

LibMap DLL Version 1.x.2.x
Documentation Version 2.0.0b
Video Game – LibMap API Documentation

Overview

This documentation will cover the API of the libmap.dll which is used to read and write binary map files.

1. Introduction
2. Definitions
3. API Function List

Introduction

The LibMap DLL is written in C++ and compiled with g++. It reads and writes the binary map format used by the game.

Definitions

Integer

Signed and 4 bytes long

String

C string in C++ (char*) and String in vb.net

Error Code

The error codes are of the integer type and correspond to the following errors:

- 0 – No Error
- 1 – File can not be opened
- 2 – Error reading from the file
- 3 – Error writing to the file
- 4 – Missing root node
- 5 – Missing closing byte
- 6 – Missing opening byte
- 7 – File not open in write mode
- 8 – Coordinates are out of the boundaries of the map size
- 9 – File is not compatible with the DLL version

API Function List

lastError () as Integer

Description

Returns the last error cast by any function.

Return Value

Error Code - Error code cast
0 - If there has been no error

openFile (FileName as String) as Boolean

Description

Opens a file stream in read mode. If the file exists it is loaded. If the file does not exist or can not be loaded by the parser nothing is done. If a map has already been loaded and not closed, the new map will be loaded over the current map replacing any mutual tiles.

Parameters

FileName - Path to the location of the map file

Return Value

True - If the file is loaded successfully

openFileW (FileName as String) as Boolean

Description

Opens a file stream in write mode. It behaves the same way as **openFile()** however the file can be written to and other processes will be blocked from editing the file.

Parameters

FileName - Path to the location of the map file

Return Value

True - If the file is loaded successfully

closeFile ()

Description

Closes any open file and clears all map data.

saveFile () as Boolean

Description

Saves the map data to the file previously opened with **openFileW()**.

Return Value

True - If the file is saved successfully

saveFileAs (FileName as String) as Boolean

Description

Closes any open file without clearing the map data and then opens a file like **openFileW()** but does not load the file but instead saves the current map data to it and then leaves the file open to be saved to later on.

Parameters

FileName - Path to save the file at.

Return Value

True - If the file is opened and saved successfully

setName (Name as String)

Description

Sets the name of the map.

Parameters

Name - New name

getName () as String

Description

Gets the current name of the map.

Return Value

String - Current name

setDescription (Name as String)

Description

Sets the description of the map.

Parameters

Name - New description

getDescription () as String

Description

Gets the current description of the map.

Return Value

String - Current description

setSize (x as Integer, y as Integer) as Void

Description

Sets the maximum size of the map using a position at the maximum coordinates. If the new size is smaller than the old size, any tiles that are now outside the boundary are removed.

Parameters

x - New maximum size in the x direction
y - New maximum size in the y direction

getSizeX () as Integer

Description

Gets the current maximum size in the x direction.

Return Value

Integer - Current maximum size in the x direction

getSizeY () as Integer

Description

Gets the current maximum size in the y direction.

Return Value

Integer - Current maximum size in the y direction

setDefaultSpawn (x as Integer, y as Integer) as Void

Description

Sets the default spawn location for players who long onto the map the first time.

Parameters

x - x component of the default spawn
y - y component of the default spawn

getDefaultSpawnX () as Integer

Description

Gets x component of the current default spawn position.

Return Value

Integer - x component of the default spawn

getDefaultSpawnY () as Integer

Description

Gets y component of the current default spawn position.

Return Value

Integer - y component of the default spawn

**addTile (x as Integer, y as Integer, tileid as Integer, actionid as Integer,
monster as String, npc as String, tp_x as Integer, tp_y as Integer) as Boolean**

Description

Adds a tile to the map. If the tile already exists the older tile is deleted and replaced with this one.

Parameters

x - x position of the tile
y - y position of the tile
tileid - item id of the ground sprite
actionid - action id of the tile used to call scripts, otherwise 0 for no action id
monster - if a monster should spawn on the tile its name, otherwise an empty string
npc - if a npc should spawn on the tile its name, otherwise an empty string
tp_x - x component of the tile's teleport destination, otherwise 0 for no teleport
tp_y - y component of the tile's teleport destination, otherwise 0 for no teleport

Return Value

True - If the tile has been added successfully

removeTile (x as Integer, y as Integer) as Boolean

Description

Removes a tile from the map.

Parameters

x - x position of the tile
y - y position of the tile

Return Value

True - If the tile exists and has been removed successfully

getTileItemID (x as Integer, y as Integer) as Integer

Description

Gets the item id of the ground sprite of a tile. This function can be used to check if a tile exists, since if a tile exists it must have an item id. Thus if it returns 0 the tile does not exist.

Parameters

x - x position of the tile
y - y position of the tile

Return Value

Integer - Item ID of the tile
0 - If the tile does not exist.

getTileActionID (x as Integer, y as Integer) as Integer

Description

Gets the action id of a tile if set.

Parameters

x - x position of the tile
y - y position of the tile

Return Value

Integer - Action Id of the tile
0 - If the tile does not exist or it does not have an action id

getTileSpawnMonster (x as Integer, y as Integer) as String

Description

Gets the name of the monster if one spawns on the tile.

Parameters

x - x position of the tile
y - y position of the tile

Return Value

String - Name of the monster
Empty String - If the tile does not exist or does not have an monster spawn

getTileSpawnNPC (x as Integer, y as Integer) as String

Description

Gets the name of the npc if one spawns on the tile.

Parameters

x - x position of the tile
y - y position of the tile

Return Value

String - Name of the npc
Empty String - If the tile does not exist or does not have an npc spawn

getTileTPX (x as Integer, y as Integer) as Integer

Description

Gets the x component of the teleport destination of the tile if any.

Parameters

x - x position of the tile

y - y position of the tile

Return Value

Integer - x component of the tile's teleport destination

0 - If the tile does not exist or is not a teleporting tile

getTileTPY (x as Integer, y as Integer) as Integer

Description

Gets the y component of the teleport destination of the tile if any.

Parameters

x - x position of the tile

y - y position of the tile

Return Value

Integer - y component of the tile's teleport destination

0 - If the tile does not exist or is not a teleporting tile