NAME

gmafile – GMA file access functions

SYNOPSIS

(If package installed globally)
`package require gmafile`

(Otherwise)
`source gmafile.tcl`

`::gmafile::default_arr` *aname id* ?`-value` *v*? ?`--`? *attr* ?*attr*...?

`::gmafile::load_dice_presets_from_file` *f* → *meta plist*

`::gmafile::load_from_file` *f* → *meta objlist*

`::gmafile::load_legacy_map_data` *vlist meta* → *meta objlist*

`::gmafile::load_legacy_map_file` *f ver oldmeta* → *meta objlist*

`::gmafile::require_arr` *aname id attr* ?`attr`...?

`::gmafile::load_legacy_preset_file` *f ver oldmeta* → *meta plist*

`::gmafile::save_arrays_to_file` *f meta elements elementtypes creatures*

`::gmafile::save_dice_presets_to_file` *f objlist*

`::gmafile::save_to_file` *f objlist*

`::gmafile::upgrade_elements` *filedata* → *dictlist*

DESCRIPTION

This module provides the functionality for reading and writing data files used by GMA, including map files and saved die-roll presets. Reading of legacy formats is supported, but new files will only be saved in the current format.

`::gmafile::default_arr` *aname id* ?`-value` *v*? ?`--`? *attr* ?*attr*...?

This function is intended for reading legacy files and may disappear in the future. Ensures that a number of required entries are defined in the array named as *aname*. For each listed attribute *attr*, the array must contain a value associated with the key *attr*`:`*id*. If no such key is found, it will be created with the default value *v* provided via the `-value` option (or the empty string if no `-value` option was given).

The `--` option may be used to explicitly terminate the list of options in case `-value` might be the name of an attribute.

`::gmafile::load_dice_presets_from_file` *f*

Given an open file stream *f*, this reads a set of die-roll presets from that file, returning them as a list of two elements:

meta A dictionary with the following fields, any of which may be empty or zero if they were not found in the file:

- *Timestamp* The time the file claims to have been generated, as a UNIX timestamp value (seconds since the epoch).
- *DateTime* The time the file claims to have been generated, as a human-readable string in arbitrary format. No attempt is made to reconcile this and *Timestamp*, they are simply presented as they were in the file.
- *Comment* A comment that was stored in the file.
- *FileVersion* The file format version of the data read.

plist A Tcl list of die-roll presets. Each of these is a dictionary with the following fields:

- *Name* The name of the die-roll preset as shown to the user. This name may include a vertical bar (`|`). If so, the text up to and including the bar

should be hidden from the user, but used for the purposes of sorting the list on-screen.

Description The description of what the preset is for.

DieRollSpec The actual specification for the die roller `dice`(3).

`::gmafile::load_from_file` *f*

Given an open file stream *f*, this reads map file data from it and returns them as a list of two elements:

meta A dictionary of values describing what was just read. This has the following fields, any of which may be empty:

Timestamp The time the file claims to have been generated, as a UNIX timestamp value (seconds since the epoch).

DateTime The time the file claims to have been generated, as a human-readable string in arbitrary format. No attempt is made to reconcile this and *Timestamp*, they are simply presented as they were in the file.

Comment A comment that was stored in the file.

Location The name of the location which the file represents.

FileVersion The file format version of the data read.

objlist A Tcl list of the objects loaded from the map file. Each element of this list is itself a Tcl list with the following two elements:

type The type of object. This will be one of the types as documented in `mapper`(5), such as `ARC`, `POLY`, `IMG`, `MAP`, `CREATURE`, etc.

d A dictionary which describes the object of the given type. The specific fields for each type are documented in the protocol specification in `mapper`(6).

`::gmafile::load_legacy_map_data` *vlist meta*

This function is provided to handle legacy-format map data and may disappear in the future. You should never need to call this function directly; it is automatically invoked as needed by `::gmafile::load_from_file`. Reads partially-parsed lines from *vlist*. Each element of this list is a 2-element list where the first element is one of:

`IMG` The second element is a dictionary describing an image advertised by the server.

`MAP` The second element is a dictionary describing a map file advertised by the server.

`RAW` The second element is a raw, unparsed line of legacy-format data loaded from the file.

The *meta* parameter is a dictionary of metadata already loaded with information about the input data.

