Riot archive file format

Archive files are split into two files: a directory and a data file. The directory file has the extension ".raf". It contains information about the contents of the data file. The data file has the extension ".dat". It only contains raw data and has not table of contents.

The following C++ code describes the data structures found in a RAF file.

namespace RAF

{

// Header that appears at start of the directory file

struct Header\_t

{

// Magic value used to identify the file type, must be 0x18BE0EF0

unsigned long mMagic;

// Version of the archive format, must be 1

unsigned long mVersion;

};

// Table of contents appears directly after header

struct TableOfContents\_t

{

// An index that is used by the runtime, do not modify

unsigned long mMgrIndex;

// Offset to the file list from the beginning of the file

unsigned long mFileListOffset;

// Offset to the string table from the beginning of the file

unsigned long mStringTableOffset;

};

// Header of the file list

struct FileListHeader\_t

{

// Number of entries in the list

unsigned long mCount;

};

// An entry in the file list describes a file that has been archived

struct FileListEntry\_t

{

// Hash of the string name

unsigned long mHash;

// Offset to the start of the archived file in the data file

unsigned long mOffset;

// Size of this archived file

unsigned long mSize;

// Index of the name of the archvied file in the string table

unsigned long mFileNameStringTableIndex;

};

}

namespace StringTable

{

// First structure in the built string table

struct HEADER

{

// size of all data including header

UINT

m\_Size;

// Number of strings in the table

UINT

m\_Count;

};

// Entry in the table of contents

struct ENTRY

{

// Offset from START OF THE STRING TABLE

// (i.e. offset from the start of the header structure)

UINT

m\_Offset;

// Size of string including null

UINT

m\_Size;

};

}