Media Container Format

aka

Blob

For use in packing of media files for easy lookup and access

Definitions:

MagicNumber - an unsigned number
HashTableHeader - a vector hash table of MagicNumber entries the InvalidBlobEntry aka an entry with a hash that couldn't be computable to that position, with an offset of 0 to the next entry
HashTableEntry - a small structure that points to yet another HashTableEntry or NULL, and the information associated with a file

Format Blob version 1.0

Offset	Size	Type	
0x0	0x2	u16	MagicNumber
0x2	MN * sizeof(HTE)	packed(u128)[MN]	HashTableHeader
...	packed(u128)	packed(u128)	HashTableEntry
	...	...	...
...	any	file	SomeFile

In the above table an entry of the HashTableHeader will point to the HashTableEntry.
ThisHashTableEntry will point to the file being the following or equivalent:

```
typedef struct BlobEntryV1{
     unsigned int hash;
     unsigned int file_size;
     unsigned int blob_offset;
     unsigned int next_entry_offset; // 0 for not exists
} BlobEntryV1_t;
```

All offsets are from the top of the file.

Other types:

```
typedef unsigned short int MagicNumber;

typedef BlobEntryV1_t * BlobTable;
```

Lookup procedure:

1. Read MagicNumber
2. Access hash table entry and traverse list of hash table entries until one with hash need is found, or 0.
3. Read memory with size specified.

Hash function to be used on file names (that may include a partial path)

Assuming one byte sized characters, for each character from left to right xor current index of unsigned int, this index increases each time and if reaches the size of the unsigned int, roll over bethoven.