NAME

mapper – battle grid save file format

DESCRIPTION

The version 17 file format for the **mapper**(6) battle grid program is described in this document.

The file is plain-text 7-bit ASCII. Each line describes one attribute of one object. Any given object will be described by multiple lines of the file, which are not necessarily contiguous, nor are the lines necessarily in any particular order in the file.

The software SHOULD allow Unicode text encoded as UTF-8 everywhere (of which 7-bit ASCII is a subset), but this has not been extensively tested, thus the above note that the file is plain text ASCII.

Five kinds of objects may appear in this file: map features, players, monsters, map file references, and images.

Meta Data

The first line of the file may have the format:

__MAPPER__:*version* [*datalist*]

Where *version* is the map file format version this file conforms to. While map clients should try to read any file they are given, they may not be able to understand anything in a file with a higher version number than they support. Such a client may refuse to read the file.

The *datalist* parameter is a TCL list of values. For version 17 files, this includes

{*comment date***}**

where *comment* is a string describing what the map represents, and *date* is a list containing one or two values: a decimal timestamp of when the file was created, and optionally a human-readable form of that same timestamp. The human-readable string may be empty or omitted.

This line must appear before any other line in the file if it is present. If it is missing, a client will not know what version of the file format it uses. The client may try to read it anyway, generate a warning, or refuse to load the file.

For example:

__MAPPER__:17 {antechamber {1570643001 {Wed Oct 09 10:43:21 PDT 2019}}}

Overall Guidelines

Every line in the file is a properly-formatted TCL list string. See the official TCL language documentation for the full and gory details, but the following should be enough to understand the file contents described here:

- Each line is a list of values separated by spaces.
- Each value in the list is itself surrounded by curly braces if it contains any embedded spaces (thus, everything inside braces is a single value as far as the list is concerned).
- Braces must be balanced.
- Braces and backslashes may be escaped with a backslash if they are to be literal characters rather than having their normal meaning.
- Braces should be omitted where they are not required.
- There is no syntactic difference between a string value with embedded spaces inside it and a sub-list; if a string value is expected to be itself a sub-list of values, then that string value is split on space delimiters just as the outer list was. (In other words, it's up to the software reading the data to determine whether a particular value is a string or a list of strings).

Consider the following strings and their meaning as lists of values:

a b c d	("a", "b", "c", "d")
a {b c} d	("a", "b c", "d")

a {b c} d	("a", ("b", "c"), "d")
a b {{c d} e f}	("a", "b", ("c d", "e", "f"))
a b {{c d} e f}	("a", "b", (("c", "d"), "e", "f"))

One thing that may not be obvious, however, is that the *entire* line is a TCL list. For a line describing an attribute of a map feature, the first value in the list is the attribute name and object ID. The second (and possibly subsequent) values in the list give the value(s) of that attribute. Some attributes expect multiple values after the ID, others expect a single value that is itself a sub-list of values, as documented in each individual case.

Valid IDs for the objects described herein must conform to the following rules:

Attributes — The names of all attributes must consist *only* of upper-case letters and underscores.

Creatures — The ID given for a creature (monster or player) must consist only of upper- or lower-case alphabetic characters, digits, underscores ("_") and octothorpes ("#").

Features — The ID given for map features must consist of hexadecimal digits (of any case, but note that *case is significant* here), underscores ("_") and octothorpes ("#").

Files — The ID given for map files mentioned in this file is not specified here but is a server-specific value. Programs which read and write these files should consider them opaque string values.

Images — The ID given for images stored on servers is not specified here and depends on the server implementation. Programs which read and write these files should consider them opaque string values.

Map Features

When describing map features, each line is in the format:

*attr***:***id* [*value-list*]

where *attr* is the name of the object attribute, *id* is the object ID being described, and *value-list* is a list of values appropriate to the attribute type, if needed by the attribute being described, formatted as a proper TCL list string. (In a nutshell: space-separated list of text strings. The list or any element will be surrounded by curly braces if it contains any spaces.)

Any attribute name which begins with an underscore ("_") is considered to be for internal use only and is not documented here nor guaranteed to be useful or understood by anyone consuming these files. They should be ignored.

The possible attributes, and their value lists, include:

TYPE — The object type. May be one of: **aoe** (spell area of effect), **arc** (filled arc), **circ** (filled circle), **group** (logical grouping of objects into a single compound object), **layer** (definition of a layer), **line** (set of one or more connected line segments), **poly** (filled polygon), **rect** (filled rectangle), **text** (text), **tile** (decorative background tile), or **token** (OBSOLETE TYPE intended for player or creature token placed on the battle grid, but those are now just tiles). Clients are not expected to recognize this type and it may be removed from this document in the future.

