the limit, the more RAM the program will consume. Note that computers with low RAM (64MB or less) will be restricted to 8 or less for this setting.

Warning Options

Save Prompt: Whenever you try to open up a new level/ROM without saving the changes to the current level/overworld, a save prompt will appear asking if you want to save or cancel. Since more experienced users may find this annoying, you can turn the behavior off.

Scan Exits on Save to ROM: This option will cause Lunar Magic to automatically scan your level for exit-enabled objects that do not have an exit destination set whenever you save your level to the ROM. If it finds any, it gives you the option of aborting the save.

You can also do this scan manually with the Scan for Undefined Exits menu item.

Count Sprites on Save to ROM: This option will cause Lunar Magic to automatically count the number of sprites in the level whenever you save it to the ROM. If it finds that you've exceeded the safe maximum limit that the game can process, it gives you the option of aborting the save.

The game can have up to 128 sprites in a level (or 255 if you're using Vitor's SA-1 patch or some other 3rd party sprite programs). Exceeding this limit may cause the extra sprites to not show up during gameplay, cause the game to freeze or display random sprites when the player reaches the screen where you've exceeded the limit.

This limit does not affect Lunar Magic's ability to save/load the sprites for the level, as the editor can process far more sprites than the game can use.

Check Object Placement on Save to ROM: This option will cause Lunar Magic to automatically check that no objects are placing tiles beyond the viewable level area in unsafe locations whenever you save a level to the ROM. If it finds such an object, it gives you the option of aborting the save.

Depending on the size of an object and where it's placed, it's possible for an object to place some of its tiles beyond the RAM that SMW set aside for the level tile map. When this happens, unintended SNES RAM locations may get overwritten during gameplay. This can be avoided by not moving/resizing objects in such a way that they go off the top/left side of the first screen or the bottom/right side of the last screen in the level.

Check if ROMFileName.ips Exists: If a file with the same name as the ROM but ending in .ips is detected in the same folder as the ROM, this option will display a warning message when saving. This is because some emulators will automatically patch the ips file to the ROM when loaded, causing changes in the editor to not show up in the emulator or other problems. It's recommended that you rename or move the IPS file to avoid this.

Check if Vertical Fireball has Buoyancy: Sprite 33 (vertical fireball) requires that sprite buoyancy in the Change Properties in Sprite Header menu be enabled, or the game will usually freeze. This option will check if it's enabled when the sprite is used in the level, and gives you the option of aborting the save.

Notify if RATS.log is written to: This option will notify you if something was written to RATS.log, which contains information on issues encountered with RATs on saving data to the ROM.

ROM Editing Options

Correct Fatal Errors: If you ever edit the level data manually or paste an incompatible object from a level with a different tile set into another, it's possible that this will result in certain invalid byte sequences that could cause your emulator to freeze or even crash when it loads your level.

Keeping this on allows Lunar Magic to modify the data to something non-fatal, and alert you about the problem.

Maintain ROM Checksum: This option will cause Lunar Magic to automatically maintain the ROM's original checksum whenever you insert data for the game. Since SNES emulators don't particularly care about checksums, it's not really necessary to keep this on. However it can be useful for other people to verify that a patch has been applied correctly when you distribute your work.

Turning this off may improve the speed of saving data back to the ROM somewhat.

Auto-Set Number of Screens: This automatically sets the number of screens the level can scroll to when you save to the ROM based on the last layer 1 screen that Lunar Magic detects an object or sprite on. If you prefer to set this manually for some reason, you can turn this off and set the number of screens in the "Change Properties in header" menu.

Use Joined GFX Files: Enabling this option will cause the menu commands dealing with GFX insertion and extraction to use a single file called "AllGFX.bin" instead of multiple *.bin files. It may be useful if you don't want to search through each file individually for tiles you want.

This option does not apply to ExGFX files.

Convert Sprite Berry GFX Tile: The original game has a berry tile in sprite graphics (first tile in slot SP2, GFX files 1 and 31) used for when Yoshi eats a berry. But since the original game uses 3bpp graphics, this tile has to be converted to use the upper half of the palette so the correct colors will be shown.

Lunar Magic will normally convert the tile for you when it inserts or extracts the GFX. However, there may be times where someone will want to use this tile for something else.

In that case, you can turn off this option to prevent the conversion from taking place.

Prefer Saving in 2MB+ ROM Area: This option will cause Lunar Magic to always first try saving data and code past the 2MB mark in the ROM before using earlier space. In LoROM games the area above 2MB is mapped to banks 40-7D/C0-FF, which in the SNES memory map does not mirror RAM 7E:0000-7E:1FFF to xx:0000-xx:1FFF like it does in lower banks. Unfortunately many 3rd party ASM patches are not written to take this into account, and will malfunction without warning if installed to banks 40+.

Since Lunar Magic's ASM hacks for the expanded ROM area can run from any bank, this option will cause LM to first save in the bank 40+ region to fill that area up before using the lower area. This will help save space below bank 40 for less versatile 3rd party ASM patches/tools to use.

If the ROM is not yet larger than 2MB, this option will have no effect. Therefore if you plan to use several 3rd party ASM hacks or tools, you may want to expand your ROM early on so LM will store as many things as possible past the 2MB mark.

As SMAS+W is already larger than 2MB, this option will have no effect for that game. The option is also ignored for SA-1 ROMs, since for SA-1 LM already saves at 4MB+ first, then 2MB+, then lower.

Silently Add Header to ROM: This option will cause Lunar Magic to automatically add a 0x200 byte copier header to your ROM if it's missing when you open a ROM, since it's required for the program to operate. If this is disabled, the program will prompt you each time the header is missing to add one.

Options Menu : Restore Point Options

This dialog will let you control how Lunar Magic handles Restore Points.

"Create Incremental Restore Points for all saves" will cause the program to record changes done by Lunar Magic with each save. If the ROM is modified outside of the program, it will automatically trigger an extra full restore point to record those changes as well.

Note for 3rd party utility programmers: you can include information to be used in Lunar Magic's restore point description for these automatic full restore points. Simply open up ROMFileName.extmod (if the file exists append to it, else create it) and print out the name and version of your program in UTF-8 text. You can also optionally include a short description of the changes performed. Lunar Magic will read the contents of the file and put them in the restore point description, then delete the file.

"Create Full Restore Points based on number of saves" will cause the program to create a full restore point after a certain number of saves, which you can specify. Note that if a full restore point is created by other means (either automatically or manually), then the count for this is reset.

"Always create restore directory" will make the program still create the SysLMRestore directory even if all the other restore options are off. It's recommended to leave this on so you'll always have a copy of the original ROM available.

"Include auxiliary ROM files in restore points" will cause Lunar Magic to track and include changes done to the extra ROM files used by the program (ROMFileName.msc,

.dsc, etc). These aren't required however, and they do take up more space in the restore file so you may want to turn this off if you only care about the ROM itself. You can also choose to omit these files when restoring your ROM to a previous state even if these files are included in the restore point.

"Scan ROM for nested RATs on Full Restore Point creation" will cause Lunar Magic to do a scan of the ROM user area just before creating any full restore point. The scan will check to see if there are any nested RATs in your ROM. A nested RAT is often an indication that a third party tool or patch has misapplied changes to the ROM, resulting in data loss or corruption. If these are detected the program will warn you that you should restore your ROM to a previous state, before the tool or patch was used.

Options Menu : Animation Rate

This will allow you to change the rate that Lunar Magic displays animations from the game at. The lowest setting is 7.5 fps, which is the default used in versions prior to 1.70. Since SMW's tile animations are 7.5 fps, this is generally adequate although the Yoshi coin palette animation will be at half speed.

15 fps has been the default since version 1.70 of Lunar Magic. At this speed, the yoshi coin animation will be correct and you can view ExAnimations that use the double speed trick. This is usually sufficient for most users.

30 and 60 fps will let you view ExAnimations that use faster speeds. 60 fps will also let you check to see if your animations will actually overwrite the game's original animations, and results in a smoother experience when using the internal emulator. If you aren't doing any of these however, you don't need to set it this high. The 30 and 60 fps settings can't be used on Win9x/ME... they require Win2k or higher.

Remember that each setting requires twice as much computing power as the one before it, but it also depends on the size of the program's window. If you have a half-decent system though, it shouldn't matter much. With Lunar Magic maximized on a 1024x768 screen on a Core2 2.4 GHZ computer and animation set to 60 fps, it doesn't even take up 1% of a single core. However performance on Windows Vista will be lower due to Microsoft's removal of hardware acceleration for GDI functions (which was re-added on Windows 7 and later). Performance on Windows 7 may vary depending on your video card and driver and whether Aero is enabled.

Options Menu : Use FastROM Addressing for this ROM

This option causes all ROM addresses Lunar Magic saves for data and ASM hacks to use the $80:8000 LoROM map, which is required for getting the speed benefit out of using FastROM. However this does not actually enable FastROM speeds or convert the rest of the original game to use FastROM... to do that you should also turn on the Enable FastROM Speed & Apply Patch setting (or alternatively apply a 3rd party hack to enable it).

Using this does have one small drawback however. Once it's been turned on and applied to the ROM, you will no longer be able to expand your ROM past 4 MB to ExLoROM. Few use that type of expansion though... you may want to consider using SA-1 instead of FastROM if you want both more speed and larger ROMs.

While you can technically enable this at any time, for maximum benefit it should be enabled on a fresh ROM before saving anything since ASM hacks and data that Lunar Magic has already inserted are not modified.

This option is not available if you're using ExLoROM or a special chip (SA-1 or SuperFX).

This setting is saved to your current ROM.

Options Menu : Enable FastROM Speed & Apply Patch

The next time you save a level to the ROM, this option will cause Lunar Magic to modify most 24 bit ROM addresses in the original game to use the $80:8000 LoROM map using a list provided by Ersanio. It also inserts a small ASM hack to enable FastROM. As this allows the game code to run slightly faster, it can potentially reduce game slowdown.

Using this does have one small drawback however. Once it's been turned on and applied to the ROM, you will no longer be able to expand your ROM past 4 MB to ExLoROM. Few use that type of expansion though... you may want to consider using SA-1 instead of FastROM if you want both more speed and larger ROMs.

This setting can be enabled at any time. But once the changes have been applied to the ROM, turning it off will not reverse the modifications. For additional performance, you should also turn on the Use FastROM Addressing for this ROM option.

This option is not available if you're using ExLoROM or a special chip (SA-1 or SuperFX). And it's only available for the SMW US ROM, not the Japanese version or SMAS+W.

This setting is saved to your current ROM.

Options Menu : Use SA-1 RAM Remap for this ROM

This option only affects SA-1 ROMs. It causes certain RAM addresses Lunar Magic uses in ASM hacks to be remapped to SA-1 RAM, to be compatible with the SA-1 patch by Vitor Vilela. The areas affected are :

$7E:0000-$7E:00FF to $XX:3000-$XX:30FF (Direct Page)
$7E:0100-$7E:1FFF to $XX:6100-$XX:7FFF (shadow RAM)
$7E:C800-$7E:FFFF to $40:C800-$40:FFFF (for levels and overworld map only)
$7F:C800-$7F:FFFF to $41:C800-$41:FFFF (for levels and overworld map only)

The setting only applies to ASM hacks that have not yet been inserted. Therefore it should be enabled before doing anything in the program that would result in an ASM hack being inserted.

The first time you open a SA-1 ROM in Lunar Magic, it will auto-detect if the ROM has had Vitor's RAM remap patch applied or not. Then the first time you save to a SA-1 ROM, it will gray out the option which will lock you into the current setting (this is to prevent accidently changing the option later and damaging your ROM).

This setting is saved to your current ROM.

Options Menu : Use Super FX RAM Remap for this ROM

This option only affects Super FX ROMs. It causes certain RAM addresses Lunar Magic uses in ASM hacks to be remapped to Super FX RAM, to be compatible with a Super FX RAM remap patch. The areas affected are :

$7E:0000-$7E:1FFF to $XX:6000-$XX:7FFF (shadow RAM)
$7E:C800-$7E:FFFF to $70:C800-$70:FFFF (for levels and overworld map only)
$7F:C800-$7F:FFFF to $71:C800-$71:FFFF (for levels and overworld map only)

The setting only applies to ASM hacks that have not yet been inserted. Therefore it should be enabled before doing anything in the program that would result in an ASM hack being inserted.

The first time you open a Super FX ROM in Lunar Magic, it will auto-detect if the ROM has had the RAM remap patch applied or not. Then the first time you save to a Super FX ROM, it will gray out the option which will lock you into the current setting (this is to prevent accidently changing the option later and damaging your ROM).

This setting is saved to your current ROM.

Options Menu : Compression Options for this ROM

This dialog allows you to change some compression options for the game. Normally the graphics in SMW are compressed using LC_LZ2, but you can choose to have the program replace the code with an optimized for speed version (using a patch by a few people on SMW Central), or use the LC_LZ3 format (using a patch by edit1754) which can get around 10% better compression.

Note that changing the options in this dialog may cause the program to automatically extract and reinsert graphics and other data to convert to the new format.

The settings are saved to your current ROM.

Options Menu : VRAM Patch Options for this ROM

This dialog allows you to change which VRAM patch type is installed. This patch rearranges VRAM so you can use an extra 2 GFX slots for more graphics in levels. If it is not installed, anything in graphics slots BG2 and BG3 will not be loaded in the game. This patch is also required for allowing horizontal levels to be resized vertically.

If Lunar Magic does not recognize the version of the patch currently installed, all options may be disabled.

The **None - Do not install patch** setting will not install the VRAM patch. This can make some features in the program unavailable or not function. This option is only available if the patch has not yet been installed.

The **Normal Version** setting installs the regular version of the VRAM patch. This is the default setting.

The setting is saved to your current ROM on your next level save.

Warning: This patch uses SNES RAM addresses that overlap with some older 3rd party ASM hacks that were made prior to Lunar Magic version 1.70, which makes them incompatible. If you have installed such modifications, either replace them with updated versions or do not install the VRAM patch.

Help Menu : Contents

Yeesh... If you can't figure out this menu on your own, you shouldn't even be trying to use a computer, let alone this program. :P

Help Menu : About Lunar Magic

Relm is not fooey. Don't believe anyone that tries to tell you otherwise. Those that ignore this message will incur the moon's wrath and be subjected to.... err... abnormally high tides and moldy green cheese.

Overworld Editor Introduction

The overworld in SMW consists primarily of two SNES layers plus sprites. Layer 2 is the bottom layer, and is the main layer on which all the land, water, and paths appear on. It uses individual 8x8 tile references to create the entire map. The back area is supposed to be completely obscured by layer 2, so do not leave any portions of the tiles for this layer transparent as the back area color is not constant and often changes without warning during gameplay. Layer 1 appears above layer 2, so most of it uses empty transparent tiles. It contains the actual level "icons" as well as a few clouds here and there. Layer 1 uses 16x16 tile references, much like SMW levels.

When you use a SNES emulator to test the overworld you've modified, *do not load a previous save state or use a save game (SRAM) that already has some levels passed!* This is because key portions of the overworld map data are decompressed/loaded from the ROM only once: after the player select menu at the start of a new game. And the directions to allow the player to leave each level from on the overworld are saved in SRAM. This means if you modify the paths between levels and the directions enabled from each, a save game from another version of SMW would be incompatible and result in the player not being able to walk on some of the modified paths.

Remember, a new game in SMW is started by selecting a game slot that displays the word "empty". Selecting a game slot that displays a "0" does NOT count as starting a new game!

Editing the overworld is a bit more complicated than editing levels, so it's highly recommended that you skim through the help file section for this editor before starting to use it.

Overworld Window Mouse Behavior

Although mouse behavior can alter depending on which editing mode you're in, the same general rule from level editing applies (a right click is used to paste and a left click is used to select). As usual, keys such as “control”, “shift”, or "alt" may offer additional functionality, so you may want to refer to the table below as a quick reference.

Left Click	If you click on an unselected tile/sprite, this will deselect any currently selected tile/sprite(s), and select the new tile/sprite. Depending on which mode you're in, the tile may also get selected into one of the tile selector windows.
Left Click on Nothing or Unselected Tile and Drag	Deselect any currently selected tile/sprite(s), create a rectangular selection area, and select all tiles/sprites within it.
Left Click on Selected	Move the selected tile/sprite(s) to a new

Tile/Sprite(s) and Drag	position. The editor will replace tiles at the old position with default tiles.
Control + Left Click on Tile Border (Layer 2 8x8 Editor Mode)	This will let you resize a selection of tiles to create a repeating pattern.
Right Click (Layer 2 8x8 Editor Mode)	If there are any 8x8 tiles currently selected in the Overworld Editor window, you will get a copy of those tiles. If no tiles are selected, this will paste a single 8x8 tile from the 8x8 Overworld Tile Selector window.
Control + Right Click (Layer 2 8x8 Editor Mode)	This will let you paste a single 8x8 tile from the 8x8 Overworld Tile Selector window regardless of whether any tiles are already selected in the Overworld Editor window.
Shift + Right Click (Layer 2 8x8 Editor Mode)	This will fill an enclosed area on the overworld with a pattern of 8x8 tiles from the currently selected area. If no tiles are selected, the pattern will be from the 8x8 Overworld Tile Selector window.

Control + Shift + Right Click (Layer 2 8x8 Editor Mode)	This will fill an enclosed area on the overworld with a pattern from the 8x8 Overworld Tile Selector window.
Left Click on Selected Tile(s) and Drag (Layer 2 Event Editor Mode)	Move the selected tile(s) to a new position. Note that because of the way the overworld is mapped out, some path tiles cannot be moved to certain positions near borders due to the way it will corrupt SNES RAM.
Right Click (Layer 2 Event Editor Mode)	If there are any tiles currently selected in the Overworld Editor window, you will get a copy of those tiles. The tiles will be inserted into the current step of the current event path you're at. If no tiles are selected, this will paste a single tile from the Layer 2 Event Tile Selector window.
Control + Right Click (Layer 2 Event Editor Mode)	This will let you paste a single tile from the Layer 2 Event Tile Selector window regardless of whether any tiles are already selected in the Overworld Editor window. The tile will be inserted into the current step of the current event path you're at.

Shift + Right Click (Layer 2 Event Editor Mode)	This acts just like a right click, except it will force the tile to be pasted as **silent step 0** of the current event path.
Control + Left Click on Tile Border (Layer 1 16x16 Editor Mode)	This will let you resize a selection of tiles to create a repeating pattern.
Right Click (Layer 1 16x16 Editor Mode)	If there are any 16x16 tiles currently selected in the Overworld Editor window, you will get a copy of those tiles. If no tiles are selected, this will paste a single 16x16 tile from the 16x16 Overworld Tile Selector window.
Control + Right Click (Layer 1 16x16 Editor Mode)	This will let you paste a single 16x16 tile from the 16x16 Overworld Tile Selector window regardless of whether any tiles are already selected in the Overworld Editor window.
Alt + Right Click (Layer 1 16x16 Editor Mode)	For level tiles, this will bring up the "Modify Level Tile Settings" dialog for that tile.
Alt + Left Click (Layer 1 16x16 Editor Mode)	Click on two Star/Pipe/Exit tiles to bring up the "Link Star/Pipe/Exit Tiles Together" dialog.

Right Click (Layer 1 Event Editor Mode)	If there are any 16x16 tiles currently selected in the Overworld Editor window, you will get a copy of those tiles. The tile will be inserted into the event you're at. If no tiles are selected, this will paste a single 16x16 tile from the 16x16 Overworld Tile Selector window.
Control + Right Click (Layer 1 Event Editor Mode)	This will let you paste a single 16x16 tile from the 16x16 Overworld Tile Selector window regardless of whether any tiles are already selected in the Overworld Editor window. The tile will be inserted into the event you're at.
Right Click (Sprite Editor Mode)	Paste a copy of currently selected custom sprite(s) at mouse position.
Alt + Right Click (Sprite Editor Mode)	Edit single custom sprite manually.
Shift + Mouse Scroll Wheel	Scroll horizontally.
Control + Mouse Scroll Wheel	Zoom in/out.

Overworld File Menu : Reload & Discard Unsaved Changes

When you first open up the overworld editor, it will automatically load the overworld from the ROM you currently have open in Lunar Magic. You can use this menu command to reload the overworld again without saving anything.

Overworld File Menu : Save Overworld to ROM

This will save the overworld map to the currently loaded ROM in Lunar Magic.

Since the overworld editor does not automatically reload the overworld when you open up a different ROM in the main Lunar Magic window, it's possible to open an overworld map in one ROM and then save it to another.

Overworld File Menu : Load Credits

This will load the credits that are displayed at the end of the game into the overworld editor for editing. The credits are nothing more than a tile map, so you can edit them the same way you would any other tile map.

In the palette editor, it will load the status bar colors from the shared palettes into colors 8-F of palettes 0 and 1. You probably shouldn't edit these, as the game already uses those in levels.

While the overworld editor is in this mode, most overworld dialogs and settings are not available. To leave this mode, simply reload the overworld.

Overworld File Menu : Save Credits

This will save the credits back into the ROM (assuming you're actually editing them).

Overworld File Menu : Load Title Screen

This will load the tile map for the title screen that is displayed on layer 3 at the start of the game into the overworld editor for editing. It will use the layer 3 GFX files of the title screen level (0xC7).

In the palette editor, the title screen colors will be loaded from the shared palettes into colors 8-F of palettes 0 and 1. These are only used for the title screen, so you can modify and save these.

If for some reason you also want access to the lower half of those palettes from your level (which are normally used in the background), open the level editor to the title screen level prior to loading the title screen. Those colors will then be copied from your level although you won't be able to modify them from here.

Note that only the upper 32x28 8x8 tiles are visible in the game. The rest of the area is not displayed in the game and is not saved by the editor.

While the overworld editor is in this mode, most overworld dialogs and settings are not available. To leave this mode, simply reload the overworld.

Overworld File Menu : Save Title Screen

This will save the title screen back into the ROM (assuming you're actually editing it).

Overworld File Menu : Insert Title Moves Playback ASM & Data

This will insert the joypad playback code ASM into the ROM, as well as the data you've previously recorded into a ZSNES or Snes9x save state, for playback during the intro level.

See the Install Title Moves Recording ASM topic for more details.

Overworld File Menu : Export Title Moves Playback Data

This will export your existing joypad playback data from the ROM for the intro level (that you've previously inserted) and save it as a fake ZSNES savestate file. While the file can't be used in an emulator, it can be used by Lunar Magic to save your recording to a different ROM. This can be useful if you lost the original savestate that held your recorded demo.

See the Install Title Moves Recording ASM topic for more details.

Overworld File Menu : Install Title Moves Recording ASM

At the beginning of the game on the title screen, you can see a demo of Mario playing a level. The demo uses level C7, and it's possible to both replace the level and record yourself playing it for the title screen demo.

To do this, save your level to some level number where you can get at it from the main overworld map (and save a copy over level C7). Use the “Install Title Moves Recording ASM” menu item to insert the recording ASM. Then open the ROM in ZSNES or Snes9x and enter the level you want to record yourself playing.

Play through the level as you normally would, but be careful to avoid certain things (see list below). If you find it difficult to play your level without making mistakes, you can create/use save states to get it right. Just don't use a save state that was created before the recording ASM was inserted.

Once you reach the point at which you want the demo to end, create a save state. Leave the emulator, go back into the overworld editor, and use the Insert Title Moves Playback ASM & Data menu to insert the data from your save state file into your ROM. Make sure you chose the right file, as sometimes the number in the file name does not match the slot number in the emulator (like with Snes9x, where the file number may be 1 less than the slot number). Check the timestamp on the file if you can't tell whether you have the right one.