Returns a list in the same format as `::gmaproto::load_from_file`.

`::gmafile::load_legacy_map_file` *f ver oldmeta*

This function is provided to handle legacy-format map data and may disappear in the future. You should never need to call this function directly; it is automatically invoked as needed by `::gmafile::load_from_file`. Reads legacy-format data from the open file stream *f*, which is already known to be in file format version *ver*. The *oldmeta* value is a Tcl list with the old-style metadata as documented in the version 17 revision of `mapper`(5).

Returns a list in the same format as `::gmaproto::load_from_file`.

`::gmafile::require_arr` *aname id attr* ?*attr...*?

This works just like `::gmafile::default_arr` except that it does not set default values into the array named as *aname*. It only checks that the named attributes already exist in the array, stored under keys called *attr*`:`*id*.

`::gmafile::load_legacy_preset_file` *f ver oldmeta*

This function is provided to handle legacy-format preset data and may disappear in the future. You should never need to call this function directly; it is automatically invoked as needed by `::gmafile::load_dice_presets_from_file`.

In a manner analogous to `::gmafile::load_legacy_map_file`, this reads data from the open stream *f* which are in legacy file format version *ver* and have the old-format metadata *oldmeta*.

Returns the same values as `::gmafile::load_dice_presets_from_file`.

`::gmafile::save_arrays_to_file` *f meta elements elementtypes creatures*

Given an open writable file stream *f*, this saves the map data stored in the following array variable names to a mapper data file:

elements This names an array variable containing all of the map elements other than creatures which appear in the mapper client. The keys for each are the object's ID, which should also be the value stored in the object's *ID* field. The values in the array are the corresponding dictionary defining that object's attributes.

elementtypes This names an array variable containing a mapping of object IDs to their internal data type. There must be an element in this array for each element in the *elements* array. The type strings are the ones as known in the mapper code, not the server's protocol, as would be output by `::gmaproto::GMATypeToObjType`.

creatures This names an array variable containing a mapping of creature IDs to a dictionary describing each creature. The keys of this array are not referenced.

The *meta* parameter is a dictionary describing this collection of map data, and has the following fields:

Timestamp The UNIX timestamp representing the date and time when this map data were generated. If this field is missing from the dictionary, the current date and time will be used.

DateTime A human-readable string describing the *Timestamp* field's value. If this field is missing from the dictionary, one will automatically be generated based on the *Timestamp* value.

Comment Any comment you wish to be associated with this map file.

Location The name of the location which the file represents.

`::gmafile::save_dice_presets_to_file` *f objlist*

Given an open writable stream *f*, this saves a collection of die-roll presets to disk. The *objlist* parameter is a Tcl list with two elements:

meta A dictionary describing this file, with the following fields:

Timestamp The UNIX timestamp representing the date and time when this preset data were generated. If this field is missing from the dictionary, the current date and time will be used.

DateTime A human-readable string describing the *Timestamp* field's value. If this field is missing from the dictionary, one will automatically be generated based on the *Timestamp* value.

Comment Any comment you wish to be associated with this file.

plist A Tcl list of dictionaries, each describing a single die-roll preset with the same fields as documented for `::gmafile::load_dice_presets_from_file`.

`::gmafile::save_to_file` *f objlist*

As an alternative to `::gmafile::save_arrays_to_file`, this function writes the map data as found in *objlist*, which is a Tcl list containing the following two elements:

meta A metadata dictionary as described for `::gmafile::save_arrays_to_file`.

elements A Tcl list of all map elements to be saved. Each element of this list is itself a Tcl list of two elements:

type The element type name as known in the mapper software.

d The dictionary describing the object.

`::gmafile::upgrade_elements` *filedata*

This function is intended for reading legacy files and may disappear in the future. Each element in *filedata* is a tuple as described for `::gmafile::load_legacy_map_data`. This function converts all entries of `RAW` type into their new-format equivalents. A new copy of *filedata* with those substitutions in place is returned.

DIAGNOSTICS

An exception is thrown if a serious error is encountered.

Messages are printed to standard output to indicate progress or provide debugging information.

SEE ALSO

`openssl`(1).

AUTHOR

Steve Willoughby / steve@madscience.zone.

HISTORY

This document describes version 1.0 of the `gmafile` package, released in December 2022.

COPYRIGHT