X — The *x* coordinate of the object's reference point on the grid.

Y — The *y* coordinate of the object's reference point on the grid.

Z — The *z* coordinate of the object. This is the *stacking order* of the object on the flat 2D plane of the map, with higher numbers overlapping lower numbers. Ideally these should be unique within a map file; which of two objects having the same *z* coordinate will be on top is undefined. For the creature's location in the vertical direction, see the **ELEV** attribute.

LEVEL — An integer describing the level on which this object exists. The mapper can display maps a level at a time, with a visual stacking effect if appropriate. Generally, the level numbers increase as one moves farther underground. By convention, you may use level 0 for the ground level, and negative level numbers for floors above ground.

GROUP The ID number of the group object to which this element belongs.

POINTS Additional coordinate pairs needed for the object type (e.g., the vertices of a polygon or opposing corner of a rectangle). This is a list of coordinate values as *x1 y1 x2 y2 ... xn yn* (always an even number of elements).

FILL The color used to fill in the object. May be a name, an RGB color specification (e.g., "**#***rrggbb*") or an empty string ("**{}**"). An empty string indicates that *no* filling will be done.

FONT The font used to display a text object. This has a form acceptable to Tk as a font name. *(Text objects only.)*

TEXT The text to display in a text object. *(Text objects only.)*

ANCHOR The placement of a text object relative to its *x* and *y* coordinates. Values may be **center**, **n**, **s**, **e**, **w**, **ne**, **nw**, **se**, or **sw**. *(Text objects only.)*

ARROW The style of arrow to place on the ends of lines. Values may be **none** (no arrowheads), **first** (arrow on initial point of the line), **last** (arrow on final point of the line), or **both** (arrows on both ends). *(Line objects only)*

DASH Indicates the dash pattern to use for the lines outlining the shape. If this is missing or has the empty string as its value ("**{}**"), then a solid line is drawn. Otherwise it may have any of the following values which represent the dash pattern: **–** (long dashes), **,** (medium dashes), **.** (short dashes), **–.** (long and short dashes), or **–..** (one long followed by two short dashes).

LINE The color used to draw the outline around a filled object. May be a name or RGB color specification (e.g., "**#***rrggbb*").

WIDTH The line width used to render the object's outline.

LAYER The ID of the layer on which this object exists. In version 1.0 of **mapper**, this is nearly always "**walls**", because layers are not yet implemented. The only other value recognized in this attribute is "**tiles**" which is rendered underneath *everything* (including grid lines).

JOIN The line join style. May be one of: **bevel**, **miter** or **round**. *(Polygon objects only.)*

SPLINE The number of lines to use per polygon line segment for spline curve smoothing. 0 (zero) means no smoothing (just straight lines between points). *(Polygon objects only.)*

ARCMODE The drawing style of an arc object. May be **pieslice**, **arc**, or **chord**. *(Arc objects only.)*

START The starting angle for the arc, in degrees counter-clockwise from the 3:00 position. *(Arc objects only.)*

EXTENT The extent of the angle for the arc, in degrees counter-clockwise from the start position. *(Arc objects only.)*

HIDDEN If this attribute is true, the object (which may be a group) is not shown at all on the map. If a group is hidden, none of its constituent objects need this attribute set.

IMAGE The name of the image to be displayed for this element. The value is fairly arbitrary and must already have been defined by loading the image via the network connection. If it wasn't, a file will be looked for in the tile cache directory. *(Tile objects only.)*

BBHEIGHT The pixel height of the bounding box for the image named in the **IMAGE** attribute when the map is zoomed at the default level. *(Tile objects only.)*

BBWIDTH The pixel width of the bounding box for the image named in the **IMAGE** attribute when the map is zoomed at the default level. *(Tile objects only.)*

AOESHAPE The shape of the area of effect (**aoe** type objects only). May be one of: **cone**, **radius**, or **ray**.

LOCKED A boolean value. If false (or this attribute is not present), then this object may be edited by clients. Otherwise, the object is immutable. Clients should not even recognize the object as a candidate to be selected for operations or modified by the user. It just becomes part of the background. (This is intended to prevent accidentally moving fundamental items such as the

dungeon background images when clicking on something else.)

Programs should accept any additional attributes not listed here but need not process them.

Monsters

Each line describing a monster has the form:

M *attr***:***id* [*value-list*]

where

id is the object ID of the creature being described, *attr* is one of the attributes described below, and *value-list* is the list of values for that attribute (see the Map Features section above for a description of how these may appear).