Now, load up the ROM in an emulator and watch the entire demo to make sure it plays correctly. Sometimes you can end up with a bad recording where the sprites are slightly out of position and Mario dies. If that happens, you'll have to do it over.

If you're satisfied with the demo, make sure to go back into the overworld editor and use the Uninstall Title Moves Recording ASM menu item. Do not forget to do this!

If you make any changes to the level after the demo is recorded, or add/remove/modify custom block ASM, the timing will change and you will probably have to redo the demo. Because of this, it's usually best to hold off on recording a demo until just before you intend to release your hack.

Warning: Failure to uninstall the recording ASM will result in nasty consequences (ie. SNES RAM will continue to be overwritten starting with the overworld event tiles) during game play. This code is not intended to permanently reside in your ROM! Delete any save states created while the recording ASM is installed so you don't accidentally try to use them later!

Things to Avoid While Recording

- Do not kill Mario. If he dies during playback the demo will put the user back in control and will endlessly repeat the level until the user resets the game.
- Do not get an item that increases the time limit (green berry). If the level's time is set to 0/infinite and Yoshi eats a green berry, that gives Mario ~20 seconds before he dies.
- Do not leave the current level in any way. That means don't use a pipe, a door, pass the level, or do anything else that would take you out of that level.

- Do not cause Yoshi's greeting message to appear (the one that comes up when you encounter him for the first time on Yoshi's Island), because it won't appear during playback and will cause sync issues.
- Do not record your demo while special world has been passed. That changes the behavior and appearance of some sprites and makes them different from what you'll actually encounter in the title screen which will cause sync issues.

Overworld File Menu : Uninstall Title Moves Recording ASM

Use this menu item to uninstall the recording ASM that you've previously installed. See the Install Title Moves Recording ASM topic for more details.

Overworld File Menu : Load Layer 3 of Level

This will load the layer 3 tilemap of the level currently loaded in the level editor for editing. This can be either one of the game's original layer 3 tilemaps, or a custom tilemap you've loaded from an ExGFX file.

In the palette editor, only the colors that can be accessed for layer 3 are copied from the level. You won't be able to modify them from here.

While the overworld editor is in this mode, most overworld dialogs and settings are not available. To leave this mode, simply reload the overworld.

Overworld File Menu : Save Layer 3 of Level to ROM and ExGFX File

This will save the layer 3 tilemap of a level to an ExGFX file and insert the file into the ROM. Note that this does not cause the level to use the ExGFX file if it isn't already set to do so. You can do that from the level editor's Layer 3 GFX/Tilemap Bypass dialog. It also does not refresh the level to display the changes you've made, although you can do that with the Preview Layer 3 in Level command or by reloading the level.

If you are doing this from the file menu, or you have not yet set an ExGFX file when you hit the save button, a dialog will appear asking for some settings. Normally these are already filled in with the settings you had in your level when you loaded layer 3 into this editor. If you make any changes here, you should make the same changes to your level's settings. If you are not using 4bpp tiles in your ROM, you must use an ExGFX file number greater than or equal to 0xE00 to avoid 3bpp/4bpp tile conversions from being incorrectly applied to your tilemap file.

The destination setting is only used here to determine if you want to save starting from the bottom half of the tilemap. And the size setting determines the size of the file.

Overworld File Menu : Preview Layer 3 in Level

This will copy the layer 3 tilemap you're editing into the level that you currently have loaded in the level editor. Note that this is for viewing purposes only... it will not change the actual level.

Overworld File Menu : Load Layer 3 of Submap

This will load the layer 3 tilemap of the submap you select for editing. This can be either one of the game's original layer 3 tilemaps, or a custom tilemap you've loaded from an ExGFX file.

In the palette editor, only the colors that can be accessed for layer 3 are copied from the overworld. You won't be able to modify them from here.

While the overworld editor is in this mode, most overworld dialogs and settings are not available. To leave this mode, simply reload the overworld.

Overworld File Menu : Save Layer 3 of Submap to ROM and ExGFX File

This will save the layer 3 tilemap of a submap to an ExGFX file and insert the file into the ROM. Note that this does not cause the submap to use the ExGFX file if it isn't already set to do so. You can do that from the overworld editor's Layer 3 GFX/Tilemap Bypass dialog.

If you are doing this from the file menu, or you have not yet set an ExGFX file when you hit the save button, a dialog will appear asking for some settings. Normally these are already filled in with the settings you had in your submap when you loaded layer 3 into this editor. If you make any changes here, you should make the same changes to your submap's settings. If you are not using 4bpp tiles in your ROM, you must use an ExGFX file number greater than or equal to 0xE00 to avoid 3bpp/4bpp tile conversions from being incorrectly applied to your tilemap file.

The destination setting is only used here to determine if you want to save starting from the bottom half of the tilemap. And the size setting determines the size of the file.

Overworld File Menu : Exit

This will **hide** the overworld editor window instead of actually closing it. If you use the editor menu in Lunar Magic to open it again, it will appear exactly the same way as you left it.

Closing Lunar Magic will automatically close the overworld editor window.

Overworld Edit Menu : Layer 1 16x16 Editor Mode

This is the mode used to edit the layout of the levels on layer 1, as well as editing the invisible Mario paths that determine where Mario can move.

Chances are that you'll want the Future Layer 1 Tiles option in the view menu on and, when needed, the Layer 1 Mario Paths option. And you may want to keep the 16x16 Overworld Tile Selector window open at the same time.

Unlike the Layer 2 8x8 Editor Mode, the Layer 1 16x16 Editor Mode involves quite a bit more than just editing a tile map. Each level tile has certain settings involving everything from which level the tile leads to and which event to pass for the level, to which direction to enable plus when to "reveal" the level tile. And then some other tiles like the pipes, stars, and exit tiles have several tables of sprite and/or layer 1 positions to determine when and where the player should end up when entering one of these.

However, the editor has been coded to automatically track and update the multiple position tables involved with moving these tiles around. So for the most part you can just concentrate on building the paths Mario follows and sorting out the levels, events and directions without worrying about the nasty tables that go with it.

You'll probably find yourself using the Modify Level Tile Settings dialog fairly often. Just check the rest of the help file for more information on the other dialogs that are available in this mode.

Overworld Edit Menu : Layer 1 Event Editor Mode

In this mode, you can alter the order and placement of *some* of the tiles that appear on layer 1 when you pass an event. And of course you may want to keep the 16x16 Overworld Tile Selector window open at the same time while you're in this mode.

The reason I say "some" is because the majority of layer 1 tile updates are actually done by simply "revealing/morphing" an existing level tile in response to an event, which is set up in the Modify Level Tile Settings dialog in the other layer 1 editing mode. So if you're trying to "reveal" one of those semi-transparent level tiles after a player passes a level, 9 times out of 10 this would be the wrong mode to try doing it in. ^^

Unlike the two types of event steps in the Layer 2 Event editor mode, **this mode only has "Silent Event" steps!** This means that you should only use this mode to update submaps OTHER than the one the player is currently on, because VRAM is not automatically refreshed with the changes you make. Unless of course the tile you're placing is invisible anyway (like one of the Mario path tiles), in which case it doesn't matter. Although Nintendo didn't ever use this mode for dynamically changing the Mario path tiles in response to events, you can technically use it for doing that if you want.

Much like the Layer 2 event editor mode, you can use the **Page Up/Down keys** to go back and forth between events. But unlike the other mode, you cannot change the creation/Z order or step through each tile one by one using the Home/End keys, as there's no point in it. When you switch to each event, all the tiles that are going to be set by that event will be placed and highlighted in blue (assuming the view menu option for this is enabled).

Whenever you insert a 16x16 tile, it will be placed in the current event number that you're at. But remember that when you're placing these tiles, it does not in any way affect the level tile settings at that position. It's almost like being in the Layer 1 16x16 editor mode with the Allow move/copy/paste/delete on Layer 1 tile settings option turned off.

Since you can only set the level tile settings in the other layer 1 editing mode, you may find yourself in a situation where you have placed a layer 1 tile in this event mode but you can't modify the level tile settings at that position because there is no level tile there in the other mode. In that case, you can simply place one of the semi-transparent placeholder level tiles at the same location in the layer 1 16x16 editor mode so you can edit the level tile settings.

Overworld Edit Menu : Layer 2 8x8 Editor Mode

Since layer 2 contains the main map, you could say this is the primary editing mode of the overworld editor. This is the mode that you start out in when the overworld editor is first opened.

In this mode, you can edit the arrangement of 8x8 tiles on layer 2 for the main map, along with a few other areas that are only visible in this mode. You would normally keep the 8x8 Overworld Tile Selector window open at the same time so you can select new ones to insert.

You'll notice that the main map consists of one large map with 6 submaps underneath for Yoshi's Island, Vanilla Dome, etc. Each of these submaps has it's own colors that are inserted into the main overworld palette, which is why a tile in one submap may use different colors when dragged into another submap.

To the right of the main map is a large area filled mostly with "X" tiles which is used to store the Layer 2 event tiles. There are two event tile sizes; the first uses 6x6 8x8's (48x48 pixels) and is stored at the top, while the second uses 2x2 8x8's (16x16 pixels) and is stored in a separate group further down. Lunar Magic splits the entire area into two columns that's 48 8x8 tiles wide. Since it's unlikely you'd need so much space for the event tiles, you can use it as a type of backup storage area for sections of the main map if you like. Note that the path tiles you see here are just for showing the player where Mario can walk after an event, not for actually affecting where Mario can walk (that's done in the Layer 1 16x16 Editor Mode).

Below the submaps is another area filled with "X" tiles, along with all the 16x16 tiles used for layer 1. You can edit the individual 8x8 tiles here just as you would any of the others.

Overworld Edit Menu : Layer 2 Event Editor Mode

In this mode, you can alter the order and placement of the tiles that appear on layer 2 when the player passes an event. You'll usually want to keep the Layer 2 Event Tile Selector window open at the same time so you can select new tiles to insert.

You can use the **Page Up/Down keys** to advance through entire events, or the **Home/End keys** to "step through" each tile as it's placed on layer 2 during the current event. Depending on the View Menu options, previous event tiles in the current event will be highlighted blue and the next event tiles will be highlighted red. Selected tiles are inverted.

If you need to simulate situations where events have been passed out of order, use the Change Events Passed dialog.

In this event editor mode, there are actually two types of steps that are organized into separate groups. The **standard steps** are done first and will make a sound during gameplay for each tile that's placed during the event. The **silent steps** are done next, and are all placed simultaneously without any delay or sound and without updating VRAM. The main difference between the two is that standard steps are intended for modifying the current map that the player is on, while the silent steps are for modifying any other map the player isn't currently on. Do NOT use one to do the other, since standard steps used to modify other maps may cause strange things to appear, while silent steps used to update the current map will not show up until the screen is refreshed. Silent steps are thus much less commonly used than the standard ones. The status bar at the bottom of the screen will tell you which type of step you're on as you go through them.

Whenever you insert an event tile, it will be placed at the current step number in the current event and in the current event type group (standard or silent). If a particular event doesn't have any existing silent steps you can append to, you can use a **shift-right click** to force a tile to be pasted as the first silent event tile.

There are 0x78 events in total (listed from 0-0x77). Events 0x70-0x77 were not used or even fully implemented by the game, but LM expands a couple tables to make them usable.

Overworld Edit Menu : Sprite Editor Mode

In this mode, you can move around the sprites on the overworld. You can also set Mario/Luigi's starting position on the overworld in a new game by moving around the Mario or Luigi sprites.

Note that the sprite system used in SMW's overworld isn't as flexible as in levels... there are only a very limited number of sprite slots available, so right-clicking to copy is not available in this mode. If you wish to change/add/delete sprites, check the Edit Sprite List dialog.

You cannot move sprites from the main map to the submaps or vice versa (except for Mario/Luigi).

The editor moves sprites in 8 pixel increments during drag operations. If you wish to move sprites pixel by pixel, select the sprites, hold down the **shift key** and use the **arrow keys** to move them.

Although the editor will allow you to move Mario/Luigi's starting position pixel by pixel as well, the start positions must be set to align with the paths (because being even a single pixel off will mess up exit tiles). So the editor may automatically adjust the position when the overworld is saved.

If you move Mario/Luigi's start position, he will simply start at that location... the part where he walks up a bit like with the default start position will not occur.

For Custom Sprites: If you're using an external third party patch for custom sprites, you can add new sprites with the **insert key**. Unlike regular sprites, custom sprites can be moved between the main and submaps and right click to copy is available. However custom sprite positions can only be stored in 8 pixel increments, so don't bother moving them in single pixel increments. Don't place a custom sprite at Y=0x40, or at X=0-1 with Y>=0x40 as those positions cannot be represented in the table format being used and the sprite will be relocated on saving/loading. Up to 0x18 sprites per submap are supported.

Overworld Edit Menu : Undo

This will undo the last tile/sprite change. To change the max number of undo operations allowed, check the General Options dialog.

Overworld Edit Menu : Redo

This will redo the last tile/sprite change. To change the max number of redo operations allowed, check the General Options dialog.

Overworld Edit Menu : Cut

This is a clipboard function that will put the selected tiles onto the clipboard, then delete them from the overworld.

Overworld Edit Menu : Copy

This is a clipboard function that will copy the selected tiles onto the clipboard.

Overworld Edit Menu : Paste

This is a clipboard function that will paste the tiles on the clipboard into the overworld. Since the clipboard functions use the standard Windows Clipboard, it's possible to copy and paste tiles between multiple instances of Lunar Magic.

Overworld Edit Menu : Delete

This will delete all selected tiles/sprites from the overworld map.

Overworld Edit Menu : Insert

This will insert a tile from one of the tile selector windows in the editor menu, depending on which editing mode you're in. But since the right mouse button can also do this by directly clicking on the map, it's unlikely you'd want to do it using this menu command.

However if you're in sprite editing mode, this will let you manually insert a custom sprite (note: this requires an external third party patch using a table similar to Lui's system or nothing will show up in game). The **command** specifies which custom sprite ID you want. The **height** controls the z value, and the **extra** field lets you specify an extra byte which can control various settings depending on the sprite in question.

Overworld Edit Menu : Edit Manual

This will let you edit a single selected custom sprite manually. See the "Insert Manual" section for more information, as the fields here are pretty much the same.

Overworld Edit Menu : Increase Palette of Event Tiles

This will change which submap palette and graphics are used to display the Event Tiles to the right of the main map in the 8x8 editor mode. This is for display purposes only however...it won't actually change how they show up on the overworld itself.

Overworld Edit Menu : Decrease Palette of Event Tiles

This will change which submap palette and graphics are used to display the Event Tiles to the right of the main map in the 8x8 editor mode. This is for display purposes only however...it won't actually change how they show up on the overworld itself.

Overworld Edit Menu : Increase Event Tile Z Order

This command is used in the layer 2 event editor mode only. It will increase the step number of all the selected tiles by one, if possible. Since all event tiles with step numbers of a higher number are displayed above tiles with a lower step number, this also has the effect of increasing their creation/Z order.

Overworld Edit Menu : Decrease Event Tile Z Order

This command is used in the layer 2 event editor mode only. It will decrease the step number of all the selected tiles by one, if possible. Since all event tiles with step numbers of a higher number are displayed above tiles with a lower step number, this also has the effect of decreasing their creation/Z order.

Overworld Edit Menu : Next Event

This command will set the current event to the next event in an event editor mode. You would normally use the keyboard shortcut for this rather than the menu, though.

Overworld Edit Menu : Previous Event

This command will set the current event to the previous event in an event editor mode. You would normally use the keyboard shortcut for this rather than the menu, though.

Overworld Edit Menu : Next Event Tile

This command will set the current step of the current event to the next step in the event path editor mode. You would normally use the keyboard shortcut for this rather than the menu, though.

Overworld Edit Menu : Previous Event Tile

This command will set the current step of the current event to the previous step in the event path editor mode. You would normally use the keyboard shortcut for this rather than the menu, though.

Overworld View Menu : Show Level Numbers

Enabling this view option will display a small text label beside each level indicating the level number of the tile.

Overworld View Menu : Show Event Numbers

Enabling this view option will display a small text label beside each level indicating the base event number that will be activated by passing the level with a normal exit. But keep in mind that passing a level using a secret exit will add the secret exit number to the base event number to get the event number to activate.

Overworld View Menu : Special World Passed Palette

This command will change the palettes used by the submaps to the colors used when Special World has been passed. This is for display purposes only, as it won't change the actual appearance of the overworld in the game.

Overworld View Menu : Tile Grid

This creates an grid allowing you to more easily see how tiles are placed. If you want to change the grid color, hit control+alt+F8.

Overworld View Menu : Zoom

This will cause the overworld editor window to display the overworld zoomed at the specified percent.

Note that some zoom levels and the filtering option may not be available if DirectX is not being used.

Overworld View Menu : Show Star/Pipe/Location Numbers

Enabling this view option will display small text labels at the position of each Star/Pipe/Location teleport area. The top number is the index assigned to this location, and the bottom number is the destination index it leads to. If the destination is invalid or leads to a location that does not have its own index, the destination index will be shown as N/A.

If a number shows up in red, this is the destination of the index shown. The tile at this location does not have its own index assigned to it, which means that going here is a one-way trip.

Note that even if a location has an index assigned to it, if there isn't a pipe or star tile on layer 1 at that position, then the player will be unable to teleport from there.

Secondary exits in levels on the other hand do not rely on overworld Star/Pipe tiles being present, and can make use of any index. Teleport location indexes meant to be used by secondary exits in levels usually link to themselves.

Overworld View Menu : Show Exit Path Numbers

Enabling this view option will display small text labels at the location of each exit path teleport location. The top number is the index assigned to this location, and the bottom number is the destination index it leads to. If the destination is invalid or leads to a location that does not have its own index, the destination index will be shown as N/A.

If a number shows up in red, this is the destination of the index shown. The tile at this location does not have its own index assigned to it, which means that going here is a one-way trip. It could also mean that even though the location has an index assigned to it, the direction to enter from on this tile is mismatched with the other one, preventing an exact link match.

Note that even if a location has an index assigned to it, if there isn't a red Exit Path tile on layer 1 at that position, then the player will be unable to teleport from there.

Overworld View Menu : Show Koopa Teleport Numbers

Enabling this view option will display small text labels at each of the 3 teleport locations that the player will be moved to if they fail to pass a level that they were pulled into by one of the 3 corresponding Koopa Kid sprites.

See the Edit Sprites List section for more information about the Koopa Kid sprites.

Overworld View Menu : Layer 1

This option can turn the display of layer 1 (the layer that contains the levels and paths for Mario to follow) on or off.

Overworld View Menu : Layer 2

This option can turn the display of layer 2 (the layer that contains the main map) on or off.

Overworld View Menu : Sprites

This option can turn the display of sprites on or off.

Overworld View Menu : Sprite Data (Hex Code)

This will display a small "x" on top of every sprite, followed by a one-digit number indicating the subslot that the sprite belongs to (for example, the "Fish x3" sprite will create 3 fish, with subslots of 0, 1, and 2). Underneath that will be a two digit hex number with the main sprite list slot number, then another two digit hex number below it which represents the sprite command number itself. Generally you'd only use this for figuring out which slot or subslot a sprite on the screen belongs to.

For custom sprites, the subslot will show a "C". The slot number and command number for custom sprites can overlap with regular sprites without causing issues, as custom sprites are maintained separately from the original game's overworld sprites.

This option is off by default, and it will also not display anything regardless of the setting if the sprite viewing option is off.

Overworld View Menu : Animation

This will cause the overworld editor window to display the tile and palette animations that the game uses.

Overworld View Menu : Next Animated Frame

This will advance the overworld animation by one frame.

overworld View Menu : Reset Animations

Using this menu command will cause Lunar Magic to reset the animations of the overworld to their initial state for viewing.

While animations are already reset whenever you edit the animated frames or edit the animation settings, there are a few one-shot animation triggers that you may want to be able to reset the effects of at will for viewing purposes. This way you don't have to keep clicking through a dialog to do it.

Overworld View Menu : ExAnimation Triggers

This submenu allows you to toggle various triggers added through ExAnimation to preview their effects in Lunar Magic. This includes all the Custom, One Shot, and Manual triggers. There are a fair number, so the keyboard shortcuts have been set up so that you first have to select the trigger you want then use another key to activate it.

If you need to change an event's passed state to view an ExAnimation, you can do so by either advancing through the events in one of the event modes, or by setting their states in the Events Passed dialog.

Remember that a One Shot trigger is designed so that it will automatically turn itself off once the animation using it has finished going through all its frames. Also note that if an animation is set to be initialized to a certain state on level load, this state will be restored every time animations get reset for viewing in Lunar Magic.

Overworld View Menu : Future Layer 1 Tiles

This option can turn the display of layer 1 that will appear in the future. Tiles that may appear or will be modified in the future will be semi-transparent.

Note that the tiles shown will depend on which mode you're in. If you're in one of the Event Editor modes, the tiles that show up will be the **ACTUAL** layer 1 tiles that will appear in a future event. If you're in one of the 8x8 or 16x16 editor modes, the tiles that show up will be the **POSSIBLE** layer 1 tiles that may be "revealed" in a future event (this one doesn't include layer 1 events).

So for example, you may have noticed that when you're in one of the non-event editor modes, the yellow switch palace is green. That's because the level you pass to get to it is on a different submap, so Nintendo had to put a placeholder level tile there and use a silent event to change the tile rather than "revealing/morphing" the tile that's already there. (And if you're asking yourself why they didn't just place the correct yellow switch palace placeholder tile there instead of the green one, most likely Nintendo thought no one would ever see it anyway. But that's just IMO. ^^)

And as another example, the bridge exit tiles that you see at places like the entrance and exit to the Forest of Illusion are technically there but were never set to be "revealed". This is because both the original and the morphed tile can be used as an exit tile and Nintendo apparently didn't want a bridge there.

There's also an extra red switch palace level tile that appears in Bowser's valley, which may have been something Nintendo was planning to add (or perhaps use as an alternate location for the red switch palace?). It isn't set to be revealed by any event, so it only shows up in the non-event editor modes. (There also no path to it for Mario to follow and the level number is an empty test level... but the level beside it is set to enable the left direction leading to it on using a secret exit, which isn't the default setting. Interesting...)

If you want a more accurate view of how the overworld looks as you step through events, you may wish to turn this option off and only use it for when you're editing.

Overworld View Menu : Static Layer 2 Tiles

This option can turn the display of static (ie. non-event path tiles) on or off. This will have no effect in anything other than one of the Event Editor modes.

Overworld View Menu : Layer 1 Mario Paths

This option can turn the display of the paths Mario can follow on or off.

The paths that Mario can follow is determined by using certain blank layer 1 tiles to define the path. But since trying to edit a path you can't see would be pretty difficult, you basically need to turn this on to see what the path would look like.

Each path tile also indicates the action Mario will perform while on the tile. Green paths are just standard ground walking tiles. If it has black lines across it, Mario will climb as though he was on a ladder. Blue tiles indicate that Mario will swim through that area, and red tiles are used to mark "exit tiles". A black X on the tile indicates Mario can stop there, but cannot enter the level. There are also 3 tiles with tiny fish symbols on them... if Mario walks across one of these, one of the fish sprites nearby will momentarily jump out of the water.

To create a path between levels, you can simply select and drag the layer 1 tiles as you would any other tile. Make sure that the path doesn't lead to a dead end and doesn't have fragments missing though, since Mario's movements become erratic and random if that happens, which could trap the player in an undesirable spot. To use a path between normal levels in the game, you have to pass the level to enable the direction of the path from that level (the direction can be set up in the level tile settings dialog for that tile).

