**Tyab's Manic Miners .dat processing utility.**

This program allows many operations on Manic Miner map .dat files.

The following features are supported:

* scanning a map file for syntax correctness.
* fixing some map errors related to tiles, heights, crystals, and ore.
* resizing a map either larger or smaller.
* merging data from one map into another map.
* replacing the script data with data from a separate file. See the script section below for details.

Simple help is shown by using the -help option.

There are three types of files used by the utility.

* A source .dat file used mainly for source merge tiles, height, crystals, ore.
* An existing destination .dat file that is both input and output. It is read in, possibly merged with data from the source file and written.
* A text file treated as a script to replace the destination script section.

For both the srcmap and outmap, those files are scanned for correctness.

All parameters may be enclosed in double quotes.

Filenames that have spaces need to be enclosed in double quotes.

If outmap does not exist, you can create it by supplying a srcmap and using -copysrc which will copy all srcmap data after fixing into outmap. Any map data in outmap is lost being completely replaced by srcmap. You can use resize and offsets, and they will result in outmap being a different size with the srcmap data at the provided offsets.

If outmap is being created by using -copysrc, then the merge options are not valid since you are already copying over all of the map's data. However as stated above you can use the resize and offset values. This can be useful if creating a larger map from a smaller map and you want the smaller map to be centered in the larger map.

When merging tiles, the original solid rock regular walls are also copied to outmap. This is so any walls that needed them are preserved.

On maps, the origin 0,0 is the upper left corner. Positive offsets move the data towards the lower right. Negative offsets move the data towards the upper left.

After operations, the outmap automatically will have the border tiles all set to solid rock regular which is tile id 38.

On the outmap, resize is always performed prior to merge, you can combine resize and merge in the same operation.

During merge, clipping is done automatically so you don't have to worry about the srcmap being larger than outmap nor worry about offsets. Offsets during merge can be negative and works as expected.

Command Line Options:

-help display help

-srcmap *filename* file name of a source merge .DAT

-outmap *filename* file name of a destination .DAT

-overwrite allow changing existing outmap

-copysrc outmap is recreated from srcmap, implies -overwrite

-mergeheight merge height values from srcmap into outmap

-mergecrystal merge crystals values from srcmap into outmap

-mergeore merge ore values from srcmap into outmap

-mergetile merge tile values from srcmap into outmap

-offsetrow *number* row offset when merging/copying srcmap into outmap, default 0

-offsetcol *number* col offset when merging/copying srcmap into outmap, default 0

-resizerow *number* resize outmap rows for tiles,height,resources

-resizecol *number* resize outmap cols for tiles,height,resources

-deftile *number*  value for invalid tiles or resize, default 1 which is a simple ground tile.

-defheight *number* value for invalid heights or resize, default 0

-defcrystal *number* value for invalid crystals or resize, default 0

-defore *number* value for invalid ore or resize, default 0

-mapname *string* levelname: value saved in outmap info section

-creator *string* creator: value to be saved in outmap info section

-fix fix invalid/missing tile, height, crystal, ore values. Changes associated errors to warnings.

-script *filename* file name of script file to replace outmap's script.

-sincdirs *paths* ; set of paths to search for included scripts

-sfixspace will fix script saved in outmap removing spaces where not allowed

-snocomment will strip comments from script except for those that use #.

-sdefine *name=value* define variable name to be replaced with value

Options may be specified in any order.

**Examples:**

Scan a map for correctness:

**-srcmap a.dat**

Copy a map to a new filename keeping the same size.

**-srcmap a.dat -outmap b.dat -copysrc**

Copy a map to a new filename and resize the map to be 52 rows and 48 columns

**-srcmap a.dat -outmap b.dat -copysrc -resizerow 52 -resizecol 48**

Copy a map to a new map with a different size offsetting rows by 10 and columns by 9.

**-srcmap a.dat -outmap b.dat -copysrc -resizerow 52 -resizecol 48 -offsetrow 10 -offsetcol 9.**

Resize an existing map to be 52 rows and 48 columns.

**-outmap a.dat -overwrite -rowresize 52 -colresize 48**

Merge the tiles and heights from one map, into another map, offsetting by 10 rows and 9 columns

**-srcmap a.dat -outmap b.dat -overwrite -mergetile -mergeheight -offsetrow 10 -offsetcol 9**

Use a separate script file for outmap and give list of directories to search for it and included scripts.

-outmap b.dat -script myscript.scr -sincdirs "..;..\scripts;c:\scripts;"

**Script File Processing.**

Using the -script and -soutdirs options you may specify a separate script file to replace the script within outmap. The file you give will be read in and it will also include any included scripts using the list of paths you give in -soutdirs. This is similar to the PATH environment variable in how you format the list of paths to search. The operation is similar to how all compilers find included files.

To find a script file the following process is used:

* The full filename provided in -script. If not found, we break the -script into a path and filename part.
* The directory where outmap is located.
* Directories where prior scripts were found.
* The ; separated list of paths specified in -soutdirs. We try both just the filename and the full filename.

The same is applied to any included script usings its filename and any path components.

**-sfixspace** will remove spaces/tabs from the script where it the Manic Miner engines does not allow spaces. These spaces will no longer generate errors. Using this option allows the ability to indent event chains and other readability changes in the input source files and they will be automatically fixed to work with the Manic Miner engine.

**-snocomment** will strip all comments, except pragma’s and those that start with #. Note that comments do not slow down the Manic Miner engine, since it strips all comments prior to processing.

**-sdefine** will allow command line defining of name=value pairs just like #pragma define name=value

**Pragma comments**

Pragma’s are special commands used by the script pre-processor. They must start at the beginning of a line, and are specified as #pragma. If #pragma is used but it is not at the beginning of a line, it is ignored but will generate a warning.

When a #pragma comment is processed it is saved into the script as ##pragma to prevent it from being processed again by a later extraction and run. When a script is read in, all #pragma commands are processed first before any other script processing happens.

**#pragma include "filename"**

Filename does not have to be quoted, but is a good idea to do so. This will allow you to include another script at this point. That entire script will be embedded and then the remainder of your script will follow. Embedded scripts may embed other scripts. An error is generated if the filename cannot be found. The -sincdir option provides the paths that are used to search for the filename in addition to the path that outmap uses.

There is no limit to the depth of included scripts. To prevent duplicate or recursive includes, it does check to see if a file was already included and it will be ignored if included again. This check is based on filename only, so you cannot include two files with the same name from different directories.

**#pragma TyabScriptDate**

The last modified time for the main -script file will be appended to this comment.

**#pragma TyabScriptIncDate**

The last modified time for the current included script file will be appended to this comment.

**#pragma define *name*=*value***

This is a simple macro substitution. Occurrences of name will be replaced by value. If you want name to be a string, enclose value in double quotes and that string with the quotes will be put into the script. For integer values, just use a number. Name must be unique or an error is generated. The name substitution will not modify comments. Because all #pragma define statements are processed before any other processing of the script, ones defined later can be referenced prior to the define. This is just like how variables work in the script engine.

There are predefined #pragma defines automatically setup and can be used without you needing to define them.

* **TyabMapRows** Integer value. This is the number of rows in the map.
* **TyabMapCols** Integer value. This is the number of columns in the map
* **TyabScriptDate**  String value. YYYYMMDD for the last modified time of the input script.
* **TyabScriptIncDate** String value. YYYYMMDD for the last modified time of current included script.

WORK IN PROGRESS: Complete script file integrity checking….