The attributes for monsters include:

TYPE The type of creature. May be one of: **player** or **monster**.

NAME The displayed name of the creature.

GX The *x* coordinate of the creature on the battle grid *in grid units* (not pixel units).

GY The *y* coordinate of the creature on the battle grid *in grid units* (not pixel units).

HEALTH The creature's current health status. If its value is the empty string or the attribute is missing, no health information is being tracked for this creature. Otherwise, this is a list with the following elements:

max Maximum hit points (at full health).

lethal Number of hit points lost due to suffering lethal damage.

sub Number of hit points lost due to suffering non-lethal damage.

con Number by which hit points are allowed to drop below zero before death occurs.

flat A boolean indicating if the creature is flat-footed.

stable A boolean indicating if the creature was stabilized.

condition The creature's condition. Under normal circumstances this is left as the empty string, in which case the map client will automatically assess and report the creature's condition based on the other health parameters. Putting a non-empty value in this element overrides all such calculation, forcing the reported condition to be whatever is explicitly given here. This only considers conditions directly related to their injury/health level, and may be one of the following: **normal** (none of the below-named conditions are in effect), **flat**, **staggered**, **unconscious**, **stable**, **disabled**, or **dying**.

There is not a specific *condition* code to indicate that the creature is dead. This is indicated by the **KILLED** attribute, which overrides any meaning of the **HEALTH** attribute. (I.e., if **KILLED** is true, the **HEALTH** attribute is ignored other than to indicate that a health bar is desired at all for the creature.)

This value shoud be set to the empty string unless you wish to manually override the condition. If a value is placed here for *condition*, then that is the condition. Otherwise, the condition is calculated from the other stats listed here.

blur This is an optional element which indicates the percentage of "blur" applied to the reported lethal damage amount by the GM's server. See **mapper**(6) for a description of this effect. If this element is omitted, is the empty string, or has a value less than or equal to 0, this means the GM's server is reporting accurate values.

ELEV The creature's elevation relative to some convenient reference point, such as sea level or the floor of the room they're standing in (or, as the case may in fact be, flying above). If they are resting on the ground (i.e., at the reference elevation), this should be 0 (zero). This value need not be an integer.

MOVEMODE The means of locomotion used by the creature is indicated by this attribute. It may be one of: **fly**, **climb**, **swim**, **burrow**, or **land** (the latter indicating the creature is moving normally along the ground, typically by walking or running.) If this attribute is set to the empty string, the default mode of "**land**" is assumed.

COLOR The color name for the threat zone around the creature.

NOTE An arbitrary text label to note special conditions of that creature, such as being weakened, dazed, blinded, etc.

SKIN For creatures with multiple images (such as shapechangers), this gives the image number currently displayed on the map. The default (base) image is 0.

SKINSIZE For creatures with multiple images (such as shapechangers), provides a list of size codes corresponding to their legal skin numbers. This is a list of *n* values for creatures which can have *n* different images, where each element gives the size of each skin starting with skin 0. For example, if the base creature is medium-size, as is its first alternate image, but it has one additional image that is large, the **SKINSIZE** attribute would have a value of **{M M L}**.

SIZE The size of the area occupied by the creature. May be an integer number of squares or one of the size category codes **F** (fine), **D** (diminutive), **T** (tiny), **S** (small), **M** (medium), **L** (large), **H** (huge), **G** (gargantuan), **C** (colossal). Capital letters indicate tall creatures while lower-case indicates long creatures, for size categories where that makes a difference.

STATUSLIST This is a list of status values which indicate the condition of the creature. The exact list of statuses is configurable by the GM but the standard mapper comes with the following set:

ability drained Marked with a small orange downward triangle on the left of the token.

bleed Marked with a small red downward triangle on the left of the token.

blinded Black double-slash (//) through the token.

confused Orange double-horizontal lines through the token.

cowering Brown double-horizontal lines through the token.

dazed Purple double-horizontal lines through the token.

dazzled Cyan double-slash (//) through the token.

deafened Small black diamond at the left of the token.

disabled Red vertical line through the token.

dying Slash (/) through the token in the creature's area-fill color.

energy drained Small black downward triangle on the right of the token.

entangled Green hash (#) through the token.

exhausted Red downward triangle around the token.

fascinated Cyan double-horizontal lines through the token.

fatigued Orange downward triangle around the token.

flat-footed Blue ring around the token.

frightened Yellow diamond around the token.

grappled Orange cross (+) through the token.

helpless Gray double-horizontal lines through the token.

incorporeal Gray ring around the token.

invisible Dashed black ring around the token.

nauseated Green upward triangle around the token.

panicked Red diamond around the token.

paralyzed Black double-horizontal lines through the token.

petrified Gray X drawn through the token.

pinned Red cross (+) through the token.

poisoned Small green circle at the right of the token.

prone Blue line through the token.

shaken Brown diamond around the token.

sickened Brown upward triangle around the token.

stable Small brown diamond at the right of the token.

staggered Blue double-horizontal lines through the token.

stunned Red double-horizontal lines through the token.

unconscious Purple double vertical line through the token.