You'll probably notice that there are 5 paths in SMW that are not displayed in the editor. These are the hardcoded "default paths" that Mario will follow for certain levels or positions if there is no path already set up on layer 1. They appear to mainly have been created so that you can cross over regular layer 1 paths at a few places, since standard layer 1 paths cannot cross over one another. Currently LM does not support displaying or editing these. You can still use them as they are if you want, or you can just ignore them. You can also turn them off in Extra Options.

Overworld View Menu : Make All Mario Paths Translucent

Normally only future layer 1 path tiles will appear translucent in the editor. However, you can enable this option to make all the paths appear translucent, which may make editing a little bit easier for some things.

Overworld View Menu : Old Path Tiles C and 13

This option can change the layer 1 path tiles displayed for tiles 0xC and 0x13 to how they appeared in Lunar Magic prior to version 2.40.

In old versions of Lunar Magic these 2 path tiles appeared the same as tile 0x10. While all 3 tiles do in fact act the same way, the 8x8 tile numbers Nintendo used suggests that the other 2 tiles were intended as a visual guide for paths that are offset to the left or right. So newer versions of Lunar Magic display them this way as well.

Overworld View Menu : Highlight Next Event Tiles (Red)

This option will control whether or not to highlight the area of the next tiles of the current event red. This can make it easier to see how the event progresses while stepping through it.

Overworld View Menu : Highlight Previous Event Tiles (Blue)

This option will control whether or not to highlight the area of the next tiles of the current event red. This can make it easier to see how the event progresses while stepping through it.

Overworld Editors Menu : Overworld Palette Editor

This opens up a dialog where you can edit the colors of the overworld. It functions mostly the same way that the palette editor in Lunar Magic does, so read the help section for that if you want the basics.

The main difference from the palette editor in LM is the drop down box that allows you to select which submap you want to edit the colors for.

Note that SMW uses a "path fading" effect that consumes around 25% of the color table. When a path is to be revealed, the game will copy colors 1-7 of palettes 4-7 into palettes 0-3 and C-F then dynamically modify them to make tiles appear to fade in/out. If you'd rather turn this effect off so you can use these colors normally, see the Extra Options dialog.

Overworld Editors Menu : 8x8 Overworld Tile Selector/Tools

This command will open up a window that allows you to select 8x8 tiles to paste into the main overworld window when it's in the Layer 2 8x8 Editor Mode. It also contains controls to edit the tiles already in the map.

In the original game, the first 0x100 tiles (FG1 and FG2) were mostly used for layer 2, and the next 0x100 tiles (FG3 and FG4) were often used for layer 1. This is mainly because of the "path fade effect", which requires tiles on layer 2 to be within FG1 and FG2 if they're involved with event paths. However you can turn this effect off in the Extra Options dialog if you don't want to put up with this restriction.

When placing tiles on layer 2, keep in mind that they should normally not use the transparent color, as the back area color that will show up behind them does not remain the static blue you see in the editor. But layer 1 tiles can use it, as layer 2 is shown behind it.

Any single tile you select in the main overworld editor window will be selected in this editor. To paste tiles from this editor back into the overworld, just use a right click in the overworld window. No other tiles in the overworld editor window must be selected when you do this though, or you'll end up with a copy of those tiles instead (you can hold down the control key to avoid that). Or you can use the standard windows clipboard keyboard shortcuts (which also places the tiles on the windows clipboard as text). To make things easier, when you select tiles in this window it will cause any tiles in the main editor window to be automatically deselected unless the auto-deselect option for it is turned off.

The row of buttons in this window starts with a **Display GFX Slot Numbers** button, which will display outlines showing the GFX files loaded in each slot. The **grid** button displays an 8x8 tile grid. And the **gradient** button beside it can toggle either viewing a gradient as the background in the window or black. The next button is to toggle using the **"Special World Passed"** palette to display the tiles. And the last button is the **zoom** button.

The 2 drop down boxes under that affect the tile selector window only. The first one is to select the **submap** graphics and palette to use in the tile selector, while the second one is to select the **palette number** to display all tiles in.

The next group of controls are for editing in the main overworld editor window. When selecting tiles in the main editor, these will change automatically to reflect the settings of all the tiles. If the settings for some of the tiles are not the same, the control will indicate this by changing to "--".

The first 2 buttons are for **flip x** and **flip y**. Next is a drop down box for the **palettes**. While there are technically 16 palettes in the SNES palette table, FG/BG tiles can only use palettes 0-7. Sprites use the remaining palettes of 8-F.

The **priority** drop down box affects tile layering. In general this can be used to make a tile appear above sprites or another layer. The actual effect will depend on the layer setting of the particular sprite, or whether the tile layers are both on the SNES main/sub screen for the overworld. For more information on how SNES layering works, you may want to consult an SNES hardware document.

Next are 4 text edit controls that contain the 8x8 GFX tile numbers of the first 4 tiles you've selected in a square formation. Since you can paste tiles simply with right clicking you'll probably only use this as a reference, but you can edit the tile numbers directly here if you want. Just keep in mind that although it will let you enter values as high as 0x3FF since this is the max that the SNES tile format allows for, only up to 0x1FF is normally available (or 0x2FF when the merge GFX slot option is on).

Near the bottom you'll find the **Remap** button. This brings up a dialog that allows you to remap selected 8x8 tiles to other tiles or palettes. In layer 3 editor modes, all tiles are remapped by default (but you can hit F9 in this window to do selected tiles only). The most common use would be to add an offset to all tile numbers, useful in cases where you may have imported a GFX file to a different slot than it was originally made for. The first field in this dialog lets you enter either a value to do exactly this.

For fine tuning or much greater control over which tiles are changed, you can instead enter a list of paired comma separated values in the large text edit in this dialog. The first value can be a single value or a range indicating the source of what you want to change, while the second value describes what to change it into. Each pair of values can be on their own line, or you can just separate them with more commas. There are several examples given in the dialog for the syntax you can use. The simplest is changing one tile into another: 100,25 would change tile 100 into tile 25. Or you can do a range:100-101,25 would change both tiles 100 and 101 into tile 25.

To add or subtract an offset, you must specify the sign to use. 100-101,+25 would change 100 to 125 and 101 to 126. Or 100-101,-25 would change 100 to DB and 101 to DC. But since the intent of adding/subtracting offsets is usually to move a range of tiles to another area, it may be easier to just use 'M' and the starting destination: 100-101,M125 would also change 100 to 125 and 101 to 126. It's also common to move a rectangle of tiles, so you can specify the top left and lower right points of a rectangle using a command like R100-111,M25 to change tiles 100 to 25, 101 to 26, 110 to 35 and 111 to 36.

You can also use this list to change which palettes are used for each 8x8 tile. P5-6,7 would cause all the tiles that use palettes 5 or 6 to use palette 7 instead. You can even change the palettes used by specific 8x8 tile numbers: 100-101,P7 would cause tiles 100 and 101 to use palette 7.

When entering a list of remap pairs, it's important to remember that the source values always refer to the tiles as they currently exist *before* any remapping occurs, not as they would exist after processing the pairs that come before it. For example, you might think the sequence 100,25,25,100 would be of little use, but it will really cause tiles 100 and 25 to be swapped. This also means in a sequence like 100,25,100,30 the first pair is effectively useless as it's superseded by the second pair.

Overworld Editors Menu : Layer 1 16x16 Tile Selector

This menu command will open up a window that allows you to select 16x16 tiles to paste into the main overworld window when it's in either of the layer 1 editor modes.

Most of the tiles will appear blank, unless you have the Layer 1 Mario paths and the Future layer 1 tiles options enabled in the view menu of the overworld editor.

Any single 16x16 tile you select in the main overworld editor window will be selected in this editor. To paste a tile from this editor back into the overworld, just use a right click in the overworld window. No other tiles in the overworld editor window must be selected when you do this though, or you'll end up with a copy of those tiles instead (or you can just hold down the control key to avoid that).

Overworld Editors Menu : Layer 2 Event Tile Selector

This command will open up a window that allows you to select event path tiles to paste into the main overworld window when it's in the Event Path Editor Mode.

You can use the **Up/Down Arrow keys** to scroll to different pages.

Note that the first few pages are for 6x6 8x8 tiles, while the rest are for 2x2 8x8 tiles.

To paste a tile from this editor back into the overworld, just use a right click in the overworld window. No other tiles in the overworld editor window must be selected when you do this though, or you'll end up with a copy of those tiles instead.

Overworld Overworld Menu : Submap GFX

Opening this dialog will cause Lunar Magic to immediately install an ASM hack into the ROM, allowing you to bypass the standard GFX files loaded by the overworld.

The list box at the top of the dialog will select which submap to change the GFX for, and will also list the first 4 FG files.

In this dialog are several drop down boxes that allow you to select which GFX files to load for the FG, the sprites (SP), or the animated tile source frames. Values 0-31 are for accessing the original GFX files of the ROM. Values 32-7E are invalid...DO NOT USE THEM! Value 7F should only be used to skip decompressing a file for a slot you aren't using. Values 80-FFF are for any ExGFX you've inserted. Each index value has a number in parenthesis that indicates the current ROM address of the GFX file in question, assuming it's inserted.

The GFX files contain 0x80 4bpp tiles, which means they should be 0x1000 bytes (4KB). The AN2 file is an exception... it can have up to 0xD0 4bpp tiles and be 0x1A00 bytes. Even if you aren't using the 4bpp option the requirements for these files are the same, as the program will internally auto-convert the files for 3bpp on insertion if needed.

You will not be able to change the AN2 file until you insert the GFX as 4bpp (you may need to reload the overworld afterwards).

The "**Edit**" buttons beside each slot will launch an external tile editor you can use on the graphics files once they have been extracted from the ROM. If the file doesn't yet exist in the ROM or extracted folders it may offer to create a new blank one for you. Just remember to reinsert any edited files to the ROM if you want your changes to show up. You can configure the tile editor to use from the "Setup Tile Editor" menu.

Note that if you have not disabled the "path fade effect" in the Extra Options dialog, you should set SP3 and SP4 to the same files as FG1 and FG2. The path fade effect also imposes other restrictions on tiles and colors used... check the option about disabling it for further info.

FG5 and FG6 will be disabled and unavailable unless you turn on the option to merge FG1 and FG2 into SP3 and SP4 in the Extra Options dialog. If you do so, SP3 and SP4 will be disabled instead but their contents will be exactly the same as FG1 and FG2, which means you could still use those tiles in a sprite if you want.

Overworld Overworld Menu : Submap Layer 3 GFX/Tilemap Bypass

In the original game, layer 3 on the overworld was used for the border around the screen and for displaying text like the level names and some menus.

Opening this dialog will cause Lunar Magic to immediately install an ASM hack into the ROM, allowing you to bypass the standard GFX files used by the overworld.

The list box at the top of the dialog will select which submap to change the GFX for, and will also list the 4 GFX files currently used by it. The checkbox under it will determine whether or not this submap will bypass the standard GFX and use the custom settings.

As with the "Submap GFX", in this dialog are several drop down boxes that allow you to select which GFX files to load for layer 3. In the original game, the layer 3 GFX were only decompressed once at the start of the game and mostly left alone. But once you save at least one level or submap in Lunar Magic that bypasses the layer 3 GFX, the program will enable bypassing layer 3 GFX on a per level and per submap basis. If the bypass isn't enabled for a particular area, it will simply use the original files (0x28-0x2B).

Layer 3 GFX files contain 0x80 2bpp tiles, which means they should be half the size of regular 4bpp GFX files. So the files should be 0x800 bytes (2KB) instead of 0x1000 bytes (4KB). Remember that 2bpp tiles can only use 4 colors (1 transparent + 3 regular colors). These colors come from the main palette table, however the palettes are assigned by using the first 4 colors for the first palette, the next 4 colors for the second palette, and so on. So only the first 16 colors in the main palette table can be accessed (the top 2 rows in the palette editor window), and only 3 out of every 4 of those colors can actually be shown.

The "**Edit**" buttons beside each slot will launch an external tile editor you can use on the graphics files once they have been extracted from the ROM. If the file doesn't yet exist in the ROM or extracted folders it may offer to create a new blank one for you. Just remember to reinsert any edited files to the ROM if you want your changes to show up. You can configure the tile editor to use from the "Setup Tile Editor" menu.

This dialog also lets you **bypass the layer 3 tilemap** by choosing a GFX file to load that actually contains tilemap data. The **destination** for the file should be "Start of Layer 3" as layer 3 was mostly used for the border around the screen. The **file size** of the tilemap file should be 0x800 bytes (2 KB) for 256x256, which is enough to cover the full screen. You could do the full tilemap, but the rest wouldn't normally be visible so there probably isn't much point.

To edit the layer 3 tilemap file, see the Load Layer 3 of Submap command in the overworld file menu.

Note that using an exit tile during the game will not reload the layer 3 tilemap, but the layer 3 graphics will reload. You should probably consider using a pipe or star tile anyway if you're planning to have a completely different border than another submap, as a fade-to-black transition is less visually jarring than abruptly changing the border while it's still being displayed.

Overworld Overworld Menu : Edit Submap ExAnimated Frames

This dialog is pretty much the same as the Edit Level ExAnimated Frames one, so you should read that section for more information.

There are however some differences for the overworld:

- **Submap** - There is a submap selection field. Each submap can have it's own set of animations, just like individual levels.
- **Trigger Event Manual 8-F** - these have replaced some level specific triggers. The new triggers activate on specified overworld event numbers. The event to activate on is taken from the Manual Frame Triggers of the same number. For example, Event Manual 8 will use the RAM value from Manual 8 as the event number to check for. You can set the event numbers using the "Trigger Init" button. Up to 8 events per submap can be checked.
- **Event Path Fade Effect and Palette ExAnimation** - The fade effect creates its own palette animation. Any color involved with this effect should not be used in palette ExAnimation. Please see the Extra Options dialog for a complete list of restrictions this effect imposes, as well as an option for turning it off.
- **Stop on Fade** - As the overworld fades work differently from levels, there was no need to have palette

ExAnimation stop on fades. The "Stop on Fade" trigger entries have simply been marked with [Do Not Use] in case they get used for something else in the future (for now they're just duplicates of the entries above them).

- **Palette Back Area Color** - This should not be used. While technically it works, the overworld already uses the back area color for menus and the exit tile transition effect. Therefore the back area color shouldn't be used as part of your maps.
- **Advanced Animation Tricks** - While in levels it was possible to overwrite the original game's animated tiles depending on the ExAnimated slot used, this isn't possible in the overworld unless you're doing a tile at 60fps. This is because there weren't many animated tiles on the overworld to begin with, so Nintendo simply transferred all of them every single frame to VRAM rather than staggering them.
- **Disabled GFX Slots** - Do not set tile destinations to be within disabled GFX slots. This will not work in the game and will cause undesirable side effects during gameplay.
- **$7FC004** - this RAM address, which is maintained for partial backwards compatibility with LM version 1.6x blocks in levels, is not available for overworld ExAnimations.

Source Frames

Tile Range	Direct Offset	Type	Description
0x600-0x6D7	0x1AD00-0x1C7FF	RAM	Extended animated tile area AN2.

0x700-0xAFF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x60.
0xB00-0xEFF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x61.
0xF00-0x12FF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x62.
0x1300-0x16FF*	0x10000-0x17FFF	ROM	Non-compressed ExGFX file 0x63.

*Requires animation slot to be set to use alternate ExGFX source, and for Alt ExGFX file to be set to appropriate file.
Note: If using direct offsets, remember that a single 8x8 4bpp tile takes up 0x20 bytes and a single SNES color takes up 2 bytes.

Destinations

Tile Range	**Direct Offset**	**Type**	**Description**
0-0x3FF	0x10000-0x12FFF	VRAM	FG graphics FG1, FG2, FG3, FG4, FG5, FG6, FG7, FG8 (in order).
0x400-0x5FF	0x16000-0x17FFF	VRAM	SP graphics SP1, SP2, SP3, SP4 (in order).
0xA000-0xA1FF	0x14000-0x14FFF	VRAM	Layer 3 graphics (2bpp tiles) LG1,

			LG2, LG3, LG4 (in order).

Note: If using direct VRAM offsets, remember that the SNES uses word (16 bit) addressing for VRAM. So if you're working with byte offsets, divide by 2 before adding 0x10000 for the direct offset to enter. This means single 8x8 4bpp tiles would really be 0x10 away from each other in VRAM addressing, not 0x20.
Note 2: If you have enabled the option to merge FG1/2 with SP3/4, the VRAM offsets for FG1/2 will be the ones used for SP3/4, while FG3+ will have the VRAM offsets of what was previously FG1+.

RAM Locations

RAM Address	**Size**	**Description**
$7FC070	0x10 bytes	Manual Frames 0-F , 1 byte each.
$7FC080	0x20 bytes	Frame Counters for submap slots 0-1F, 1 byte each.
$7FC0A0	0x20 bytes	Frame Counters for global slots 0-1F, 1 byte each.
$7FC0F8	0x4 bytes	One Shot Triggers 0-1F, 1 bit each.
$7FC0FC	0x2 bytes	Custom Triggers 0-F, 1 bit each.

Overworld Overworld Menu : Edit Global ExAnimated Frames

This dialog is pretty much the same as the Edit Submap ExAnimated Frames one, so you should read that section for more information.

Just remember that animations added here will run in all submaps, potentially consuming processing and vblank time. So think carefully about what you decide to put here.

Overworld Overworld Menu : Edit Animation Settings

Since there may be submaps where you will want to save on VBlank time, processing time, or tile/palette space, this dialog will allow you to set whether or not certain animations are enabled in a particular submap.

You can disable the game's original animated tiles, the original level dot palette animation (for colors 6D and 7D), the original lightning palette animation (for color 47), Lunar Magic's global animations, or Lunar Magic's submap animations. Note that even if LM's animations are disabled, any trigger initializations you've set up will still activate on submap load.

For some reason the source colors for the lightning palette animation are taken directly from the current submap's working palette table, namely palette 2 colors 9-F. You can cause these colors to be read from the ROM instead in the Extra Options dialog.

Note: For ExAnimation to work as it should with exit tiles, you must first insert the graphics at least once as 4bpp then save the overworld.

Overworld Overworld Menu : Change Max Event 6x6 Tile Area

As you know, the event path tiles are split into two groups. The first collection of tiles are for 6x6 8x8 tiles, and the rest are for 2x2 8x8 event tiles. This command brings up a dialog that can alter how much of the event tile space is allocated for the 6x6's (and thus the 2x2's). If you change this setting, the 2x2's will be automatically moved to the end of the 6x6 area, and the overworld will be rebuilt to adjust for the difference.

Do not use the higher settings, as you could wipe out the 2x2 area entirely!

Overworld Overworld Menu : Change Events Passed

In this dialog, you can change which event number you're currently on and which events have already been passed in the overworld editor window. But remember, this is for testing purposes only. It doesn't change anything in the ROM.

Overworld Overworld Menu : Modify Level Tile Settings

This dialog can only be used if you select a single layer 1 level tile (or you can just point the mouse cursor at one and use an alt-right click on it).

Standard Settings

The **level number to use for this tile** setting determines which level number the player will be taken to when they enter the level. Note that not all the levels can be directly entered into from the overworld. Only the ones listed in this dialog can be used for that.

The **base event for when the above level is passed** setting is used in determining which event number to activate when the person passes the level. If the normal exit is used, the base event number is activated. If the player uses a secret exit, the game will add the secret exit number to the base event number to get the event number to activate (which means secret exit 1 activates base event number+1, etc). So make sure to plan ahead when allocating events to levels, since you don't want a secret exit to activate an event that you're already using for another level.

Each event number can normally only be activated once. If a particular event has already run before, then nothing will happen. And if the base event is set to "No event", nothing will happen for any exit in the level other than being taken back to the overworld.

You can only set one base event per level number. If you placed another level tile on the overworld with the same level number but gave it a different base event number, it would change the base event number for the first level tile too. Since the game also uses the level number to keep track of enabled directions for level tiles, it's generally recommended that you don't set more than one tile on the overworld to use a particular level number.

The **direction to enable when normal exit is used** setting will enable one of four directions from this level tile that Mario will walk in when the level is passed and an event is run. If no event is run, no new direction will be enabled. If there is no path on layer 1 in the direction that is enabled, and none of the 5 hardcoded default paths connect to this position or use this level number, Mario will do nothing. In cases where there is a standard path right where a default path also begins, the game will use the standard path.

You don't have to worry about enabling the direction from a level tile BACK to whatever level tile that the player walked there from. The game automatically enables the direction for the return path when Mario walks onto a level tile from another tile.

You may have noticed that there are 2 pipe tiles available. Pipe tile 0x82 is special as it doesn't care about allowed directions and lets the player walk from it to any standard path that connects to it, excluding the default paths. Nintendo likely used this tile for all their pipes in the original overworld just so they could assign all the pipe tiles to level 0 (as allowed directions are tracked by level number). Pipe tile 0x5B on the other hand doesn't have this special allow-all-directions status. This tile would be useful for cases where you have more than one path to a pipe and may not want directions to both paths enabled at the same time, but that situation never came up in the original game.

Star Road tiles cannot be "passed" as levels, but you must still assign a different level number to each one so the game can track the allowed directions for the tile. Unfortunately this wastes the level number for use as an actual level on the overworld. (Interestingly, since *unrevealed* star tiles can be passed as levels instead of used as warps, you could for example change the unrevealed star tile's appearance to a red star and use it as a level that the player has to pass before it gets revealed as a yellow warp star. If you don't intend to do something like that though, you needn't bother assigning the Star Road an event number like Nintendo did on occasion.)

The **direction to enable when secret exit 1 is used** setting is the same as the above setting, except it's only enabled by passing the level using secret exit 1. The same goes for the direction to enable settings for the other secret exits.

The **reveal this level tile on any of these events** setting will cause a placeholder level tile to be "revealed" when someone passes a particular event. While you can only reveal a single tile per event, you can reveal the same tile from 2 or more different events. You can use a comma separated list for more than one event. Revealing a level tile that has already been revealed (or has no tile to reveal itself into) will have no effect. Most unrevealed tiles are invisible, but others like the door aren't. You can change the appearance of any of the unrevealed tiles if you want. Also, you should not attempt to "reveal" levels that exist on submaps other than the one that the player is currently on. Use the Layer 1 Event Editor Mode for doing changes to other submaps.

Initial Setting Flags

In this section are 8 flags that are loaded for each level number at the beginning of a new game. Just to clarify, a new game in SMW is started by selecting a game slot that displays the word "empty". Selecting a game slot that displays a "0" does NOT count as starting a new game!

The **Enable Up, Down, Left, Right** flags control the initial states of the allowed directions that Mario can walk from a level tile. Most of the time these are enabled through events or by walking onto the tile (which automatically enables the return direction), but in some cases (star roads, Yoshi's house) you may need to manually enable certain directions right from the start.

The **Level has been passed** flag is set by the game whenever the player passes a level (using a standard or secret exit). It's only used to determine if the player can use start-select to exit the level without passing it again, or to block entry for the "No entry if level is passed" setting. This flag's state has no effect on the events whatsoever... setting it will not pass any event.

The **Midway point obtained** flag is set by the game if the player got to the midway point but left the level without passing it. The flag is cleared when the player passes the level. If the flag is set when a player enters the level, he'll start at the level's midway point entrance.

The **No entry if level is passed** flag was added via an ASM hack. If this flag is enabled, the game will check the "Level has been passed" flag when a player tries to enter the level. If the level has been passed, the player cannot get back into the level.