Additional values are permitted but will be silently ignored by clients unless they have been defined locally (e.g., by the **DSM** protocol command). In this way, any number of conditions and their corresponding markers may be put in use for a particular game. These markers are used to draw status indicators on the creature tokens even when not in battle conditions, whereas the **HEALTH** attribute is more combat-oriented and will auto-generate some conditions if that attribute exists.

Similar symbols are used for conditions that have similar effects (e.g., double horizontal lines for conditions usually associated with denying some or all normal actions to the creature).

AOE If this attibute exists and has a non-empty value, then the creature has an ongoing magic emanation effect centered on them currently in effect. The *value* is a list with at least one element. The first element gives the shape of the emanation. The supported shapes are:

radius This specifies a spherical area centered *around the perimeter* of the creature's occupied space. The remaining elements in the list are:

r The radius of the effect, in number of grid squares.

color The color with which to fill in the area on the map.

AREA The threatened area around the creature, using the same values as for **SIZE** (see above).

REACH Integer value indicating if the creature currently has reach mode enabled. 0 indicates normal melee threat range for the creature's size. 1 indicates that a reach weapon is in use, so that it only threatens far spaces but not adjacent ones. 2 indicates that the full range from the creature out to the reach limit is threatened.

KILLED Boolean value indicating if the creature is currently dead.

DIM Boolean value indicating if the creature is currently *not* in the active initiative slot.

Programs should accept any additional attributes not listed here but need not process them.

Players

Each line describing a player has the form:

P *attr*:*id* [*value-list*]

and has the same format as monsters (see above).

Maps

Each line describing a server-side map file which may be retrieved by a client has the form:

F *fileID*

where *fileID* is the server-side ID code used to retrieve the file.

For example, given the map file line

F abcdefghijklmnop

the mapper will retrieve the file from a web server at a URL such as **https://example.com/gma/map/a/ab/abcdefghijklmnop.map**. Note that the presence of this line causes the map client to retrieve the file into its local cache in preparation to read it at a future point. It does not cause the file to be opened and its contents displayed.

Images

Each line describing an image has the form:

I *imageID zoom filename*

where *imageID* is an identifier for the image, *zoom* is the zoom factor this image is displayed for, and *filename* is the name on disk where the image file should be loaded from. This is a pathname opened just as given in the file (i.e., it is relative to the current working directory of the **mapper** process unless it is an absolute pathname).

If *filename* begins with an at-sign ("@") character, then rather than being a local filename it is instead an ID tag used to download the image file from an external service. (The **rendersizes**(6) utility knows how to generate these files appropriately and will create suitable entries in a map file for them if given a **−w** option.)

Typically these files will be available from a web server using a URL based on the remaining characters of *filename*.

For example, given the map file line

I #SimonKazar 1.0 #SimonKazar@50.gif

the mapper will load the contents of the local disk file **#SimonKazar@50.gif**, and use that to display the image with the internal (to the mapper) image name **#SimonKazar** (which it would automatically do for a character token for a creature named **SimonKazar** who was still living), at zoom factor 1.0.

On the other hand, the line

I #Firefly 2.0 @0oSmGY0XERJRrA8ZiK_igg_Firefly@100

tells the mapper client that the image data for the tile **#Firefly** at zoom factor 2.0 may be obtained from a web server at a URL such as

https://example.com/gma/map/0/0o/0oSmGY0XERJRrA8ZiK_igg_Firefly@100.gif

(although multiple image formats may be provided other than GIF, which the client may request using appropriate filename suffixes; the specific formats provided are site-dependent but for the author's game they include GIF, JPEG, and PNG formats with suffixes **.gif**, **.jpg**, and **.png** respectively).

EXAMPLE

```
__MAPPER__:17 {{Example map} {1570643001 {Wed Oct 09 10:43:21 PDT 2019}}}
TYPE:1 line
LINE:1 black
FILL:1 #00e400
X:1 425
```

Y:1 107
WIDTH:1 5
LAYER:1 walls
POINTS:1 {433 337 553 392 600 243 424 108}
TYPE:2 rect
LINE:2 black
FILL:2 #00e400
WIDTH:2 5
X:2 593
LAYER:2 walls
POINTS:2 {807 443}
Y:2 122

This file contains two objects:

#1 is a black line filled in the shade of green with (R=0, G=0xe4, B=0), with line segments of width 5 going between points (425,107) – (433,337) – (553,392) – (600,243) – (424,108).

#2 is a black rectangle filled in with the same shade of green as line #1, with a line width of 5, between opposing corners (593,122) and (807,443).

WEIRD SIZES

While the mapper implements the standard d20/Pathfinder creature size categories, including tall (uppercase) and wide (lowercase) variants, sometimes there are special cases which fall outside that list. The following special codes are also usable:

L0/l0 Large creature with no threat zone and no reach (such as a swarm of teensy critters).

M20/m20
Medium creature (5-foot space, 5-foot threat zone) with a 20-foot reach zone.

SEE ALSO

mapper(6), **rendersizes**(6).

FILES

~/.gma/mapper/cache/*name***@***zoom***.gif**
The GIF-format file for image with the given *name* and scaled to the given *zoom* factor.

~/.gma/mapper/cache/*id***.map**
The map file with server ID *id*.

HISTORY

This describes the 4.0 (GMA/Python) and 3.19 (Standalone Tcl/Tk) versions of **mapper**.

Changes as of File Format 2

Version 2 files differ from version 1 files in that they add the **JOIN**, **SPLINE**, **START**, **EXTENT** and **ARCMODE** attributes, the **arc** object type, and the "no-fill" (empty string) mode for the **FILL** attribute.

Changes as of File Format 3

This file retains compatibility with the version 2 format, with the following differences.

Each object now has a z coordinate in addition to the (x,y) coordinates of its reference point as in previous versions. The z coordinate indicates the display "stacking order" of objects on the screen. Each object is "flat" and are rendered in order from the smallest to the largest value of z so that the lowest z value is at the "bottom" with the others overlapping on top of them. These serve only to indicate the relative positioning of objects. There is no absolute meaning to any particular value for z.

Added the **HIDDEN**, **LEVEL**, and **GROUP** object attributes.

Changes as of File Format 4

This format is compatible with File Format 3, except that it adds the following attributes to support tile objects:

LAYER
The layer attribute may now include the value **tiles** which is rendered below everything else, including the grid lines.

IMAGE
This attribute appeared for the first time in this version.

SIZE This attribute appeared for the first time in this version.

Programs which read newer version files should deal correctly with older files which do not contain the newer elements.

Changes as of File Format 5
This introduced images, creatures, and spell areas of effect.

Changes as of File Format 6
The image definition syntax was extended to allow externally downloaded image files.

Changes as of File Format 7
Additional creature attributes were added, such as **NOTE** and **SKIN**.

Changes as of File Format 8
Added text objects. This includes the addition of the **ANCHOR**, **FONT**, and **TEXT** attributes.

Changes as of File Format 9
Added **ARROW** and **DASH** attributes. For backward compatibility with format 8, if either of these are missing from the file, defaults are assumed.

Changes as of File Format 10
Added **ELEV** and **MOVEMODE** attributes.

Changes as of File Format 11
Added **HEALTH** attribute.

Changes as of File Format 12
Added **F** record type.

Changes as of File Format 13
Added the optional *blur* element of the **HEALTH** attribute. Removed the **SELECTED** attribute. This is a transient state used during the mapper program's operation and has no point being saved with the object. This attribute was moved to the private name **_SELECTED** instead.

Changes as of File Format 14
Added the **STATUSLIST** attribute.

Changes as of File Format 15
Added the **SKINSIZE** attribute.

Changes as of File Format 16
Added the **LOCKED** attribute.

Changes as of File Format 17
SIZE only applies to creature tokens. The version 16 document incorrectly specified that map elements (notably tiles) also had a **SIZE** attribute but this was never implemented. Instead, they had unofficial attributes **_BBHEIGHT** and **_BBWIDTH** which have now been promoted to official status as **BBHEIGHT** and **BBWIDTH**.

AUTHOR

Steve Willoughby / steve@madscience.zone; Elevation and movement modes added by John Mechalas.

BUGS

The **LEVEL**, **GROUP**, and **LAYER** attributes are not actually implemented yet in the mapper program. They're documented here and will be accepted by the mapper but none of their behavior documented above will actually happen.

The **token** object type is deprecated and clients are free to be unaware of its existence.

COPYRGHT