The **Save prompt** flag was added via an ASM hack. If this flag is enabled, the game will bring up a save game prompt when a player passes the level. The game already does this automatically for certain layer 1 tiles (castles, ghost houses, etc), but you can disable that in the Extra Options menu and use this flag to do all saving on a level-by-level basis.

Overworld Overworld Menu : Destroy Level Tile Settings

No, this dialog isn't for destroying settings. ^^ It's used to control the settings of the layer 1 level tiles that get destroyed, such as the castle, fortress, and switch palace level tiles. This dialog can only be used if you select one of those tiles first.

The **event number to destroy this level tile on** setting is a list of 0x18 events that you can use to store which event to destroy a particular level tile on. You cannot destroy more tiles than there are entries in this list. The ones from 0x10-0x17 don't appear to have been used in the original game, so you can use them for collapsing additional levels.

There's a text field below this list where you can **change the above event number** to some other value that you want.

Only one tile can be destroyed per event, although you can "destroy" one tile and "reveal" some other tile using the same event since those actions are performed separately. Do not put in more than one entry for a particular event number, as the game will use the one closest to the bottom of the list and ignore the extra ones.

Overworld Overworld Menu : Set Koopa Teleport Here

This dialog can only be used if you select a tile on layer 1 in the main map.

When the player fails to pass a level that they were drawn into by a Koopa Kid, they are teleported to one of 3 locations on the main map. Each location must be assigned it's own index number so that the game can keep track of which kid corresponds to which teleport location. The overworld editor can track these tiles and update their coordinates in the table when you move them.

To view the teleport locations, use the Show Koopa Teleport Numbers command in the view menu.

See the Edit Sprites List section for more information about the Koopa Kid sprites.

Overworld Overworld Menu : Set Location Teleport Here

This dialog can only be used if you select a tile on layer 1.

Lunar Magic allows secondary exits in levels to exit to the overworld. This dialog lets you specify which index to use for a specific overworld location. Note that it uses the same table as the Star/Pipe tiles, but unlike those you only need to specify a single index (which is then linked to itself).

Secondary exits can also make use of indexes that are assigned to overworld stars and pipes, but just remember to use the index assigned to the tile that teleports to the location you want to reach, not the one assigned to the destination.

Ignore the "Create a one-way link" entry in the list, as it does nothing in this dialog.

To view the teleport locations in the overworld editor, use the Show Star/Pipe/Location Numbers command in the view menu.

See the Modify Secondary Entrances section for more information about how these teleport locations are used.

Overworld Overworld Menu : Exit Path Tile Settings

Notice: This dialog is considered obsolete. While you can link exit tiles together in this dialog using an older procedure listed below, the preferred method is to use the Link Star/Pipe/Exit Tiles Together dialog instead.

This dialog can only be used if you select one of the layer 1 exit tiles first. However these tiles are often invisible, so you should enable the Layer 1 Mario Paths setting in the view menu. The exit tiles are the path tiles that are in red.

An exit tile is normally used by the game to allow the player to walk from one map of the overworld to another. Each tile must be assigned it's own index number in the exit list so that the game can keep track of the coordinates to use for transporting Mario to another location. The overworld editor can track these tiles and update their coordinates in the table when you move them.

Potential Exit Tile Issues

- If Mario's coordinates are off by even a single pixel when he hits an exit tile, it will malfunction and Mario will end up right back at the same location he was trying to leave. If relinking the exit tiles doesn't help and you've set the correct side to enter from, it means you haven't created proper layer 1 paths between your levels and Mario's position is a little bit off from where he should be normally. Examine your paths and either correct or simplify them.
- If you have not yet inserted the GFX as 4bpp then saved the overworld, the game will not reload any graphics when you use an exit tile. This wasn't an issue in the

original game since the main map and submaps all used the same graphics files. But if you've chosen different graphics files to load for submaps and you're still using 3bpp tiles, it's better to use a pipe or star road instead to force the correct graphics to load.
- The game sets Mario's new screen position after touching the exit tile, but *before* fading the screen out. This is noticeable if you happen to use an exit tile that goes to the same screen that you're currently in. What you may not expect is that it can also show up when going between the main map and submaps! This is because the submaps are all stored together on a single map that is the same size as the main map, and if the coordinates are close enough you'll get the same effect. Keep this in mind when arranging the location of exit tiles on the main map compared to your submaps so you can avoid this minor graphical glitch.
- If you place exit tiles right beside one another you may have to relink them if you later move them apart again, as the editor may not be able to tell which index belongs to which tile.

Relinking Exit Tiles (old method)

Carefully follow this procedure to re-link any two exit tiles: (Just make sure to sort out which two exit indexes you're going to use ahead of time.)

- Select the first exit tile, and enter this dialog. For the **Exit index to use,** select the first index you're going to use. Set the **Destination exit index** to "No Setting". Change the **Side to enter from** setting to the correct side. Hit ok.
- Select the second exit tile, and enter this dialog. For the **Exit index to use,** select the second index you're going to use (must be different than the one you used

for the first exit tile). Set the **Destination exit index** to the first index you're using (the same one you set for the first exit tile). Change the **Side to enter from** setting to the *opposite* side you used for the first exit tile. Hit ok.

- Select the first exit tile again, and enter this dialog. You can now set the **Destination exit index** to the second index you're using. Hit ok.

And that's it. The two exit tiles should now be re-linked to one another.

Overworld Overworld Menu : Star and Pipe Tile Settings

Notice: This dialog is considered obsolete. While you can link star/pipe tiles together in this dialog using an older procedure listed below, the preferred method is to use the Link Star/Pipe/Exit Tiles Together dialog instead.

This dialog can only be used if you select one of the layer 1 star or pipe tiles first.

The star and pipe tiles are normally used by the game to allow the player to warp from one map of the overworld to another. Each tile must be assigned it's own index number in the star/pipe list so that the game can keep track of the coordinates to use for transporting Mario to another location. The overworld editor can track these tiles and update their coordinates in the table when you move them.

Potential Star/Pipe Tile Issues

- Do not place a Star or Pipe tile at X=0 with Y>=0x20 (left side of the submaps section of the overworld), as they don't function correctly from there due to a limitation with one of SMW's tables. The editor will warn you if trying to link to a tile already at that location.

Relinking Star/Pipe Tiles (old method) Carefully follow this procedure to re-link any two star and/or pipe tiles: (Just make sure to sort out which two indexes you're going to use ahead of time.)

- Select the first tile, and enter this dialog. For the **Exit index to use,** select the first index you're going to use. Set the **Destination exit index** to "No Setting". Hit ok.

- Select the second tile, and enter this dialog. For the **Exit index to use,** select the second index you're going to use (must be different than the one you used for the first tile). Set the **Destination exit index** to the first index you're using (the same one you set for the first tile). Hit ok.
- Select the first tile again, and enter this dialog. You can now set the **Destination exit index** to the second index you're using. Hit ok.

And that's it. The two tiles should now be re-linked to one another.

Overworld Overworld Menu : Link Star/Pipe/Exit Tiles Together

This dialog lets you link 2 Star/Pipe/Exit tile teleport locations together in one step. To use it select a layer 1 star/pipe/exit tile, use this menu command, then select the second tile to link to and use this menu command again to bring a dialog up (alternatively you can just point the mouse cursor at the first tile, use an alt-left click, point the mouse cursor at the second tile, and use alt-left click again to bring the dialog up).

The star/pipe/exit tiles are used by the game to allow the player to warp from one map of the overworld to another. Each tile is normally assigned it's own index number in either the star/pipe list or the exit tile list so that the game can keep track of the coordinates used for transporting Mario to another location. The overworld editor can track the tiles and automatically update the warp coordinates in the list if you move them.

When you bring this dialog up, you should select 2 unique indexes you want to use for the 2 tiles you've selected. If they already have indexes, their current indexes will be selected by default. For exit tiles, you must also select the side that the player will be entering the first tile from as this affects the coordinates (note that the player will still be moving in the same direction after they go through the warp). Once you hit ok, the location information for the 2 entries you selected will be updated and the tiles will be linked to each other so the player can teleport between them.

If you want to clear an entry for a tile that already has an index assigned to it, you can select "No Setting" for that tile and hit ok.

It's also possible to create a one-way link between 2 tiles by selecting "Create a one-way link to this tile" for one of the tiles before hitting ok. When you do this, the tile you select an index for will warp to the second tile, but the second tile's existing warp destination (if any) will not be changed. This can be used to create more complex warps. For example, you could have multiple pipes all go to the same pipe. Or you could have 3 pipes (A, B, and C) where pipe A goes to pipe B, pipe B goes to pipe C, and pipe C goes to pipe A.

To view the Star/Pipe locations in the overworld editor, use the Show Star/Pipe/Location Numbers command in the view menu.

To view the Exit Tile locations in the overworld editor, use the Show Exit Path Numbers command in the view menu. To see the tiles themselves, use the Layer 1 Mario Paths command in the view menu. The exit tiles are the path tiles that are in red.

Potential Exit Tile Issues

- If Mario's coordinates are off by even a single pixel when he hits an exit tile, it will malfunction and Mario will end up right back at the same location he was trying to leave. If relinking the exit tiles doesn't help and you've set the correct side to enter from, it means you haven't created proper layer 1 paths between your levels and Mario's position is a little bit off from where he should be normally. Examine your paths and either correct or simplify them.
- If you have not yet inserted the GFX as 4bpp then saved the overworld, the game will not reload any graphics

when you use an exit tile. This wasn't an issue in the original game since the main map and submaps all used the same graphics files. But if you've chosen different graphics files to load for submaps and you're still using 3bpp tiles, it's better to use a pipe or star road instead to force the correct graphics to load.

- The Layer 3 tilemap is not reloaded. If you are using different overworld borders between submaps, you should consider using a pipe or star tile instead.
- The game sets Mario's new screen position after touching the exit tile, but *before* fading the screen out. This is noticeable if you happen to use an exit tile that goes to the same screen that you're currently in. What you may not expect is that it can also show up when going between the main map and submaps! This is because the submaps are all stored together on a single map that is the same size as the main map, and if the coordinates are close enough you'll get the same effect. Keep this in mind when arranging the location of exit tiles on the main map compared to your submaps so you can avoid this minor graphical glitch.
- If you place exit tiles right beside one another you may have to relink them if you later move them apart again, as the editor may not be able to tell which index belongs to which tile.

Potential Star/Pipe Tile Issues

- Do not place a Star or Pipe tile at X=0 with Y>=0x20 (left side of the submaps section of the overworld), as they don't function correctly from there due to a limitation with one of SMW's tables. The editor will warn you if trying to link to a tile already at that location.

Overworld Overworld Menu : Edit Boss Sequence Text

In this dialog you can edit the text that appears when you beat one of the 7 castles in SMW. As the controls are pretty much the same as for the Edit Message Box Text dialog, check there for a description of what you'll find in here.

If you ever accidently corrupt the castle text tilemap data in the game by patching over it and can't seem to get rid of it, hit control+shift+F8 in the overworld editor and save to re-initialize it.

Text editing is not supported for the Japanese version of SMW.

Overworld Overworld Menu : Edit Message Box Text

Probably one of the first questions you'll have when you see the message box editor in the overworld menu is "Shouldn't this be in the level editor instead of the overworld editor?" The answer is no. As it turns out, each level on the overworld is assigned two messages based on its level number. When you enter a level from the overworld, these two messages are used no matter which other levels you happen to go to through doors, pipes, etc before getting back to the overworld.

Within this dialog, you can edit the text for all the message boxes in the game, and even add messages to the levels that didn't have any before (thanks to some ASM reprogramming). Any message you're editing will be saved automatically whenever you switch messages or hit ok. At the top of the dialog is the list box for all the overworld levels, with 2 entries per level. Under this on the left are the font tiles (check the Edit Level Names menu for a description of these and of escape sequences), beside it is the preview area for the message, and below that is a text edit area for the currently selected message. Hitting the cancel button will undo all the changes you've made since opening the dialog.

You'll also notice some options beside the text editor. **Word Wrap** is pretty self-explanatory. **Auto Space** will distribute extra spaces remaining at the end of the line between words on the line, much like the original game does. Auto spacing does not occur on a line if it or the next line starts with a space, or if you're on the last line of the text box. You can turn these options off if you want full manual control over how the text will appear. **Zoom Text** displays the text at x2 size and stretched to fit the dialog. **Level Font** will use the font (which is from LG3 and LG4 of the Layer 3 GFX) and colors of the level currently loaded in the level editor. If you turn this off, it will use the default GFX and colors. If you change levels while this dialog is open, you can turn this off then on again to refresh the display.

The **Clear Text** button will clear the text of the current message, while the **Clear All Text** button will clear the text for all message boxes.

A few levels are special cases. Level 0 message 1 is used for the intro to the game, and message 2 is displayed when you get Yoshi for the first time. (The "Rescued Yoshi" message will only appear if the overworld level is on the "Yoshi's Island" submap. Getting Yoshi for the first time apparently isn't saved to SRAM, so it's possible to get the message again after loading a saved game.) For "Yoshi's House" level, instead of the message box displaying one of the two messages based on the X coordinate, the game will display message 1 if Yoshi hasn't been rescued yet, and message 2 if he has. In the switch palace levels, message 1 will have special blocks added to it and will also activate the "!" blocks that fly off of the level on the overworld. You can change some of the levels that use this special code in the Change Special Message Levels dialog.

As the game now uses text code 0xFE for the end of message indicator, you cannot use this tile number in your message box text. Not like you'd want to anyway, as it's part of a layer 3 background.

Text editing is not supported for the Japanese version of SMW.

Overworld Overworld Menu : Edit Level Names

Each level on the overworld is assigned a level name based on its level number. Within this dialog, you can edit the level names for each of the overworld level numbers. The original setup Nintendo used for this was rather annoying to work with, so the game has been recoded to allow each overworld level to have a completely separate name.

On the right side of the dialog is the list box containing all 0x60 level name entries. Level 0 is usually left blank, but you can edit it if you like to change the text string that shows up when Mario is on a pipe tile. Any item you select in this list will be copied into the edit control at the bottom of the dialog for editing, and the item you were editing before will automatically be copied back into the list.

Under this is a drop down box that lets you select which **submap font** (which is from LG3 and LG4 of the Layer 3 GFX) and colors to display for the preview area. You can also select to show the Special World passed colors.

The **Clear Name** button will clear the text of the current level name, while the **Clear All Names** button will clear the names for all levels.

On the left side of the dialog are the current font graphics available to choose from. Clicking on one of these tiles will result in the letter or escape sequence for it being inserted at the current cursor position of the level name being edited in the box at the bottom of the dialog.

Just above the edit control at the bottom is a one-line preview area where you can see what your level name is going to look like in the game. Beside it is a smaller nonstretched version. It will update itself as you type in the edit control.

Since not all the font characters can be represented by keystrokes (Yoshi's paw, for example), the edit control allows you to enter escape sequences using the backslash character (\) followed by a 2-digit hex code to indicate the tile number. You don't need to know the exact tile number though, as you can just click on the tile you want to insert into the level name. Nintendo did use a few "squashed" tiles to squeeze in the full level name a few times, which will show up as escape sequences in the editor.

Text editing is not supported for the Japanese version of SMW.

Overworld Overworld Menu : Change Boss Sequence Levels

When you pass a Boss, the game checks a list of overworld level numbers to determine if it should display a special event sequence. This usually consists of showing Mario destroying the castle with a text message, or ending the game. This dialog will let you change which level numbers the game will check for to display each sequence.

The two end game selections are identical, it's just that Nintendo needed to use separate level numbers for the front and back doors to Bowser's castle.

The earthquake sequence is the one used when you pass the sunken ship in the original game.

You can set one level to activate secret exit 1 when a boss is defeated (normally it would activate the regular exit), which was used in the game for the Big Boo boss but will work for any boss. There is actually no sequence associated with this.

Also note that the Big Boo boss and Reznor count as regular bosses, so they can be used to activate a boss sequence.

Overworld Overworld Menu : Change Special Message Levels

For the switch palace messages, the game inserts a few extra graphics into message 1 and activates the "!" blocks that fly off on the overworld map after hitting the switch. And for Yoshi's house, the message box text is determined by whether or not Yoshi has been rescued yet. This dialog will let you change which level numbers the game will check for to determine if special processing for the message is required.

This function is not supported for the Japanese version of SMW.

Overworld Overworld Menu : Change Special World Levels

When you pass the last level in Special World, SMW does a few things to change the appearance of the game. This dialog will let you change which level numbers the game will check the passed state of to determine if the "Special World Passed" effects should be enabled.

The **Change Overworld colors if passed level** setting will control on which level the special overworld colors will be activated. The **Change Koopa palettes if passed level** setting will control on which level to activate the special Koopa palettes (green shells turn yellow and red shells turn blue). And the **Change Sprite Graphics if passed level** setting will control on which level to activate the special graphics for levels (Bullet Bills turn into Pidgets and Koopas resemble Mario).

The game checks the "level has been passed" flag of the level number in question to determine when to activate the special settings. It should be pointed out though that for the overworld colors setting, the flag is only checked when the screen is refreshed... the problem is that when you pass a level the flag is typically set AFTER the overworld screen has already been redrawn! The reason you don't notice this when you finish Special World in the original game is because Nintendo set the colors for the special palette of Special World to the same colors they were in the standard palette.

If you don't ever want the special effects to be enabled, just change the level numbers to some other level that cannot be passed or isn't used, such as level 0.

Overworld Overworld Menu : Change No-Auto-Move Levels

In the original game, when you pass a switch palace you don't automatically move in the enabled direction so that the save prompt can come up immediately. This list determines on which level numbers the auto-movement will be turned off.

You'll note that there are 6 entries, not just 4. Four are for the switch palaces, one is for Yoshi's house, and one is for the unused red switch palace in Bowser's map.

If you don't want any level to have this property, just change the level numbers to some other level that cannot be passed or isn't used, such as level 0.

Overworld Overworld Menu : Edit Sprite List

This opens up a dialog where you can change/edit/delete the sprites on the overworld. If you wish to alter the sprite positions, check the Sprite Editor Mode menu.

Entry 4 should be left as "Nothing", as setting it to something else can cause bits of the sprite to appear at other locations on the screen while a path is being created on layer 2. Be careful not to set a slot to a "Not Used" index unless you've inserted your own custom sprite ASM for it, since otherwise it will most likely crash the game.

Most sprites will only appear on the main map, unless otherwise specified. A sprite list description is provided below:

Lakitu	A lakitu sprite that flies around and follows Mario on the overworld screen. The sprite was never used in the original game.
Blue Bird	Acts much like the lakitu. The sprite was never used in the original game.
Fish x3	This creates 3 fish for the main map, that jump up when Mario walks across one of the 3 fish path tiles. If you attempt to fill more than one slot with this, you just get more than one fish at the same locations as the first 3.

Pirhana Plant	A small pirhana plant. The sprite was never used in the original game.
Moving Cloud	A cloud that scrolls on the overworld. As there's already 3 of these that randomly appear on the map, there's little point in wasting slots on more.
Koopa Kid x3	This creates 3 Koopa Kid sprites, which will remain invisible until Mario walks on one of 3 specific layer 1 tiles, causing one of the 3 sprites to appear and move one tile to the left to intercept Mario. If Mario isn't there, the sprite will pause then move back and repeat the action until Mario gets off the tile. But if Mario is there and the layer 1 tile used as the trigger is a level tile, the player will be forced to play the level immediately. If the player passes the level, the sprite will no longer appear. But if the player dies in the level or uses Start-Select to exit, Mario will instead be teleported to one of 3 corresponding locations on the main map (which can be set with the Set Koopa Teleport Here dialog). The 3 tiles that trigger each of the 3 kids to appear can be changed in the “layer 1” section at the bottom of the dialog. Note that the sprite appears to be intended to pop out from under a layer 1 tile that has its layer 1 priority bits enabled. This sprite was never used in the original game (technically it was on the map, but the tile setting would

	have prevented it from ever showing up). To view the teleport locations, use the Show Koopa Teleport Numbers command in the view menu. To figure out which of the 3 kids corresponds to which of the 3 numbered teleport locations or the 3 trigger tiles, look at the subslot number when viewing the Sprite Data (Hex Code) which is enabled in the view menu.
Smoke x2	This creates 2 smoke sprites, one for the main map and one for Yoshi's island. The one for Yoshi's island cannot be moved to any other submap. Do not fill more than one slot with this, as additional smoke sprites will appear with messed up graphics.
Bowser Sign	The sign for Bowser's castle. It will not appear on anything other than Bowser's submap.
Bowser	A sprite of Bowser flying around. It will not appear on anything other than Bowser's submap, and it will not show up unless it's in the 7th slot (which means you can't have more than one).
Ghost x2	This will create 2 ghosts, one for the main map and one for the submaps. Putting a ghost in any slot other than the last 3 isn't recommended, since then the submap ghost will appear in a random position.

Overworld Overworld Menu : Edit Reveal Tile List

This opens up a dialog where you can edit which tiles for layer 1 can be "revealed" into other tiles after an event, intended to show the next level on the map after you pass a level. The default setup is usually fine for most people though. Note that there's 3 extra entries for tiles that weren't used in the original game (tiles 52,53,54).

Although you can only specify tiles on page 0, tiles on other pages will reveal into tiles on the same page they came from.

Overworld Overworld Menu : Edit Submap Music Selection

Each submap in Mario World has its own theme music. In this dialog, you can change which submap (displayed on the left) plays which music track (displayed on the right). Do not use the "Not Used" music choices, as doing so may freeze the game.

Overworld Overworld Menu : Extra Options

This opens up a dialog where you can modify a few extra overworld options.

Each of these options make a very minor ASM modification to the ROM, but they can be reversed whenever you want simply by changing the options and saving the overworld again.

The **Allow using Start-Select to Exit Passed Levels** option can disable a player's ability to use Start-Select to exit a level that has the "level passed" flag set. The "level passed" flag itself is unaffected by this setting.

The **Allow using Start to Scroll on Main Overworld Map** option can disable a player's ability to use Start to scroll around on the main map to view other areas.

The **Allow hardcoded default layer 1 paths** option can disable the 5 hardcoded default layer 1 Mario paths that the game used for certain levels or positions to allow some paths to cross over one another. In the original game these connect level 9 with 15 in Donut Plains, level 23 with 1B in Chocolate Island, level 24 with a pipe (by position) in Chocolate Island, level 11F with 120 in Forest of Illusion, and level 10C with 10D in Valley of Bowser. Most of the time you can just ignore them since starting a path with a regular path tile will override them anyway, but if you need to you can turn them all off.

The **Allow players to exchange lives** option can disable all players from being able to exchange lives. Life exchange can normally be done by pressing L/R on the overworld, or when a player gets a game over screen.

The **Allow using L/R for life exchange** option can disable all players from being able to use L/R to exchange lives. If disabled, the only other way to exchange lives would be from getting a game over screen. Note that if life exchange itself has been disabled, this option has no effect.

The **Allow using L+R to Enter Destroyed Castles and Fortresses** option can disable a player's ability to enter certain destroyed levels using L+R. This option is not available for Japanese ROMs, as apparently Japan's version of SMW doesn't allow using L+R for this in the first place.

The **Allow level 24 to redirect screen exits** option can prevent this overworld level from adjusting where Mario ends up when going through a pipe or door based on coins collected or time remaining. It's recommended to disable this if you're using this level for your own game, to avoid this behavior.

The **Save game after intro message** option will make the game save after the intro message from sprite 19 at the beginning of the game even if the overworld map start position has changed. The original game already did this as part of the little intro march on the overworld, but that normally gets disabled once you move Mario's starting position.

The **Bring up a Save Prompt when the player passes a Castle, Fortress, Ghost House, Switch Palace, level tile 0x80, or level tile 0x7E** option is pretty much self-explanatory.

The **Hide second ghost of sprite slot C until event passed** option causes the ghost that was in the Forest of Illusion submap in the original game not to be shown unless any of the listed events have been passed. While you can list more than one event, they must all be within the same group of 8 (so from 0-7, or 8-F, or 10-17, etc) because of how the game coded it.

The **Disable event path fade effect** option turns off the way event tiles appear to fade in during each step of path revealing on layer 2. The fade effect imposes rather severe restrictions and consumes a surprising amount of resources: colors 1-7 of palettes 0-3 and C-F in the palette table get overwritten, colors 8-F of palettes C-E are zeroed, colors 8-F of palettes 0-2 are constantly set and cannot be used in palette ExAnimation, palettes C-F are made translucent in sprites, GFX slots SP3 and SP4 must be set to the same files as FG1 and FG2, and layer 2 event path tiles plus any tiles overwritten by layer 2 event path tiles must only use 8x8 tiles from FG1 or FG2 and only use colors 0-7 of palettes 4-7. Disabling the effect removes all these restrictions. You can also then use the **Path Reveal Speed** setting to control how fast the game reveals each step of the path. The **Enable translucency for palettes C-F in sprites** option can also be turned back on if you need it for custom sprites (this just turns on the bit in CGADSUB for sprites).

The **Make lightning use colors from ROM instead of palette** option causes the lightning palette animation to read the colors from the shared palette in the ROM instead of from the current submap's working copy of the palette in RAM. By doing this, it effectively frees up colors 9-F of palette 2 for other uses in submaps that use lightning (assuming you've enabled custom palettes).

The **Merge FG1 and FG2 into SP3 and SP4** option causes FG1-2 to share the same area in VRAM as SP3-4, leaving room for 2 additional FG slots to be used. Since in the original game SP3-4 already had to use the same GFX files as FG1-2 to create the event path fade effect, merging them together will make sense for most people. The only time you might not want to do it is if you intend to use all of SP3-4 to create new custom overworld sprites after disabling the event path fade effect. Most of SP2 is already free for new sprites though, so you'll have to decide for yourself if you'd rather use those 2 GFX slots for new sprites or new FG graphics (or a mix of both, since if the slots are merged together then the tiles can be accessed by both sprites and FG layers).

Overworld Options Menu : Animation Rate

This will allow you to change the rate that Lunar Magic displays animations from the game at. The lowest setting is 7.5 fps, which is the default used in versions prior to 2.10. Since SMW's tile animations are 7.5 fps, this is generally adequate although the flashing level palette animation will be at half speed.

15 fps is the default since version 2.10 of Lunar Magic. At this speed, the flashing level animation will be correct and you can view ExAnimations that use the double speed trick. This is usually sufficient for most users.

30 and 60 fps will let you view ExAnimations that use faster speeds. 60 fps will also let you check to see if your animations will actually overwrite the game's original animations. If you aren't doing either of these however, you don't need to set it this high. These two settings will not have any effect for Win9x/ME users... it requires Win2k or higher.

Remember that each setting requires twice as much computing power as the one before it, but it also depends on the size of the program's window. If you have a half-decent system though, it shouldn't matter much. With Lunar Magic maximized on a 1024x768 screen on a Core2 2.4 GHZ computer and animation set to 60 fps, it doesn't even take up 1% of a single core. However performance on Windows Vista will be lower due to Microsoft's removal of hardware acceleration for GDI functions and other changes. Performance on Windows 7 may vary depending on your video card and driver and whether Aero is enabled.

Overworld Options Menu : Allow Move/etc on Layer 1 Tile Settings

When you move around tiles on layer 1 (in the 16x16 editor mode, not the event mode), the overworld editor automatically tracks and updates certain tables of sprite, layer, and VRAM positions. It also tracks other settings such as the level and event number for the tiles, and the directions to enable.

Disabling this setting will cause the overworld editor to NOT track and update these settings as you move things around. About 99% of the time that would be an extremely BAD thing to do, but it can be useful if you want to replace a single level tile with a different one in the same position while preserving the settings for the first tile.

In all other cases however, it is very strongly recommended that you leave this setting on at all times!

Overworld Options Menu : Auto-Deselect on Editor Select

This option causes any selected items in the main overworld editor to be deselected automatically if you make a selection in one of the editor windows related to the current editing mode. This can make things a bit easier if you don't want to have to remember to hold down control when right-clicking to paste into the main window from a selector window.

Overworld Options Menu : Allow Tile Resize Without Ctrl

This option allows tiles to be resized with the mouse when the resize mouse cursor appears on tile borders without having to hold down the control key. If you find it gets in the way, you can turn it off.

The key will still be required in the layer 2 8x8 editing mode if zoom is below 200%.

Overworld Help Menu : Contents

Opens up Lunar Magic's help file.

Overworld Help Menu : About Overworld Editor

F9 was used by the programmer a lot,
to see what the pointers would do.
F8 is a key everyone forgot,
to stop the hack of blue.
5 were the keys to press for M16,
reduced from 7 due to a request by Smalls.
5 and 20 was also the day of presents and trees,
so you could use bitmaps to deck the halls.
Four were the choices in dates,
a fun little game for all.
Three keys there are that still wait,
to hold the sprites in thrall.
Two were the names of girls from beyond,
both pink haired and fair.
One was the title of a movie with Bond,
that made a star appear in the air.

One Editor to rule them all,
one Editor to find them.
One Editor to bring them all,
and in 65816 bind them.

In FuSoYa's Niche,
where the lost key lies.

The end.
No, really!
I mean it this time!!
...maybe.

Command Line Arguments

Starting in version 3.00, Lunar Magic supports a limited number of command line functions for some common tasks. The syntax for the commands available are:

```
"Lunar Magic.exe" -ExportGFX "ROMFileName.smc"
"Lunar Magic.exe" -ExportExGFX "ROMFileName.smc"
"Lunar Magic.exe" -ImportGFX "ROMFileName.smc"
"Lunar Magic.exe" -ImportExGFX "ROMFileName.smc"
"Lunar Magic.exe" -ImportAllGraphics "ROMFileName.smc"
"Lunar Magic.exe" -ExportLevel "ROMFileName.smc"
"MWLFileName.mwl" LevelNumber
"Lunar Magic.exe" -ImportLevel "ROMFileName.smc"
"MWLFileName.mwl" [LevelNumber]
"Lunar Magic.exe" -ImportMap16 "ROMFileName.smc"
"Map16FileName.map16" LevelNumber [X,Y]
"Lunar Magic.exe" -ImportCustomPalette "ROMFileName.smc"
"PalFileName.pal" LevelNumber
"Lunar Magic.exe" -ExportSharedPalette "ROMFileName.smc"
"PalFileName.pal"
"Lunar Magic.exe" -ImportSharedPalette "ROMFileName.smc"
"PalFileName.pal"
"Lunar Magic.exe" -ExportAllMap16 "ROMFileName.smc"
"Map16FileName.map16"
"Lunar Magic.exe" -ImportAllMap16 "ROMFileName.smc"
"Map16FileName.map16"
"Lunar Magic.exe" -ExportMultLevels "ROMFileName.smc"
"DirectoryPath" [Flags]
"Lunar Magic.exe" -ImportMultLevels "ROMFileName.smc"
"DirectoryPath" [Flags]

"Lunar Magic.exe" -TransferLevelGlobalExAnim
"DestROMFileName.smc" "SourceROMFileName.smc"

"Lunar Magic.exe" -TransferOverworld "DestROMFileName.smc"
"SourceROMFileName.smc"
"Lunar Magic.exe" -TransferTitleScreen "DestROMFileName.smc"
"SourceROMFileName.smc"
"Lunar Magic.exe" -TransferCredits "DestROMFileName.smc"
"SourceROMFileName.smc"
```

```
"Lunar Magic.exe" -ExportTitleMoves "ROMFileName.smc"
"TitleMovesFileName.zst"
"Lunar Magic.exe" -ImportTitleMoves "ROMFileName.smc"
"TitleMovesFileName.zst"
```

Arguments with [] are optional. All numbers are in hex unless otherwise stated.

ImportMap16 is not for full game Map16 tileset files, and **ImportCustomPalette** is not for shared palette files (there are separate commands for those). The coordinates for ImportMap16 are optional unless you're importing an old .bin file.

For ExportMultLevels/ImportMultLevels, Flags is a hex value where each bit controls an option. If the flags value is not supplied, the default settings are used. For **ExportMultLevels**, 1 will export modified files only (default is on). For **ImportMultLevels**, 1 will clear all secondary exits in ROM before importing (default is off... note however that LM already clears the secondary entrances of the target level number before importing via command line). 2 will enable auto-set number of screens (default is on).

Most warnings and error messages from Lunar Magic will still be displayed using message boxes.

Note: Since Lunar Magic is not actually a console program, you may notice a few cosmetic oddities if running it in a console window. If not running it from a .bat file, the console will not wait on LM to finish. On WinXP the command prompt may not display afterwards unless you hit enter. For versions of windows before WinXP, you will not see any output from LM in the console but you can still redirect LM's console output to a file to see it that way (the file will use UTF-8).

RATS (ROM Allocation Tag System)

When Lunar Magic attempts to save something in the expanded portion of the SMW ROM, it looks for free space by scanning for consecutive bytes with a value of 0. However, this simplistic method is not always adequate since there are some types of data that contain areas filled with bytes using that value. Therefore, a tag system was created by FuSoYa to ensure that data would not be overwritten unintentionally. This tag system is referred to as RATS (ROM Allocation Tag System).

When Lunar Magic stores data to the ROM, it prepends a tag to the data that indicates the size of the data stored after it. This makes scanning for free space much safer, as the program will automatically scan and skip over the areas defined by the tags. Erasing the data is also safer, since instead of parsing the data a program can simply examine the tag to find out how large the section of data is before deleting it.

Since this also makes it much easier for third party programs to locate free space for storing data, it is **highly** recommended that any data you store in the expanded portion of the SMW ROM be protected with RATS.

Note that data stored within the original SMW ROM space (first 512 KB) does not use RATS. Be very careful using space in this area, as either Lunar Magic or other programs may modify it without warning. For ASM hacks, it's usually best to just put in a JSL then put your code in the expanded area of the ROM with a RATS tag.

You can use the Scan ROM menu to get some stats on RAT usage in the user area of the ROM currently open in LM.

There is also an option in the restore system to warn you of nested RATs found, and a general option to warn you when RATS.log gets written to (which can happen when LM notices something odd about RATs in the ROM when saving).

Implementation Details

A RAT is exactly 8 bytes. The first 4 bytes are the tag identifier “STAR”, which is just the word RATS reversed in all-caps. The next 2 bytes contain the size-1 of the data protected by the tag in little endian format, not included the space taken up by the tag itself. The last 2 bytes are the bitwise inverse of the previous 2 bytes (obtained by XORing with 0xFFFF), which is used to verify the tag is correct.

Given the 2 byte size field, this means the amount of bytes protected by a single RAT can range from 1-0x10000 bytes. 0x10000 bytes happens to be the size of 2 LoROM banks.

Example:

To RAT protect the word “FuSoYa” in the ROM, the tag and data should be 53 54 41 52 05 00 FA FF 46 75 53 6F 59 61 in a hex editor (8 bytes for the tag, 6 bytes for the data).

When searching for free space, you check for consecutive zeros but must also check for the RATS identifier. If you come across the identifier, check that the size and inverse fields will give 0xFFFF when XORed together. If they don't, the tag should be considered invalid and ignored. If it is valid, you must jump over the data protected by the RAT and continue scanning.

When erasing data protected by a RAT, you must still check that the identifier exists, and that the size and inverse fields will give 0xFFFF when XORed together to verify that the tag is valid.

Nested RATs are not allowed!! There should not be any RATs within an area that is protected by another RAT. If this occurs, it's an indication that something has gone wrong. While Lunar Magic does have a few extra checks for nested RATs to preserve data integrity, you should not depend on this behavior, and third party programs may be even less forgiving.

When writing your own code to save RAT protected data, make sure you check that the data you're saving doesn't have a size of 0, since technically the format doesn't allow for a RAT that protects 0 bytes of data. The size field is (size-1), so in that case your code might calculate the size field to (0-1)=-1, which would become 0xFFFF in hex… and would get interpreted as a size of 0x10000 when used in the tag!

If you don't want to write code for managing RATs, you can use the Lunar Compress DLL on FuSoYa's page, which has RAT functions in it.

64 Mbit (8 MB) ExLoROM Support

Normal LoROM games can go up to 32 Mbit (4 MB) in size. But if you need an insanely large amount of space, Lunar Magic can go beyond this by converting the ROM to a 64 Mbit (8 MB) ExLoROM game. However, this involves moving the original ROM banks around and not all emulators or other SMW utilities support this map type. Also, any custom ASM hacks already inserted by other programs that used SNES pointers with bank bytes within the 80-FF range instead of the standard 00-7F range (for example, a standard FastROM patch) will malfunction or crash and will need to be manually repaired or moved.

Therefore, usage is not recommended except for users that truly need the space and know for sure that their other programs, patches, and custom ASM are ExLoROM compatible. Lunar Magic normally won't expand to this size until it runs out of room in a 4 MB ROM, but you can press Shift + Page Down in the main level editor to force the conversion to be done immediately. You will need Snes9x version 1.39a to 1.43 or 1.54+ to play the ROM (Snes9x 1.50-1.53 limits ExLoROM to 48Mbits, and the official build of ZSNES doesn't support it at all). Or you can use one of FuSoYa's custom 8MB Snes9x or ZSNES builds.

64 Mbit (8 MB) SA-1 Support

If you are using SA-1 in your ROM, Lunar Magic supports up to 64 Mbit (8 MB) ROMs. However you must set the SA-1's bank switch registers in $2220-$2223 to the expected values of 4,5,6,7 to be able to access all 8MB of the ROM at once (if using Vitor Vilela's SA-1 patch, Lunar Magic can set these for you). When you do so, the first 2 MB of the ROM will be mapped to $00-$3F:8000-FFFF, the next 2 MB will be mapped to $80-$BF:8000-FFFF, and the last 4 MB will be mapped to $C0-$FF:0000-FFFF. This means the first 2 MB of the ROM has the same map as a regular LoROM game would, so if you're using utilities or patches that do not recognize this SA-1 map, it would be best to use them in this area.

For SA-1 ROMs larger than 4MB, if you want to use Snes9x it requires version 1.54+ or FuSoYa's custom 8MB Snes9x build. For SA-1 ROMs larger than 6MB, if you want to use ZSNES it requires FuSoYa's custom 8MB ZSNES build.

Note that the official build of ZSNES 1.51 and lower has a bug with SA-1 ROMs larger than 4 MB where the first 2 banks past 4 MB cannot be accessed. So if you expand your ROM with Lunar Magic, the program will work around this by pre-filling these 2 banks to make them unavailable for saved data. The banks will be freed up if you later expand to 8 MB.

If you want to use SA-1 in your ROM but don't know much about it, look for Vitor Vilela's SA-1 patch on SMW Central.

GFX File Information

The original game used 3bpp (bits per plane) graphics for most of the GFX files (see chart below) as a method of saving space. Lunar Magic installs an optional 4bpp ASM expansion when inserting the GFX so that you can use the full 16 colors per palette for these files. But to deal with a few compatibility issues, the ASM hack will reconvert some tiles back to 3bpp when needed. Currently only portions of files F and 21 are affected.

If for some reason you choose not to use the 4bpp hack, keep in mind that Lunar Magic still automatically converts the 3bpp tiles into 4bpp tiles when extracting to files since many tile editors cannot handle 3bpp tiles. The files are then later internally converted back to 3bpp during insertion, so it's important to remember that the upper bitplane will be lost in the actual game! In other words, when not using the ASM hack you can only use the first 8 colors of the palette in any file marked as 3bpp, and not the full 16 you would normally expect with 4bpp tiles.

Original Graphics Files (GFX)

These are extracted to the "Graphics" folder in the same directory as the ROM. Each file is named "GFX#.bin", where "#" is a hex number from 00-33. By default they're normally in split individual files, but can be extracted to a single file called "AllGFX.bin" if the Use Joined GFX Files option is enabled (which some people find easier when starting out, but split files tend to be more convenient in the long run).

Be careful not to make these files larger than they should be, as that can cause issues in the game.

Note: Lunar Magic has the ability to load external files in place of GFX32.bin and GFX33.bin for display purposes in the editor. Just place files called ExternalGFX32.bin and ExternalGFX33.bin in the Graphics folder for your ROM (they will not be inserted into the ROM).

GFX File Number	**Bits Per Plane (original game)**	**Bits Per Plane (Lunar Magic)**	**File Size**
00-26	3	4	4KB
27	(Mode 7)	(Mode 7)	3KB
28-2B	2	2	2KB
2C-2E	3	4	4KB
2F	2	2	1KB
30-31	3	4	2KB
32	4	4	23.2KB
33	3	4	12KB

Extra Graphics Files (ExGFX)

These are held in the "ExGraphics" folder in the same directory as the ROM. Each file is named "ExGFX#.bin", where "#" is a hex number from 80-FFF (do not insert a leading zero for numbers less than 100). These do not exist in the original game and are supplied by the user.

The files sizes can vary depending on what you're using them for. Files meant for regular FG/BG/SP slots are typically 4KB, graphics for layer 3 slots are normally 2KB, and layer 3 tilemap files might be 2, 4, or 8KB.

Files 60-63 are special as they are stored in the ROM uncompressed and can be up to 32KB (0x8000) bytes in size. They are only intended for use in ExAnimation.

ExGFX	**Bits Per Plane**	**File Size**

File Number		
60-63	Various	Up to 32KB
80-FFF	Various	Various

External Graphics Files (ExSpriteGFX)

These are held in the "ExternalGraphics" folder in the same directory as the ROM. Each file is named "ExSpriteGFX#.bin" where "#" is a hex number from 00-07. These do not exist in the original game and are supplied by the user.

The files can be up to 32KB (0x8000) bytes in size. They are not inserted into the ROM. They are only intended for use in displaying dynamic type sprites in the editor that don't store all their frames in VRAM at once. See Custom Tooltips for Sprites on how to use them.

ExSpriteGFX File Number	Bits Per Plane	File Size
00-07	4	Up to 32KB

External Palette Files (ExSpritePalette)

These are held in the "ExternalGraphics" folder in the same directory as the ROM. Each file is named "ExSpritePalette#.mw3" or "ExSpritePalette#.pal" where "#" is the hex number 00 (as only 1 file is currently supported). These do not exist in the original game and are supplied by the user.

The .mw3 file is just raw SNES 16-bit colors in little endian format, while the .pal file is in YY-CHR's format. Note that you can only use 1 file type for each hex number.

The files can hold up to 0x400 palettes of 0x10 colors each. For an mw3 file that would be up to 32KB (0x8000) bytes in size, and for a .pal YY-CHR file that would be up to 48KB (0xC000) bytes in size. They are not inserted into the ROM. They are only intended for use in displaying dynamic type sprites in the editor. See Custom Tooltips for Sprites on how to use them.

GFX Decompression Routine Information

This section is for 65816 ASM programmers only. Do not email me questions about this.

Since version 1.70, Lunar Magic has provided a subroutine that allows decompressing a GFX or ExGFX file to a specified location in RAM. You can access this by executing a JSL $0FF900.

The entry/exit conditions for the subroutine are as follows:

On Entry:
A should contain the 16 bit GFX/ExGFX file number you want to decompress.
$00 in RAM should contain the 24 bit address of where to decompress the data to.

On Exit:
Processor bits are preserved.
Contents of X are preserved.
Contents of Y are preserved.
A is not preserved.

Note that you cannot currently use this routine to decompress GFX files 32 or 33.

Screen Exit Routine Information

This section is for 65816 ASM programmers only. Do not email me questions about this.

Since version 3.00, Lunar Magic has provided a subroutine that calculates which screen Mario is currently on for screen exits. You can access this by executing a JSL $03BCDC.

The entry/exit conditions for the subroutine are as follows:
On Entry:
A/X/Y are 8 bit.

On Exit:
A/X/Y are 8 bit.
X holds the screen number.
Contents of A/Y are modified.
$00 in RAM (16 bit) may be modified.

Map16 Gameplay Programming Information

This section is for 65816 ASM programmers only. Do not email me questions about this.

Lunar Magic inserts code that allows one to make any 16x16 tile behave like another tile by intercepting and altering the tile value that is reported to Mario World's game engine. But this doesn't help when the behavior you want doesn't exist in any of the tiles that Nintendo has already made. For example, the breakable brick in the Demo World hack requires custom code to define its behavior.

To make this task somewhat easier, I've left a bit of extra space in Lunar Magic's FG Map16 re-mapping code for JSL instructions. This can be used to implement your own custom routines to check for certain tiles. You can examine the Demo World hack to see how the breakable brick routines were integrated with this.

Below you'll find the information necessary for setting up your own gameplay ASM code to interface with Lunar Magic's inserted code. What you will not find however is a definitive guide describing exactly how to code behavior for your own tiles. I recommend checking tutorials and doing several traces to discover the information you need to implement whatever behavior it is you're looking for.

Anyway, here are the PC file offsets in SMW that you can place your JSL's, and the actions handled by each:

0x37890	Mario touched a block from below.
0x378A0	Mario touched a block from above.
0x378B0	Mario touched a block from the side.
0x378C0	Mario touched the top corner of a block. *new to version 1.64
0x378D0	Mario is within a block (body). *new to version 1.64
0x378E0	Mario is within a block (head). *new to version 1.64
0x378F0	Reserved for future use.
0x37920	Sprite touched a block from above/below.
0x37930	Sprite touched a block from the side.
0x37940	Reserved for future use.
0x37950	Reserved for future use.
0x37980	Mario hit a block from the side with his cape.
0x37990	Reserved for future use.
0x379C0	Mario hit a block with a fireball.

Note that Mario can stand on the far edge of a block to trigger the “touched top corner” action without triggering the “touch from above” action. So most of the time you will want to handle both with the same code. Also, the “touched top corner” action only triggers when the tile's corner forms the left or right edge of a ledge... meaning if the tile has solid blocks on both sides and Mario walks across, the action will not occur.

There are 2 “Mario is within block” actions, one for the upper 16x16 half of Mario and one for the lower 16x16 half. If Mario is small or crouched down, he only occupies a single 16x16 so both actions will be triggered on the same block.

There are additional actions that LM's code does not currently check for, which may or may not be added in future versions. The ones already implemented are the most common types however, and you'll find that most of the other properties can be "borrowed" from other pre-existing tiles using Lunar Magic's tile gameplay setting in the Map16 editor.

For each of the locations above, there's enough room to place up to 3 JSL instructions (4 bytes each). While you technically only need one, the others might be useful for inserting the routines of more than one programmer.

The entry/exit conditions for a custom subroutine are as follows:

A/X/Y are 8 bit.
Contents of X should be preserved.
Contents of Y should be preserved.
A doesn't have to be preserved.

Y contains the upper byte of the 16x16 tile number that is going to be reported to SMW's game engine. You can find the lower byte stored at $1693. You may modify both of these if you wish, although remember that since LM's own gameplay setting code runs before yours, changing them will effectively override LM's setting. If you want to know the last tile number that was used by LM's tile gameplay lookup code, you can find the 16-bit value stored at $03. So for example if someone sets tile 0x201 to act as tile 0x200, and tile 0x200 is set up to act as tile 0x2B, $03 will contain 0x0200 and Y/$1693 will contain 0x002B. In cases where the tile is set to be itself (ie. setting tile 0x2B to act as tile 0x2B), $03 and Y/$1693 will be identical.

Since LM enforces using a gameplay tile setting less than 0x200, the value reported by Y/$1693 should always be below 0x200. $03 on the other hand can be from 0-0x7FFF. As most of the time you will want to code for tiles 0x200-0x7FFF, you should usually check the contents of $03 to detect which tile you're dealing with rather than using Y/$1693. Doing so means that once you've implemented a behavior for one tile, you can easily set other tiles to act the same way just by setting the gameplay property in LM.

A couple other memory locations you may find useful are:

$19	0=currently small Mario
	1=big Mario
	2=cape Mario
	3=flower Mario
$140D	0=Not a drill jump
	1=Drill jump
$192B	Current SP tile set (0-F for levels)
$1931	Current FG tile set (0-E for levels)
$1933	Which layer the block is on for main player interactions (0 is layer 1, 1 is layer 2/3). For sprite interactions use $185E, and for the cape and fire hits use $0F.

There's a lot of others, as well as subroutines within SMW that you can call to do certain tasks, but you'll have to check other more in depth documents for that or locate the rest on your own.

.map16 File Format

All values are little endian and 32 bit unsigned ints unless otherwise stated.

First 4 letters are "LM16"
2 bytes (WORD) for file format version number (default is 0x100, change major number to prevent opening in older programs).
2 bytes (WORD) for game ID (0 is none, 1 is SMW).
2 bytes (WORD) for program version.
2 bytes (WORD) for program ID (0 is none, 1 is LM). Ask FuSoYa if you want your program assigned a number, or just use 0.
4 bytes extra flags (set to 0).

4 bytes for offset of offset+size table in file.
4 bytes for size of offset+size table.
4 bytes size X (max is 0x10 for LM).
4 bytes size Y (max is whatever is currently scrollable in LM).

4 bytes for base coord X.
4 bytes for base coord Y.
4 bytes for various flags and info (can set to 0).

   04 FG relative base coords (2.50+)
   08 BG relative base coords (2.50+)
   Note:If neither BG or FG relative flags are set, assume absolute coordinates. For absolute, if Y is from 400-7FF you may assume that the file was meant to be compatible

with versions of LM before 2.50 and that you should adjust the Y coordinate to be relative to the BG area by first subtracting 0x400.

+0x14 unused bytes (set to 0)

0x40 bytes total

Optional Data here (comment field)
Lunar Magic x.xx
©2011 FuSoYa
Defender of Relm

Now the offset+size table (each index has 4 bytes for offset and 4 bytes for size)

0 Map16 data (8 bytes per 16x16=2 per 8x8, standard SNES tilemap format for each 8x8. Order for SMW is top left, bottom left, top right, bottom right)
1 Act As data (2 bytes per tile)
2-7 Only used for full game Map16 export/import, so set to 0.

0x40 bytes total

Then append all data referenced by the offset table.

Custom Toolbar/GUI Images

You can customize Lunar Magic's toolbars to use your own images for the buttons. For the main window just create a bitmap with a height of 16 and a width of 656 (or if you want to use buttons larger than the default 16x16 ones, set the height to whatever you like and the width to 41*height). Leave the first button blank, and fill in the rest based on the order of the buttons in the toolbar.

Rename your bitmap file to "Lunar Magic.ff4", and place the file in the same directory as the Lunar Magic exe file (make sure that viewing file types is enabled on your computer so that you don't end up renaming it to "Lunar Magic.ff4.bmp"). Then the next time you start Lunar Magic, it will use your bitmap to draw the toolbar buttons.

For the overworld editor toolbar, use a bitmap with a height of 16 and a width of 512 (or if using larger buttons, set the height to whatever you like and the width to 32*height) and rename it to "Lunar Magic.ff2".

For the overworld 8x8 selector buttons, use a bitmap with a height of 16 and a width of 96 and rename it to "Lunar Magic.ff1".

For the Map16 editor buttons, use a bitmap with a height of 16 and a width of 320 and rename it to "Lunar Magic.ff5".

For the palette editor buttons, use a bitmap with a height of 16 and a width of 192 and rename it to "Lunar Magic.ff3" (for the level palette editor) or "Lunar Magic.ff6" (for the overworld palette editor).

For the Add Object/Sprite editor buttons, use a bitmap with a height of 16 and a width of 96 and rename it to "Lunar

Magic.ff7" (for the Add Object editor) or "Lunar Magic.ff8" (for the Add Sprite editor).

For the background editor buttons, use a bitmap with a height of 16 and a width of 144 and rename it to "Lunar Magic.ff9".

To display a custom tiled image in the black non-editable areas in the level editor, use a bitmap of any size and rename it to "Lunar Magic.ffx". For the background editor, rename it to "Lunar Magic.ffxhd". For the overworld editor, rename it to "Lunar Magic.ffx2".

Minimum Height and Scaling: The minimum image height is normally 16. Note however that if the minimum height when adjusted for your screen's DPI is more than 25% larger than the image you provide, scaling will occur.

Custom Tooltips for Direct Map16 Tiles

You can customize Lunar Magic's tooltip text for individual tiles in levels. Keep in mind that these tooltips are not used for the objects, or the tiles that make up the objects... they're only for the tiles placed using the Direct Map16 Access feature.

To do this, create a UTF-8 text file with the same name as your ROM but with a .dsc extension (ie. marioworld.dsc) and put it in the same folder as your ROM file. The format of each line in the text file is:

Tile Number (in hex), TAB, Reserved Value (in hex), TAB, Tooltip text.

The reserved value should be zero for single tiles, or 1 to fill an entire "page" (0x100) tiles with that same tooltip.

If you want to include a comment in the file, start the line with a semicolon ";" character.

The custom tooltips will use a different background color than Lunar Magic's standard tooltips. But there are special escape sequences that you can include inside the tooltip text to change the text and background color of the current tooltip. The codes available are:

\n	Line feed for tooltip text.
\r	Carriage return for tooltip text (not needed).
\\	Lets you use the \ character.
\bXXXXXX	Change the tooltip background color to

	XXXXXX (24 bit RGB hex value).
\fXXXXXX	Change the tooltip foreground (text) color to XXXXXX (24 bit RGB hex value).
\dXXXXXX	Change the amount of time that the tooltip is displayed to XXXXXX ms (6 digit hex value).
\wXXXXXX	Change the width of the tooltip to XXXXXX pixels (6 digit hex value).

Other Effects (Editor Display Only)

Starting in version 3.31, you can have a tile that uses screen exits be treated as exit-enabled (for undefined exit scans or for viewing). Just add 20 to the reserved value for this.

Starting in version 2.41, you can make the tile appear translucent in the editor. Simply set the reserved value to 8 for this.

If however you want the tile to show a translucent placeholder tile to simulate an invisible object or a POW object, you must specify this on a separate line from the tooltip. To do this, set the reserved value to 2 (for an invisible object) or 4 (for an invisible POW object), and instead of the tooltip text put the hex value of the placeholder tile you want to display when the relevant view option in Lunar Magic is enabled.

For example, to make tile 0x300 appear to be an invisible question block, put this:

```
300 2 120
```

Starting in version 3.03, you can specify block contents to appear in the editor. You must specify this on a separate line from the tooltip. To do this, set the reserved value to 10 and instead of the tooltip text put the hex value of the sprite Map16 tile you want to display when the relevant view option in Lunar Magic is enabled.

For example, to make tile 0x300 appear to have a star inside, put this:

```
300 10 105
```

Custom Tooltips for Sprites

You can customize Lunar Magic's tooltip text for sprites. To do this, create a UTF-8 text file with the same name as your ROM but with a .ssc extension (ie. marioworld.ssc) and put it in the same folder as your ROM file. The format of each line in the text file is:

Sprite Number (in hex), TAB, Type (in hex), TAB, Tooltip text.

For example, to change the tooltip for Rex (sprite AB), put this on one line:

```
AB 0 This is Rex.  He's a cute little dinosaur that can get
squashed by Mario.  He takes two hits to kill.
```

The type value should be 0 for regular sprites, 10 for custom sprites that use an extra value of 1, 20 for custom sprites that use an extra value of 2, and 30 for custom sprites that use an extra value of 3.

For more information on special escape sequences that can be used in the tooltip text, see the Custom Tooltips for Direct Map16 Tiles topic.

Custom Sprite Tile Arrangements (Editor Display Only)

You can also modify a sprite tile arrangement displayed in LM by including mapping information in the tooltip file (note that this only affects the editor, not the actual game). The format of the line should look like this:

Sprite Number (in hex), TAB, Type, TAB, Xoffset (in dec), COMMA, Yoffset (in dec), COMMA, SpriteMap16Tile (in hex), SPACE (if putting in multiple tiles), etc (repeat starting at Xoffset if putting in multiple tiles)

For example, to change Rex into 3 stars, put this:

```
AB 2 0,0,105 -8,8,105 8,8,105
```

The X and Y offsets are in pixels and get added to the sprite's position in the level. The tile number is relative to the start of LM's internal sprite Map16 data. You can make the tile appear translucent in the editor by adding 0x8000 to the tile number. Do not include trailing spaces at the end of the line, or the next line may not be processed.

The type value should be 2 for regular sprites, 12 for custom sprites that use an extra value of 1, 22 for custom sprites that use an extra value of 2, and 32 for custom sprites that use an extra value of 3.

Starting in version 2.31, you can set separate appearances and tooltips based on the last 1, 2, 3, or 4 bits of the sprite's X and/or Y position. The original game also did this to vary the behavior/appearance of some sprites. To do this, treat the type value as a 4 digit value with the first digit representing the Y offset and the next digit representing the X offset, with the last 2 digits set to the type you would normally use. Note that if you don't set the appearance/tooltip for a particular offset, the editor will just use the one for X=0 Y=0.

For example, to change Rex into 3 stars or 3 flowers based on X&1, put this:

```
AB 0002 0,0,105 -8,8,105 8,8,105
AB 0102 0,0,104 -8,8,104 8,8,104
```

Starting in version 3.30, you can set separate appearances and tooltips based on the lower N bits (up to the full byte) of one of the sprite's extension bytes. You can only choose 1 extension byte to use per sprite, and you cannot use both this and X/Y based appearances on the same sprite. To do this, treat the type value as a 5 digit value with the first digit representing the byte # of the sprite to use for the appearance (the extension bytes start after the 3 original sprite bytes, so 3 is the minimum value), the next 2 digits as the byte value to check for, and the last 2 digits set to the type you would normally use. Note that if you don't set the appearance/tooltip for a particular byte value, the editor will just use the one for value=0.

For example, to change Rex into 3 stars or 3 flowers based on the lowest bit of the first extension byte, put this:

```
AB 30002 0,0,105 -8,8,105 8,8,105
AB 30102 0,0,104 -8,8,104 8,8,104
```

Starting in version 2.30, you can use a type value of 6 to define the sprite tile arrangement used by LM to display Mario at level entrances. In this case, the sprite number is actually the action that Mario takes when entering the level (0-6, check the action list in an entrance dialog for the order). Add 0x10 to the sprite number if you want to change the arrangement for when the "face left" option is on (if an arrangement for facing left for an action isn't specified, the 0-6 arrangement will be used).

For example, to change the "do nothing" Mario entrance into a translucent Mario, put this:

```
00 6 0,0,8300 0,16,8310
```

Starting in version 1.90, you can display a text label instead of a tile by using *LabelText* in place of the SpriteMap16Tile field (ASCII text only). This can be useful for sprite commands that have no relevant graphics to display. You may also use \n as a line feed within it. But you should only use this for short labels, as the user is going to be manipulating this label as though it were the actual sprite... use a tooltip for more descriptive text. Using Rex as another example:

```
AB 2 0,0,*This is Rex.\nCute eh?*
```

For more information on Sprite Map16, see the Custom Map16 for Sprites topic.

For the overworld: You can do the same thing for the overworld editor using the .sscov file extension. For custom sprites, do it the same way you would for a level by assuming the extra value is 1. Some things are not supported (no separate appearances/tooltips based on the sprite's X/Y position or other bytes, no changing Mario's appearance). If you want to display a shadow under the sprite, add 1 to the type value when doing the tile arrangement.

Custom Sprite GFX Information (Editor Display Only)

Starting in version 3.30, you can use a type value of 8 to set the slot GFX information that appears when you select a sprite in the "Add Sprites" window list. You can set this information for each X/Y based appearance or extension byte appearance. Note that if you don't set the information for a particular appearance, the editor will just use the one for extension value=0 or X=0 Y=0.

You can specify multiple GFX sets, where each set has 4 hex values separated by commas that correspond to the contents of slots SP1-4. If the sprite doesn't require any file in a particular slot, set it to 7F. If the sprite requires a specific combination of GFX files in more than one slot, you can add 8000 to the file numbers to force that combination to be displayed separately.

For example, to have Rex display "SP4=11/20", put this:

```
AB 8 7F,7F,7F,11 7F,7F,7F,20
```

Or if you want Rex to display "SP3=06 SP4=11 or SP3=13 SP4=20", put this:

```
AB 8 7F,7F,8006,8011 7F,7F,8013,8020
```

Using External GFX files for Sprites (Editor Display Only)

Starting in version 3.20, you can use graphics that come from external files that are separate from the ROM to display sprites. This may come in useful for dynamic type sprites that do not store all their frames in VRAM at once but only transfer their current frame to VRAM when needed.

See GFX File Information on where to store the files. To actually access the tiles in sprite Map16 however, you must set the base graphics for specified sprite Map16 tiles in the sprite tooltip file. The format of the line should look like this:

10000 (in hex), TAB, Type, TAB, StartSpriteMap16Tile (in hex), HYPHEN, EndSpriteMap16Tile (in hex), COMMA, Base8x8Tile (in hex), SPACE (if putting in multiple definitions), etc (repeat starting at StartSpriteMap16Tile if putting in multiple definitions)

For example, to set the base graphics for a page of SpriteMap16 tiles at 2300-23FF to the start of the first external graphics file for sprites, put this:

```
10000 0 2300-23FF,0
```

The first value (10000) is just a static number that should be the same for all definitions, the type value should be 0, the Start and End SpriteMap16 tiles for the range are relative to the start of the SpriteMap16, and the Base8x8Tile is relative to the start of the external graphics file area for sprites.

For the overworld: Only the first external file can be used.

Using External Palette files for Sprites (Editor Display Only)

Starting in version 3.20, you can use palettes that come from external files that are separate from the ROM to display sprites. This may come in useful for dynamic type sprites.

See GFX File Information on where to store the files. To actually access the palettes in sprite Map16 however, you must set the base palette for specified sprite Map16 tiles in the sprite tooltip file. The format of the line should look like this:

20000 (in hex), TAB, Type, TAB, StartSpriteMap16Tile (in hex), HYPHEN, EndSpriteMap16Tile (in hex), COMMA, BasePalette (in hex), SPACE (if putting in multiple definitions), etc (repeat starting at StartSpriteMap16Tile if putting in multiple definitions)

For example, to set the base palette for a page of SpriteMap16 tiles at 2300-23FF to the start of the first external palette file for sprites, put this:

```
20000 0 2300-23FF,0
```

The first value (20000) is just a static number that should be the same for all definitions, the type value should be 0, the Start and End SpriteMap16 tiles for the range are relative to the start of the SpriteMap16, and the BasePalette is relative to the start of the external palette file area for sprites.

Note that each palette is assumed to have 0x10 colors. However since the Map16 format lets you access 8 palettes, setting the base palette means you can access both that palette and the next 7 after it for use in that 16x16 tile.

Custom Map16 for Sprites (Editor Display Only)

You can customize the Map16 data Lunar Magic uses for displaying the sprites (note that this only affects the editor, not the actual game). This may be useful if you've modified the ROM to change some of the sprite tile numbers.

To view the sprite Map16 data in the Map16 editor, hit Ctrl-Shift-Page-Down and scroll down past the BG Map16 data. The first 4 pages (0x400 tiles) are reserved for Lunar Magic to display basic sprites, while the next 0x38 pages (0x3800 tiles) are intended for custom sprites.

If you want to save any changes you've made for custom sprites, hit Ctrl-Shift-F9 to save it to a file with the same name as your ROM but with a .s16 extension (ie. marioworld.s16) in the same folder as your ROM file. Whenever you re-open your ROM, your custom sprite Map16 will be loaded and used.

You can also save changes made to the first 4 pages reserved for basic sprites by hitting Ctrl-F9, which saves the data to a file using the .m16 extension. Your changes will then be loaded and used instead of LM's internal data whenever you re-open this ROM.

Warning: When a new version of Lunar Magic is released, it may contain updated basic sprite Map16 data. Using a .m16 file will override any new updated data, so you may have to adjust your changed Map16 data to match the new mappings or some sprites may not show up correctly in the editor.

It is also possible to change the tile arrangements LM displays for sprites, which may come in handy if you've done more extensive sprite modifications. See the Custom Tooltips for Sprites topic.

For the overworld: You can do the same thing for the overworld editor using the .s16ov file extension, which gives you 8 pages for custom sprites (there are 4 pages before that for basic sprites, but these cannot be modified). If you need to display a blank tile, sprite 8x8 tile 2 has been hard coded to show nothing in the editor.

Custom Collections of Sprites

The Add Sprites window has a custom section that allows you to store your own arrangements of sprites in external files for use with a particular ROM. This is mainly intended to hold custom sprites that are inserted by a third party program, but can be used for regular sprites as well.

Lunar Magic uses RomName.mwt for the text description list file, and RomName.mw2 for the actual sprite data to use. These should both be in the same folder as the ROM being edited.

Each line of the text list file would normally contain a 2 hex digit command of the sprite, a tab, and then a short description of the sprite in question (to include more information or longer descriptions, you should use a custom tooltip instead), and must be terminated with a line feed. However, the 2 digit command and tab can be omitted.

The sprite data file is in the exact same format as you'd find the sprite data in the ROM. For more specifics on the game's sprite data format, consult a format document. As with the ROM data, the first byte of the file is the sprite data header (reserved, should be zero for this), and the file should end with a 0xFF byte. Each sprite should be created as though it were going to appear on the first top subscreen of a level. Lunar Magic uses the “bottom subscreen” indicator bit to separate the sprites into groups that correspond with the list included in the text file. Turning this bit on tells the program that the current sprite belongs in the next group. This means you can have more than one sprite in one group… Lunar Magic will display the group of sprites together and can paste them into a level the same way it would with an individual sprite.

If you want to play with this feature a bit without manually editing files, Lunar Magic does have a basic built-in way of adding sprites to this list. Simply go to a horizontal level in LM, select the sprites you want and center them within the first top left subscreen of the level, and hit Ctrl-Shift-F12. To delete a sprite in the list, hit the delete key. To move a sprite in the list, select it and move it with the arrow keys while holding shift.

Custom Sprite List Sizes

Starting in version 1.80, Lunar Magic allows sprites to have a user-defined size for the number of bytes they take up in the sprite list for the level. These extra bytes can be set when adding a sprite manually.

Typically the sizes would be set by a 3rd party utility. To set them yourself, you must create and store a 0x400 byte table containing the sprite sizes somewhere inside the ROM (first 0x100 bytes are for sprites 00-FF that use an extra bit of 0, next 0x100 are for sprites 00-FF that use an extra bit of 1, etc). Place the SNES address for this table at 0x7750C PC. Then put 0x42 at 0x7750F to enable use of the table by Lunar Magic.

The max size value for a sprite is currently 0xF.

Warning: This will only enable the table for use by Lunar Magic. It's up to you or a 3rd party utility to modify the game code to take the new sizes into account.

Starting in version 3.11, 3rd party programs can also notify Lunar Magic that the sprite count limit per level has been increased to 255 to avoid the sprite count warning message appearing. To do this, clear the lowest bit of the byte at 0x801E0 PC.

```
macro spr(id, x, y, z, extra) db <id>|((<x>&1)<<7) db
(<x>>>1)|((<y>&7)<<5) db (<y>>>3)|(<z><<3) db
<extra> endmacro



;Note: for LM these are offsets instead sprite_data:

   dw .MainMap - sprite_data dw .YoshisIsland - sprite_data
dw .VanillaDome - sprite_data dw .ForestOfIllusion -
sprite_data dw .BowsersValley - sprite_data dw
.SpecialWorld - sprite_data dw .StarWorld - sprite_data

.MainMap:

   dw $0000



.YoshisIsland:

   %spr(!some_id, !x, !y, !z, !extra) dw $0000



.VanillaDome:

   dw $0000



.ForestOfIllusion: dw $0000



.BowsersValley:
   dw $0000


.SpecialWorld:

   dw $0000


.StarWorld:

   dw $0000
```

Warning: Lunar Magic only inserts the table. It's up to you or a 3rd party utility to modify the game code or apply a patch to make use of it. If you don't, it won't actively harm anything but no custom sprites displayed by Lunar Magic's overworld editor will show up in the game.

Custom Music Track Names

You can customize the text Lunar Magic uses for the music track names. To do this, create a UTF-8 text file with the same name as your ROM but with a .msc extension (ie. marioworld.msc) and put it in the same folder as your ROM file. The format of each line in the text file is: Track Number (in hex), TAB, Reserved Value (in hex), TAB, Track name text.

The reserved value should be zero for level music tracks, and 1 for overworld music tracks.

If you want to include a comment in the file, start the line with a semicolon ";" character.

Custom User Toolbar

Lunar Magic allows a second toolbar to be added to the main program's window, which appears just under the program's own toolbar. This second toolbar is user defined and can be used to launch external applications or activate internal Lunar Magic functions. It also allows for assigning keyboard shortcuts which can replace Lunar Magic's existing keyboard shortcuts.

External Application Warning: While Lunar Magic is compatible with other programs making changes to the ROM while it's still "open" in LM (see note below), the other programs may not be so forgiving. Some programs are written to ignore any changes made after loading the ROM file and/or completely overwrite the original file when saving. If you were to open such an application and load the ROM in it, save a level in Lunar Magic, then save in the other application, ***the application will effectively erase the level changes you just saved in Lunar Magic!*** (or potentially corrupt the level without warning, if the application doesn't write out the whole ROM but still uses the previously loaded ROM to search for free space). Whether or not this will happen depends entirely on the coding of the external application. If you aren't sure, either ask the application's author or don't perform save operations in LM until after the external application has been closed.

For External Application Authors: With the above in mind, the simplest way to avoid such issues with your program is by reloading the ROM during the start of your saving process, and by having the OS lock the ROM file for exclusive write access when saving.

Note: When a user "opens" a ROM in Lunar Magic, the ROM file is not actually kept open. The program just reopens the file for brief periods when it has to read or write to it, such as with loading or saving levels. When writing however the file is locked for exclusive write access.

User Toolbar File Format

The user toolbar is configured by placing a UTF-8 text file called "usertoolbar.txt" in the same folder as Lunar Magic's exe file. This file is examined whenever you launch Lunar Magic, and contains a list of commands detailing the buttons to set up.

The file defines the buttons in the same order that they will appear on the toolbar. A button definition is started by putting "***START***" on a line by itself, and ended with putting "***END***" on a line by itself. It is also legal to start a new button definition before ending the current one... in that case, the current button definition is ended automatically and the new one is started. For example:

```
;A simple usertoolbar.txt
;Note that semi-colon is used to mark comments

***START***
LM_SPACER       ;not really a button, just a separator
***START***
LM_DEFAULT      ;first button, does absolutely nothing
***START***
LM_VIEW_ADD_SPRITE      ;second button, opens "Add Sprites"
window
***START***
"notepad.exe";third button, opens notepad (no tab before ; )
***END***       ;we're done defining buttons for now
```

While each of the above buttons use only 1-line definitions, buttons can have up to 5 lines with each line defining specific properties for the button. If you don't want to specify anything for a particular line you can just put "LM_DEFAULT" to skip that line. Each line is used as follows:

Line 1: Internal LM function name *or* path and file to external program.
Line 2: Icon number to use for toolbar button image, comma, tooltip text for button.
Line 3: Options list (separated by commas).
Line 4: Keyboard shortcut directives (separated by commas).
Line 5: Current directory to start external program in.

So a full button definition could look something like this:

```
;A slightly less simple usertoolbar.txt

***START***
LM_SPACER              ;to line up with LM's main toolbar
***START***
"notepad.exe" "readme.txt"
0,Read the readme.txt file for Lunar Magic.
LM_CLOSE_ON_CLOSE      ;if LM closes, so do you!
'n',VK_CONTROL         ;Ctrl+N for keyboard shortcut
"%4"    ;start in LM directory (not the best way)
***END***
```

Button Definition Line Guide

Line 1: LM Function / External Application

On this line, you can either put a pre-defined LM function name, or the path and filename of an external application to launch. Alternatively you can put LM_SPACER, which does not actually create a button but just places a blank space separator in the toolbar which can be used for grouping buttons.

If you specify a path and file name, it's usually best to enclose it in quotes to avoid problems with spaces. There are also some predefined placeholders you can use that LM will fill in the values for. These are:

%1 path and file name of current ROM
%2 path of current ROM (includes trailing slash)
%3 file name of current ROM
%4 path of LM exe (includes trailing slash)
%5 file name of current ROM without file extension
%7 current level number open in LM, in hex
%8 current version of LM, in hex (so 2.40 would be 240)
%9 handle of window to issue requests to LM plus verification code, in the form XXXXXXXX:YYYY where X=32 bit handle of window and Y=16 bit verification code, in hex. See issuing requests to LM for more info.

Line 2: Icon # to use, comma, button tooltip text

For internal LM commands (or if LM_USEIMAGE_LIST is used in the options line), the image used for the button will be from the image list and this number will be relative to the current image base index (see Image List section for details). Otherwise, the image used will be an icon extracted directly from the external application's exe resources and the number will be relative to the order that the application's icons are stored in. In this case you'd normally use 0 as that would get you the first icon in the exe, which is the icon Windows Explorer would display for the program. The extracted icon will be added at the end of the main image list for the toolbar, which means it can also be used by other buttons in the toolbar. If an icon cannot be extracted from the external application, you will just get a generic icon.

The tooltip text you specify will appear over the button when the mouse hovers over it.

Note that having LM extract icons from multiple exe files for the toolbar means that it has to examine each one on startup. While this isn't likely to be much of an issue for today's computers, if you want to speed this up you may want to consider using a single image list file instead for all toolbar buttons.

Line 3: Options list (separated by commas)

This line can be used to specify multiple options, separated by commas. Check the Button Options List for what you can place here.

Line 4: Keyboard shortcut directives (separated by commas)

This line can be used to specify a keyboard shortcut for the button. This can be a single character enclosed in single quotes, one of the predefined keyboard definitions, or the actual virtual keycode for the key in hex. You can also optionally specify one or more modifier keys to allow for keyboard combinations that require Ctrl, Alt, or Shift.

Note that if you specify a keyboard shortcut that is already used by Lunar Magic itself, your keyboard shortcut will replace LM's shortcut. However if you specify a keyboard shortcut that you already assigned to another button, both will be triggered by the same shortcut.

Line 5: Current directory to start external program in

You can put a path here for which directory you want your external application to start in. By default this is set to the external program's own directory, unless you changed it with an option on line 3. You can also use the placeholders listed in the description for line 1.

Global Options

You can place certain directives outside of button ***START/END*** definitions to control things that affect more than one button. Check the Global Options List for what you can place here.

Image List

Lunar Magic builds a list of images to use for the buttons in the custom toolbar. You can add multiple bitmap image files to the list, and LM also adds any icons extracted from external applications to it.

Bitmap files can contain multiple button images in a horizontal stripe, and will be split up depending on the button image height that has been set and the width of the bitmap. If it is the first image to be added, the height of the bitmap will be used to set both the height and width of the button images used for the toolbar. The first image in the stripe is used to determine the transparent color used in the bitmap, so it should be left blank.

The list is built in the order that the file is parsed, so indexes are assigned sequentially. For example, suppose we had a bitmap with a height of 16 and a width of 48 and used it like this:

```
;playing with usertoolbar.txt image list

LM_ADDIMAGE "ourbitmap.bmp"

***START***
LM_SPACER       ;to line up with LM's main toolbar
***START***     ;first button, notepad
"notepad.exe"
0,To write text.
```

```
***START***
LM_VIEW_ADD_SPRITE      ;second button, opens "Add Sprites"
window
1,To add sprites.
***START***
LM_VIEW_ADD_OBJECT      ;third button, opens "Add Objects"
window
2,To add objects.
***START***     ;fourth button, mspaint
"mspaint.exe"
0,To draw pictures.
***END***       ;we're done defining buttons for now
```

In the above example, the image list will contain 5 images: index 0 is the first 16x16 image from ourbitmap.bmp (normally left blank), index 1 is the second 16x16 image from ourbitmap.bmp, index 2 is the third 16x16 image from ourbitmap.bmp, index 3 is the notepad.exe icon, and index 4 is the mspaint.exe icon.

When specifying an external application, the icon index beside the tooltip is relative to the application exe's own internal icon resources (unless you specify LM_USEIMAGE_LIST on the options line to force it not to extract an icon).

It's also possible to modify the base index used for images depending on other commands you can use. For example:

```
;playing with usertoolbar.txt base index

LM_ADDIMAGE "ourbitmap.bmp"

***START***
LM_SPACER       ;to line up with LM's main toolbar
***START***
LM_VIEW_ADD_SPRITE      ;first button uses image 1 of
ourbitmap.bmp
1,To add sprites.
***END***
```

```
LM_NEWIMAGE "newbitmap.bmp"

***START***
LM_VIEW_ADD_SPRITE      ;second button uses image 1 of
newbitmap.bmp
1,To add sprites.
***END***

LM_IMAGEBASE_PREVIOUS

***START***
LM_VIEW_ADD_SPRITE      ;third button uses image 1 of
ourbitmap.bmp
1,To add sprites.
***END***
```

Minimum Height and Scaling: The minimum image height is normally 16. Note however that if the minimum height when adjusted for your screen's DPI is more than 25% larger than the image you provide, scaling will occur.

"LM_FILE_OPEN_ROM",

"LM_FILE_OPEN_FILE",

"LM_FILE_OPEN_LEVEL",

"LM_FILE_OPEN_LEVEL_ADDRESS", "LM_FILE_NEXT_LEVEL",

"LM_FILE_PREVIOUS_LEVEL",
"LM_FILE_SAVE_LEVEL_TO_ROM_AS",
"LM_FILE_SAVE_DIRECTORY", "LM_FILE_SAVE_FILE",

"LM_FILE_SAVE_FILE_AS", "LM_FILE_SCAN_ROM",

"LM_FILE_TILE_EDITOR_SETTINGS",
"LM_FILE_EXTRACT_GFX",

"LM_FILE_INSERT_GFX",

"LM_FILE_EXTRACT_EXGFX", "LM_FILE_INSERT_EXGFX",
"LM_FILE_EXTRACT_EXGFX_LIST",
"LM_FILE_INSERT_EXGFX_LIST",
"LM_FILE_EXTRACT_PALETTE", //shared palettes
"LM_FILE_INSERT_PALETTE", "LM_FILE_EXPORT_PALETTE",
//custom palettes "LM_FILE_IMPORT_PALETTE",
"LM_FILE_EXIT",

"LM_FILE_EXPORT_DIRECTORY",
"LM_FILE_ENCRYPT_LEVELS",
"LM_FILE_CLEAR_OLD_LEVELS", "LM_FILE_EMULATOR_RUN",
"LM_FILE_EMULATOR_SETTINGS", "LM_FILE_EXPAND_ROM2",

"LM_FILE_EXPAND_ROM3",

"LM_FILE_EXPAND_ROM4",

"LM_FILE_EXPAND_ROM8", //ExLoROM

"LM_FILE_EXPAND_ROM6_SA1",
"LM_FILE_EXPAND_ROM8_SA1",
"LM_FILE_CREATE_RESTORE", "LM_FILE_RESTORE",

"LM_FILE_CREATE_IPS",

"LM_FILE_APPLY_IPS",

"LM_FILE_EXPORT_DIRECTORY_BITMAP",
"LM_FILE_EXPORT_BITMAP", "LM_FILE_RELOAD_ROM",

<!-- "LM_FILE_CLOSE_ROM", -->
"LM_FILE_ANALYZE_LEVELS",

"LM_FILE_SAVE_BUTTON", //button version commands
"LM_FILE_EXTRACT_GFX_BUTTON",
"LM_FILE_INSERT_GFX_BUTTON",
"LM_FILE_EXTRACT_EXGFX_BUTTON",
"LM_FILE_INSERT_EXGFX_BUTTON",
"LM_FILE_INSERT_ALL_GRAPHICS",

"LM_FILE_INT_EMULATOR_RUN",
"LM_FILE_INT_EMULATOR_PAUSE",
"LM_FILE_INT_EMULATOR_MUTE",
"LM_FILE_INT_EMULATOR_USE_F4",
"LM_FILE_INT_EMULATOR_STOP_LEVEL_CHANGE", <!--
"LM_FILE_INT_EMULATOR_UNLOAD",
"LM_FILE_INT_EMULATOR_TILES",
"LM_FILE_INT_EMULATOR_FRAME_ADVANCE",
"LM_FILE_INT_EMULATOR_PAUSE_TRANSLUCENT", -->

"LM_FILE_RECENT_MENU",

"LM_EDIT_CUT",

"LM_EDIT_COPY",

"LM_EDIT_PASTE",

"LM_EDIT_EDIT_LAYER_1", "LM_EDIT_EDIT_LAYER_2",
"LM_EDIT_SPRITES",

"LM_EDIT_INSERT",

"LM_EDIT_DELETE",

"LM_EDIT_DELETE_ALL",

"LM_EDIT_SELECT_ALL",

"LM_EDIT_BRING_TO_FRONT", "LM_EDIT_SEND_TO_BACK",
"LM_EDIT_BRING_FORWARD", "LM_EDIT_SEND_BACKWARD",
"LM_EDIT_ZORDER_UP",

"LM_EDIT_ZORDER_DOWN",

"LM_EDIT_INCREASE_X",

"LM_EDIT_DECREASE_X",

"LM_EDIT_INCREASE_Y",

"LM_EDIT_DECREASE_Y",

"LM_EDIT_UNDO",

"LM_EDIT_REDO",

"LM_EDIT_CDM16",

"LM_EDIT_REMAP_DM16",

<!-- "LM_EDIT_PROPERTIES", "LM_EDIT_ESCAPE", //new Map16 editor only "LM_EDIT_SELECT_FG", //new Map16 editor only "LM_EDIT_SELECT_BG", //new Map16 editor only "LM_EDIT_SELECT_ALL", //new Map16 editor only --> "LM_EDIT_EDIT_MANUAL",

"LM_VIEW_LAYER_1",

"LM_VIEW_LAYER_2", <!-- "LM_VIEW_BACKGROUND", --> "LM_VIEW_LAYER_3",

"LM_VIEW_SPRITES",

"LM_VIEW_SPRITE_DATA",

"LM_VIEW_SCREEN_EXITS", "LM_VIEW_SCREEN_GRID",

"LM_VIEW_SCREEN_GRID_2", "LM_VIEW_SURFACE_OUTLINE", "LM_VIEW_LINE_GUIDE_OUTLINE", "LM_VIEW_ALL_ENTRANCES",<!-- depreciated "LM_VIEW_LEVEL_ENTRANCE", "LM_VIEW_LEVEL_ENTRANCE_2", "LM_VIEW_MIDWAY_POINT", --> "LM_VIEW_BLOCK_EXITS",

"LM_VIEW_BLOCK_CONTENTS", "LM_VIEW_GREEN_SWITCH", "LM_VIEW_YELLOW_SWITCH", "LM_VIEW_BLUE_SWITCH",

"LM_VIEW_RED_SWITCH",

"LM_VIEW_SPECIAL_WORLD", "LM_VIEW_INVISIBLE",

"LM_VIEW_INVISIBLE_2",

"LM_VIEW_LINE_ON",

"LM_VIEW_INCREASE_FRAME", "LM_VIEW_ANIMATION", "LM_VIEW_SILVER_POW",

"LM_VIEW_POW",

"LM_VIEW_512HEIGHT_BG", "LM_VIEW_TILE_GRID",

"LM_VIEW_CDM16",

"LM_VIEW_ZOOM",

"LM_VIEW_ZOOM_TOGGLE",

"LM_VIEW_ZOOM_DEFAULT", "LM_VIEW_ZOOM_PLUS",

"LM_VIEW_ZOOM_MINUS",

"LM_VIEW_ZOOM_FILTER",

"LM_VIEW_HAVE_STAR",

"LM_VIEW_TIME_100",

"LM_VIEW_5YOSHI_COINS", "LM_VIEW_RESET_ANIMATION",

"LM_VIEW_ADD_OBJECT", //editors "LM_VIEW_ADD_SPRITE",

"LM_VIEW_8x8", <!-- "LM_VIEW_16x16_OLD", //don't include in published list --> "LM_VIEW_16x16",

"LM_VIEW_BACK",

"LM_VIEW_OVERWORLD",

"LM_VIEW_PALETTES",

"LM_VIEW_LAYER_3_EDITOR",

"LM_LEVEL_SUPER_BYPASS", "LM_LEVEL_SUPER_BYPASS2", "LM_LEVEL_BYPASS_FG", //old "LM_LEVEL_BYPASS_SP", //old

"LM_LEVEL_BYPASS_MUSIC", "LM_LEVEL_EXITS",

"LM_LEVEL_ENTRANCE",

"LM_LEVEL_ENTRANCE2",

"LM_LEVEL_SCAN_EXITS",

<!-- depreciated, moved to BG editor "LM_LEVEL_BG",

-->

"LM_LEVEL_GRAPHICS",

"LM_LEVEL_PROPERTIES",

"LM_LEVEL_OTHER",

"LM_LEVEL_SPRITES",

"LM_LEVEL_EXTEND_ANI", //old <!-- depreciated, moved to BG editor "LM_LEVEL_BG_MAP16",

"LM_LEVEL_BG_OFFSET",

-->

"LM_LEVEL_EX20_LEVEL",

"LM_LEVEL_EX20_GLOBAL", "LM_LEVEL_EX20_SETTINGS", "LM_LEVEL_LAYER3_BYPASS", "LM_LEVEL_LAYER3_BYPASS2", "LM_LEVEL_LAYER3_SETTINGS",

"LM_OPTIONS_GENERAL",

"LM_OPTIONS_RESTORE",

"LM_OPTIONS_ANIM_RATE", "LM_OPTIONS_COMPRESSION", "LM_OPTIONS_VRAM",

<!-- we may move most of these into a dialog later, so don't publish "LM_OPTIONS_ALLOW_FRAGMENT", "LM_OPTIONS_MAINTAIN_CHECKSUM", "LM_OPTIONS_AUTO_SCREENS", "LM_OPTIONS_ATTACH_FILES", "LM_OPTIONS_SPRITE_OBJECT_ID", "LM_OPTIONS_BG_CURSOR", "LM_OPTIONS_TRANSLUCENT", "LM_OPTIONS_WINDOW_SIZE", "LM_OPTIONS_SCAN_EXITS", "LM_OPTIONS_SCAN_SPRITES", "LM_OPTIONS_INSTALL_VRAM", "LM_OPTIONS_CUSTOM_SPRTES", "LM_OPTIONS_AUTO_HEADER", "LM_OPTIONS_SAVE_PROMPT", "LM_OPTIONS_MOUSE_GESTURES", "LM_OPTIONS_SAVE_GESTURES", "LM_OPTIONS_ANIM_RATE", "LM_OPTIONS_USE_FASTROM", "LM_OPTIONS_PATCH_FASTROM", "LM_OPTIONS_RESTORE",

"LM_OPTIONS_WARN_IPS",

"LM_OPTIONS_CONVERT_BERRY", "LM_OPTIONS_COMPRESSION", "LM_OPTIONS_OTHER_BYPASS", "LM_OPTIONS_PAST_2MB",

"LM_OPTIONS_WARN_OBJECT", "LM_OPTIONS_CORRECT_FATAL_ERRORS", -->

"LM_HELP_CONTENTS",

"LM_HELP_ABOUT",

```
"LM_VIEW_CUSTOM_TRIGGER_0",
"LM_VIEW_CUSTOM_TRIGGER_1",
"LM_VIEW_CUSTOM_TRIGGER_2",
"LM_VIEW_CUSTOM_TRIGGER_3",
"LM_VIEW_CUSTOM_TRIGGER_4",
"LM_VIEW_CUSTOM_TRIGGER_5",
"LM_VIEW_CUSTOM_TRIGGER_6",
"LM_VIEW_CUSTOM_TRIGGER_7",
"LM_VIEW_CUSTOM_TRIGGER_8",
"LM_VIEW_CUSTOM_TRIGGER_9",
"LM_VIEW_CUSTOM_TRIGGER_A",
"LM_VIEW_CUSTOM_TRIGGER_B",
"LM_VIEW_CUSTOM_TRIGGER_C",
"LM_VIEW_CUSTOM_TRIGGER_D",
"LM_VIEW_CUSTOM_TRIGGER_E",
"LM_VIEW_CUSTOM_TRIGGER_F",
"LM_VIEW_ONESHOT_TRIGGER_00",
"LM_VIEW_ONESHOT_TRIGGER_01",
"LM_VIEW_ONESHOT_TRIGGER_02",
"LM_VIEW_ONESHOT_TRIGGER_03",
"LM_VIEW_ONESHOT_TRIGGER_04",
"LM_VIEW_ONESHOT_TRIGGER_05",
"LM_VIEW_ONESHOT_TRIGGER_06",
"LM_VIEW_ONESHOT_TRIGGER_07",
"LM_VIEW_ONESHOT_TRIGGER_08",
"LM_VIEW_ONESHOT_TRIGGER_09",
"LM_VIEW_ONESHOT_TRIGGER_0A",
"LM_VIEW_ONESHOT_TRIGGER_0B",
"LM_VIEW_ONESHOT_TRIGGER_0C",
"LM_VIEW_ONESHOT_TRIGGER_0D",
"LM_VIEW_ONESHOT_TRIGGER_0E",
"LM_VIEW_ONESHOT_TRIGGER_0F",
"LM_VIEW_ONESHOT_TRIGGER_10",
"LM_VIEW_ONESHOT_TRIGGER_11",
"LM_VIEW_ONESHOT_TRIGGER_12",
"LM_VIEW_ONESHOT_TRIGGER_13",
"LM_VIEW_ONESHOT_TRIGGER_14",
"LM_VIEW_ONESHOT_TRIGGER_15",
"LM_VIEW_ONESHOT_TRIGGER_16",
"LM_VIEW_ONESHOT_TRIGGER_17",
"LM_VIEW_ONESHOT_TRIGGER_18",
"LM_VIEW_ONESHOT_TRIGGER_19",
"LM_VIEW_ONESHOT_TRIGGER_1A",
"LM_VIEW_ONESHOT_TRIGGER_1B",
"LM_VIEW_ONESHOT_TRIGGER_1C",
"LM_VIEW_ONESHOT_TRIGGER_1D",
"LM_VIEW_ONESHOT_TRIGGER_1E",
"LM_VIEW_ONESHOT_TRIGGER_1F",
"LM_VIEW_MANUAL_TRIGGER_INC_0",
"LM_VIEW_MANUAL_TRIGGER_INC_1",
"LM_VIEW_MANUAL_TRIGGER_INC_2",
"LM_VIEW_MANUAL_TRIGGER_INC_3",
"LM_VIEW_MANUAL_TRIGGER_INC_4",
"LM_VIEW_MANUAL_TRIGGER_INC_5",
"LM_VIEW_MANUAL_TRIGGER_INC_6",
"LM_VIEW_MANUAL_TRIGGER_INC_7",
"LM_VIEW_MANUAL_TRIGGER_INC_8",
"LM_VIEW_MANUAL_TRIGGER_INC_9",
"LM_VIEW_MANUAL_TRIGGER_INC_A",
"LM_VIEW_MANUAL_TRIGGER_INC_B",
"LM_VIEW_MANUAL_TRIGGER_INC_C",
"LM_VIEW_MANUAL_TRIGGER_INC_D",
"LM_VIEW_MANUAL_TRIGGER_INC_E",
"LM_VIEW_MANUAL_TRIGGER_INC_F",
"LM_VIEW_MANUAL_TRIGGER_DEC_0",
"LM_VIEW_MANUAL_TRIGGER_DEC_1",
"LM_VIEW_MANUAL_TRIGGER_DEC_2",
"LM_VIEW_MANUAL_TRIGGER_DEC_3",
"LM_VIEW_MANUAL_TRIGGER_DEC_4",
"LM_VIEW_MANUAL_TRIGGER_DEC_5",
"LM_VIEW_MANUAL_TRIGGER_DEC_6",
"LM_VIEW_MANUAL_TRIGGER_DEC_7",
```

"LM_VIEW_MANUAL_TRIGGER_DEC_8",
"LM_VIEW_MANUAL_TRIGGER_DEC_9",
"LM_VIEW_MANUAL_TRIGGER_DEC_A",
"LM_VIEW_MANUAL_TRIGGER_DEC_B",
"LM_VIEW_MANUAL_TRIGGER_DEC_C",
"LM_VIEW_MANUAL_TRIGGER_DEC_D",
"LM_VIEW_MANUAL_TRIGGER_DEC_E",
"LM_VIEW_MANUAL_TRIGGER_DEC_F",
"LM_VIEW_CUSTOM_TRIGGER_INC",
"LM_VIEW_CUSTOM_TRIGGER_DEC",
"LM_VIEW_ONESHOT_TRIGGER_INC",
"LM_VIEW_ONESHOT_TRIGGER_DEC",
"LM_VIEW_MANUAL_TRIGGER_INC",
"LM_VIEW_MANUAL_TRIGGER_DEC",
"LM_VIEW_CUSTOM_TRIGGER_CURRENT",
"LM_VIEW_ONESHOT_TRIGGER_CURRENT",
"LM_VIEW_MANUAL_TRIGGER_CURRENT_INC",
"LM_VIEW_MANUAL_TRIGGER_CURRENT_DEC",

"LM_KEY_SPRITE19_FIX",

"LM_KEY_MENUCOLOR_FIX", "LM_KEY_GRID_COLOR",

"LM_KEY_ADD_CSPRITE",

"LM_KEY_TRUNCATE", <!--

"LM_KEY_2BPP_MODE",

"LM_KEY_DUMP_DATA", --> "LM_KEY_MARIO_REGIONS",
"LM_KEY_BGEDIT_ONTOP", <!--

"LM_KEY_LAYER3_16X16_MODE", -->
"LM_KEY_EXANIM_SLOTS",

"LM_KEY_GFX_OVERRIDE",

"LM_MOUSE_LEVEL_BACK",

"LM_MOUSE_LEVEL_FORWARD", "LM_MOUSE_SCREEN_EXIT",
"LM_MOUSE_EDIT_SCREEN_EXIT",

```
"VK_CONTROL",

"VK_SHIFT",

"VK_ALT",

<!-- "VK_EXTENDED",

"VK_NOT_EXTENDED",-->

"VK_LSHIFT",

"VK_RSHIFT",

"VK_LCONTROL",

"VK_RCONTROL",

"VK_LALT",

"VK_RALT",

<!-- "VK_SCANCODE",

"VK_LWIN",

"VK_RWIN",

"VK_APPS",

"VK_ALT_GR",

-->

"VK_NUMPAD_ENTER",

"VK_NUMPAD0",
"VK_NUMPAD1",
"VK_NUMPAD2",
"VK_NUMPAD3",
"VK_NUMPAD4",
"VK_NUMPAD5",
"VK_NUMPAD6",
"VK_NUMPAD7",
"VK_NUMPAD8",
"VK_NUMPAD9",
"VK_NUMPAD_MULTIPLY",
"VK_NUMPAD_ADD",
"VK_NUMPAD_SEPARATOR",
"VK_NUMPAD_SUBTRACT",
"VK_NUMPAD_DECIMAL",
"VK_NUMPAD_DIVIDE",
"VK_F1",
"VK_F2",
"VK_F3",
"VK_F4",
"VK_F5",
"VK_F6",

"VK_F7",

"VK_F8",

"VK_F9",

"VK_F10",

"VK_F11",

"VK_F12",

"VK_F13",

"VK_F14",

"VK_F15",

"VK_F16",

"VK_F17",

"VK_F18",

"VK_F19",

"VK_F20",

"VK_F21",

"VK_F22",

"VK_F23",

"VK_F24",

"VK_INSERT",
"VK_DELETE",

"VK_HOME",

"VK_END",

"VK_PAGEUP",

"VK_PAGEDOWN",

"VK_ESCAPE",

"VK_TAB",

"VK_BACK",

"VK_RETURN",

"VK_UP",

"VK_DOWN",

"VK_LEFT",

"VK_RIGHT",

"VK_SPACE",

"VK_PAUSE",

"VK_MBUTTON", //Middle Mouse

"VK_XBUTTON1", //Mouse 4

"VK_XBUTTON2", //Mouse 5
```

Lunar Magic Button Options List

This is a list of options that can be specified on the options line for buttons. You can list more than one by separating them with commas.

LM_DEFAULT	Does nothing. Effectively just a placeholder you can use on any line if you want to be able to skip that line without specifying anything for it.
LM_USEIMAGE_LIST	Use an image from the image list instead of extracting an icon from the external app's exe to use as the toolbar image.
LM_NO_BUTTON	No button will be displayed on the toolbar for the current item. This can be useful if you only want to assign a keyboard shortcut to something.
LM_NO_CONSOLE_WINDOW	No console window will be shown for console programs. This can be useful for running .bat files without showing a console window. Do not use this option with console programs or scripts that may need user interaction or they'll be left hanging. This option is ignored if used with LM_OPEN_OTHER. Only effective on Windows 2000/XP and later.
LM_UPDATE_MENUKEY	For internal LM commands, this will look up the corresponding menu item in Lunar Magic and change the keyboard shortcut text listed to whatever the keyboard shortcut is for the current button. If Lunar Magic doesn't have text to

	describe the shortcut, it will blank any existing keyboard shortcut text for that menu item.
LM_ALLOW_MULT_INSTANCES	Allows multiple instances of an external program to run. Normally LM only launches one copy of a program per button, and when you click on the button again it will switch the window focus to the already open program. Otherwise it will launch another copy of the program.
LM_OPEN_OTHER	Can be used to open things other than programs, such as a folder in explorer, a website, a document, etc. Uses ShellExecute. For a folder it's best to have a trailing slash at the end of the path to avoid potential confusion for the OS in cases where there's a file with the same name as a folder. Note: LM does not keep track of items opened with this method, therefore other options like LM_CLOSE_ON_CLOSE will not apply to it.
LM_DIR_PROGRAM	For external applications, this will cause it to set the current directory to the external program's folder. This is the default, so it does not need to be specified.
LM_DIR_ROM	For external applications, this will cause it to set the current directory to the currently open ROM's folder.
LM_DIR_LM	For external applications, this will cause it to set the current directory to Lunar Magic's folder.
LM_CLOSE_ON_CLOSE	When Lunar Magic is closed, it will send a WM_CLOSE message to all top-level WS_VISIBLE windows that were opened by programs from this button. This may also be sent if Lunar Magic unloads a ROM without immediately loading a new ROM (ROM lock operation).
LM_CLOSE_ON_NEW_ROM	When Lunar Magic opens a new ROM, it will send a WM_CLOSE message to all top-level WS_VISIBLE windows that were opened by programs from this button.
LM_AUTORUN_ON_NEW_ROM	When Lunar Magic opens a new ROM, it will automatically start the program associated with this button. Only applies to buttons that start external applications. **Note:** If combined with LM_CLOSE_ON_NEW_ROM and multiple instances aren't allowed, LM will not wait on an already running instance to close and will actually allow a second instance to start, however the first instance will no longer receive any messages

	from LM if it chooses to stay open. **Note 2:** Windows from newly launched programs will normally appear over LM, so don't use this if you find that annoying.
LM_NOTIFY_ON_NEW_ROM	When Lunar Magic loads a new ROM, it will send a custom message to the external application using the Win32 API PostMessage() function to notify it of this. The message ID will be 0xBECA (WM_APP+3ECA). The wParam value will be a handle to a window (hwnd), which you can use GetWindowText() on to get the filename and path of the new ROM to open. The upper 16 bits of the lParam value (lParam>>16) has a confirmation code that you should use to confirm that the message is from LM (should be equal to 0x6942... if it isn't, discard the message). The lower 6 bits (lParam&0x3F) contains the notification type (0=new ROM), and the next 10 bits ((lParam>>6)&0x3FF) contains the notification variable (for a new ROM, this specifies the ROM identification. 0=Mario World US, 1=Mario World Japanese, 2=Mario All Stars + World, 3=Mario Advance 2).
LM_NOTIFY_ON_NEW_LEVEL	See LM_NOTIFY_ON_NEW_ROM. The only difference is with the notification type (1=open new level), and the notification variable contains the new level number. The wParam value should be ignored.
LM_NOTIFY_ON_SAVE_LEVEL	See LM_NOTIFY_ON_NEW_ROM. The only difference is with the notification type (3=save level to ROM), and the notification variable contains the level number. The wParam value should be ignored. See the Save Notifications Warning.
LM_NOTIFY_ON_SAVE_MAP16	See LM_NOTIFY_ON_NEW_ROM. The only difference is with the notification type (4=save Map16 to ROM), and the notification variable contains the level number. The wParam value should be ignored. See the Save Notifications Warning.
LM_NOTIFY_ON_SAVE_OV	See LM_NOTIFY_ON_NEW_ROM. The only difference is with the notification type (5=save overworld to ROM), and the notification variable should be ignored. The wParam value should be ignored. See the Save Notifications Warning.
LM_NOTIFY_ON_CLOSE	See LM_NOTIFY_ON_NEW_ROM. The only

difference is with the notification type (2=close Lunar Magic), and the notification variable should be ignored. The wParam value should be ignored. This may also be sent if Lunar Magic unloads a ROM without immediately loading a new ROM (ROM lock operation).

Save Notifications Warning: External applications should not automatically modify the ROM in response to getting a save type notification from LM. This is because if 2 or more applications try to save to the ROM at the same time in response to a save notification, they could clash and potentially either corrupt the ROM or be unable to open the ROM. There is also the potential to clash with LM itself if LM is doing a multiple level import operation. Not to mention that if an external program modifies the ROM every time LM saves something, it will trigger creating a full restore point for every save, which will bloat the restore file. These save notifications are mainly just meant for monitoring purposes only.

Lunar Magic Global Options List

This is a list of options that can be specified outside of button definitions.

LM_DISPLAY_ERRORS [max errors in decimal]	This will cause LM to report on syntax errors while parsing the file (default is off). You can optionally specify the maximum number of errors to report (default is 3).
LM_NO_TOOLBAR	Causes the user toolbar to not be shown. However keyboard shortcuts will still be active.
LM_ADDIMAGE "path and file name"	Adds a bitmap specified by the path and file name to the image list. The bitmap can contain multiple button images in a horizontal stripe, and will be split up depending on the button image height that has been set and the width of the bitmap. If this is the first image to be added, the height of the bitmap will be used to set both the height and width of the button images used for the toolbar. The first image in the stripe is used to determine the transparent color used in the bitmap, so it should be left blank. The images are added at the end of the list, and the base index is not changed.
LM_NEWIMAGE "path and file name"	Does the same thing as LM_ADDIMAGE, except it resets the

	base index to start relative to the new image. This means buttons immediately following it can access the new images starting at 0. This can be useful for defining a new set of buttons without having to worry about how many images have already been added to the list and figuring out which image index to use. Note that images from earlier in the list can still be accessed counting backwards from 0, or by using one of the reset base index commands.
LM_IMAGEBASE_PREVIOUS	This resets the base index for images back to whatever it was before the last base index change.
LM_IMAGEBASE_GLOBAL	This resets the base index back to the actual start of the list, so 0 will be the first image in the full list.
LM_SETIMAGE_SIZE (image height in decimal)	Sets the image height (and image width) to use for the toolbar buttons. This will have no effect if an image/icon has already been added, or if a bitmap image is added before any icon. This should only be used if the first thing you're adding to the image list is an icon instead of a bitmap file and you don't want to use the default size of 16.
LM_USEIMAGE_FORCE [starting index in decimal]	This forces all buttons that would have extracted an icon from an external app's exe file to instead use an image from the global image list. The image indexes used will be assigned sequentially in order, and you can optionally specify the starting index to use (default is 1).
LM_USEIMAGE_FORCE_ALL [starting index in decimal]	This is like LM_USEIMAGE_FORCE, but it also overrides buttons that were already set to use images from the image list to instead use a sequentially assigned image index.
LM_ALLOW_MULT_INSTANCES_FORCE_ALL	This acts the same way as the LM_ALLOW_MULT_INSTANCES option

	for an individual button, but is applied to all buttons.
LM_CLOSE_ON_CLOSE_FORCE_ALL	This acts the same way as the LM_CLOSE_ON_CLOSE option for an individual button, but is applied to all buttons.
LM_CLOSE_ON_NEW_ROM_FORCE_ALL	This acts the same way as the LM_CLOSE_ON_NEW_ROM option for an individual button, but is applied to all buttons.
LM_NO_AUTORUN	This causes the LM_AUTORUN_ON_NEW_ROM option to be ignored for all buttons.
LM_NOTIFY_ON_NEW_ROM_FORCE_ALL	This acts the same way as the LM_NOTIFY_ON_NEW_ROM option for an individual button, but is applied to all buttons. Warning: It is considered bad behavior to send such window notification messages to a program that is not expecting it. While it's unlikely to do anything, it's best not to use this unless all your programs are set up to receive it.
LM_NOTIFY_ON_NEW_LEVEL_FORCE_ALL	This acts the same way as the LM_NOTIFY_ON_NEW_LEVEL option for an individual button, but is applied to all buttons. Also see above warning.
LM_NOTIFY_ON_CLOSE_FORCE_ALL	This acts the same way as the LM_NOTIFY_ON_CLOSE option for an individual button, but is applied to all buttons. Also see above warning.

Lunar Magic Send Requests

If you use %9 as a command line argument for a program launched from LM, this will provide the launched program with a handle to a window that you can use to issue requests to LM using the Win32 API PostMessage() function. The format of the info provided by %9 is XXXXXXXX:YYYY where X=32 bit handle of window and Y=16 bit verification code, in hex.

The message ID will be 0xBECB (WM_APP+3ECB). The wParam value should be 0 unless otherwise stated. The upper 16 bits of the lParam value (lParam>>16) should be the verification code that LM provided, or the message will be ignored. The lower 6 bits (lParam&0x3F) contains the request type (2=Reload current ROM, 3=Reload current level), and the next 10 bits ((lParam>>6)&0x3FF) contains the request variable (not currently used).

Note that LM is compatible with other programs making changes to the ROM while it's still "open" in LM, so it does not normally need to be told to reload the ROM. Since reloading can be disruptive to the user (who will be given the chance to save current data and may chose to cancel the reload request), you should therefore not send a reload request unless you have changed some kind of resource or data displayed by LM that you believe should be updated immediately.

For 64 bit Applications: The 32 bit window handle will still be valid and usable.

FAQ: Is Lunar Magic backwards compatible with prior versions?

Yes. Lunar Magic is backwards compatible with both MWL files and hacks created with older versions of the program, all the way back to version 1.0.

However this does not apply to 3rd party ASM hacks that you or other tools have inserted. Many of these have been known to be incompatible with newer versions of Lunar Magic. Of particular note are ones created prior to 1.70, as this version of LM included a number of significant ASM changes compared to 1.6x (which had been in use for approximately 6 years). Turning off the VRAM patch in Lunar Magic's options menu before doing any saving may help with some of these older hacks, but it means giving up the extra 2 GFX slots.

Hacks created prior to 2.30 that used a third party patch for layer 3 GFX uploading may also encounter problems, as the patch for doing so modified one of LM's ASM hacks which has since been updated to handle Layer 3 GFX uploading on its own.

Hacks created before 3.00 may have a few incompatibilities with the Dynamic Levels patch added in that version. One common issue is being unable to interact with the goal tape sprite. That can be resolved by updating whichever Sprite Tool program was used on the hack to the latest version. Alternatively you can turn off the VRAM patch option before doing any saving to prevent installing the Dynamic Levels patch, but you'll lose the ability to use anything other than the default level height in horizontal levels.

Note that Lunar Magic is only backwards compatible, not forwards compatible. **DO NOT** attempt to use older versions of the program on a ROM that a newer version was used on. This may result in corrupting your data. While you can often get away with it if the versions are within the same .0x series, they still aren't guaranteed to be compatible with each other.

To check which version of Lunar Magic was last used to modify a ROM, you can use the Scan ROM function.

FAQ: How do I edit the graphics?

The graphics in SMW are compressed. So right from the start, you have to use the Extract GFX from ROM menu command to export the graphics into .bin files (if you want all the graphics to be exported into a single file instead of multiple ones, change the Use Joined GFX Files option).

Once you've exported the graphics, you can either use the built-in 8x8 Tile Editor in Lunar Magic for altering the graphics loaded by a particular level, or you can just use any SNES tile editor of your choice on the .bin file(s).

When you're finished editing the graphics, use the Insert GFX to ROM menu command to insert them back into the game.

Lunar Magic can also insert "Extra Graphics" (ExGFX) into the ROM that SMW did not originally have. For more information on that, look up the Insert ExGFX to ROM menu and the Extract ExGFX from ROM menu.

FAQ: Why is my level filled with water tiles?

YY-CHR (a common tile editor) has a tendency to try and save new files as 8KB while the allowed size for most SMW GFX/ExGFX files is 4KB (or 2KB for layer 3 GFX). Telling SMW to decompress a file that's too large can result in the data overwriting part of the level tilemap in SNES RAM with mostly 0's, which happens to be the value for the water tile.

Make sure that your GFX/ExGFX files are the right size for the slots you're trying to use them in.

FAQ: How do I control where pipes and doors go?

In SMW, the destination of "exit-enabled" pipes and doors are determined by the "screen" number that they're on, and the exit number set for that screen. If you hit F1, Lunar Magic will split up the level into screens and will display the exit destination of each one at the top.

To alter the destination level for a screen, check the Modify Screen Exits menu command.

FAQ: How do I add one coin/note/whatever block instead of four?

Lunar Magic can change the x/y size of most objects (or for blocks, the number of blocks) that appear simply by selecting the object, holding down shift and using the arrow keys. You can also resize them using the mouse. So although the blocks in the Add Objects Window are often displayed in groups of four, you can easily change how many blocks actually appear once you paste the object into the level.

FAQ: Why are some sprites glitched in SMW and Lunar Magic?

The SNES does not have enough VRAM (Video RAM) space to allow all the graphics of the game to be displayed at will. So the game uses tilesets, for both the scenery and the sprites, to determine which graphics to load for a particular level. If you don't have the correct graphics loaded for a sprite you use, it will be displayed using the tiles of another sprite.

To change the tilesets, check the GFX Index in Header dialog, or if you've already bypassed it, the Super GFX Bypass dialog.

FAQ: Why are my sprites glitched in SMW but not in Lunar Magic?

Some sprites in SMW simply aren't compatible with one another. For example, Yoshi and a baby Yoshi should never be in the same level at the same time or Yoshi's head will look like a baby Yoshi. If you do run into this problem, try isolating and removing the offending sprite from the level.

You may also want to try adjusting the memory setting in the Change Properties in Sprite Header dialog to match the setting of a level that has that particular sprite.

FAQ: Why do I get a "Could not open ROM for Writing" error message?

Your ROM file is most likely marked as "Read-Only". In windows, right-click the ROM file and select properties, uncheck the "Read-Only" box, then hit ok.

FAQ: Where's the rest of Chocolate Island 2 (level 24)?

This level is a bit unusual, as the pipe destinations can take you to different levels depending on the number of coins collected and the current time limit. The extra levels are not accessible from within the standard 0x200 level pointers. However, the actual address locations in the ROM of these extra levels are known. Check the list given in the Open Level Address description. Modifying these levels will require saving them to an unused level number in the standard 0x200 levels, and then manually repairing the necessary pointer with a hex editor.

Alternatively, it may be easier to just turn this behavior off. You can find the option for this in the Overworld, Extra Options dialog.

FAQ: Why can't I edit Boss Monster rooms?

The majority of the boss monster levels use a non-zero "fight mode" value in the SNES Registers and Level Modes setting. This causes the game to skip looking at the level object data so it can load its own custom tile map and settings for that particular boss. This effectively means that there's no object data for Lunar Magic to render.

However, sprites tend to be exempt from this. Some people have taken to temporarily changing the level mode setting mentioned above to some other value so Lunar Magic will let them add sprites to the level, then changing the setting back before saving it.

FAQ: How do I create an IPS Patch?

To create an IPS patch to distribute your custom version of SMW to others, you can either use the "Create IPS" command in Lunar Magic's "File", "Restore" menu, or you can use a separate utility (such as Lunar IPS (LIPS), which you can get from FuSoYa's site). You'll need a copy of the original **unmodified** ROM and your modified ROM to create a patch.

It's best not to distribute entire ROMs, as it's illegal and rather inconvenient for some of the people that have to download it.

Do not apply an IPS patch for SMW to anything other than an unmodified copy of the ROM, and do not apply multiple patches to the same ROM unless you know what you're doing. Doing so can have unpredictable (and usually undesirable) effects.

Also remember that ZSNES and snes9x have internal support for auto-patching IPS files to ROMs. All you have to do is place the ROM and the IPS file in the same directory and give them the same file name (ie. smw.smc and smw.ips). The ROM will be patched in memory when you play the game, leaving the original ROM file intact.

FAQ: Why are my changes not showing up when I play the game?

First, make sure that you're actually editing the same ROM that you're trying to play. It may sound obvious, but it's an easy mistake to make when you have more than one copy of the ROM laying around. To avoid the issue, you can have the editor run your emulator with the currently open ROM for you.

Most SNES emulators have auto-IPS patching support. Move any IPS files in the folder that the ROM is in to a different folder, or your emulator may automatically apply the patch to the ROM.

Be careful not to store your ROM in a protected location, like the Program Files folder. On Windows Vista and later the OS will intercept attempts by 32 bit non-Administrator, non-manifested-as-Vista-or-later programs to write to files in there and redirect them to a totally separate copy of the file that it'll throw in an obscure place like "C:\Users\(username)\AppData\Local\VirtualStore\Program Files\". All such programs would then start to use that copy for reading or writing. But 64 bit programs, programs run as Administrator, and programs manifested as being for Vista or later will still see and use the original ROM that's in the Program Files folder, which can result in considerable confusion.

For testing overworld changes, make sure to start a new game. A new game in SMW is started by selecting a game slot that displays the word "empty" on the title screen. Selecting a game slot that displays a "0" does NOT count as starting a new game! And neither does using a savestate. The reason this is necessary is that the directions to allow the player to leave each level from on the overworld are only initialized once at the start of a new game, and get updated individually as you move between levels and also when you pass them. Savestates are even worse, as key portions of the overworld map data are also decompressed/loaded from the ROM only once: after the player select menu on the title screen.

FAQ: Why is the overworld displaying corrupt path tiles?

Did you remember to uninstall the title moves recording asm after using it? If not, do so... and don't use any savestates made while it was installed.

If the corruption is only for brief moments as an event tile fades in/out, you should read about the restrictions involved with the event path fade effect in the Overworld Extra Options section, which has an option for turning the fade effect off if needed.

Change Background Map16 Bank

The original backgrounds in Mario World are 432 pixels in height, and each background could only access a single page of BG Map16. The height was increased to 512 pixels in Lunar Magic version 1.70 to allow backgrounds in vertical levels to correctly tile, and the page limit was replaced with a bank limit. Each background can now access up to a bank of Map16 tiles at a time (0x10 pages, so pages 40-4F, 50-5F, etc). The tiles that can be accessed can be found in the Map16 Editor.

This dialog will let you change which bank of Map16 tiles is used to render the background tile map for this particular level.

Copy Background Image

This can let you easily copy the background of another level in the ROM to the level you currently have open. The BG palette and back area color settings in the header will also be altered to match the source level. Naturally you cannot use this on layer 2 levels, since they have a level instead of an image on layer 2. Also, certain tile sets simply aren't compatible with certain backgrounds, although Lunar Magic will usually warn you about this (the one thing it doesn't check compatibility for is the animated tiles).

If you're using a custom palette or bypassing the FG/BG GFX, you will have to set those up yourself.

Remap Background Tiles

Since there are times when importing a background where you may import background map16 that does not reside in the same location as the original tiles, you can use this dialog to add or subtract an offset from all 16x16 tiles in the background image, or to do more fine tuned remapping.

For example, if the Map16 tiles were originally on page 0x42 but you've imported them into page 0x43 (which is exactly 0x100 tiles later), you would put 100 for the offset to add.

If adding the offset to any of the BG tiles results in a tile within a different BG bank, the current BG bank setting for the whole background will be changed to match. Note however that you cannot use 16x16 tiles from 2 different BG banks in the same background.

You can do more advanced remapping of the tiles by entering a list of paired comma separated values in the large text edit in this dialog. The first value can be a single value or a range indicating the source of what you want to change, while the second value describes what to change it into. Each pair of values can be on their own line, or you can just separate them with more commas. There are several examples given in the dialog for the syntax you can use. The simplest is changing one tile into another: 100,25 would change tile 100 into tile 25. Or you can do a range:100-101,25 would change both tiles 100 and 101 into tile 25.

To add or subtract an offset, you must specify the sign to use. 100-101,+25 would change 100 to 125 and 101 to 126. Or 100-101,-25 would change 100 to DB and 101 to DC. But since the intent of adding/subtracting offsets is usually to move a range of tiles to another area, it may be easier to just use 'M' and the starting destination: 100-101,M125 would also change 100 to 125 and 101 to 126. It's also common to move a rectangle of tiles, so you can specify the top left and lower right points of a rectangle using a command like R100-111,M25 to change tiles 100 to 25, 101 to 26, 110 to 35 and 111 to 36.

When entering a list of remap pairs, it's important to remember that the source values always refer to the tiles as they currently exist *before* any remapping occurs, not as they would exist after processing the pairs that come before it. For example, you might think the sequence 100,25,25,100 would be of little use, but it will really cause tiles 100 and 25 to be swapped. This also means in a sequence like 100,25,100,30 the first pair is effectively useless as it's superseded by the second pair.