File Specification for Calc Projects

Equ Header

This is the overall header format that you will be presented with and you will write an output file using this format. Be sure to note the magic number. That will tell you if a file is valid. If there is any deviation from this format, reported error `INVALID_EQU_HEADER`. Please note, the flags field is going from most significant to least significant. [Compressed | Serialized | Encrypted | Solved | Reserved | Reserved | Reserved | Reserved]

Name	MagicNumber	FileID	NumEquations	Flags	EquationsOffset
Length (Bytes)	4	4	8	1	4
Purpose	Should be 0xDD77BB55	ID of the equation file	Number of Equations in the file	[Compressed | Serialized | Encrypted | Solved | Unused | Unused | Unused | Unused]	Where the operations begin

This is the optional header that you will be presented with and you will write an output file using this format. The purpose of this header is to hold encryption keys, compression, serialization algorithms, and tell you if the file is digitally signed. If there is any deviation from this format, report error `INVALID_OPT_HEADER`. For flags field, same as above: [Asymmetric Encryption Key | Symmetric Encryption Key | Securely Signed | Compression Algorithm | Serialization Algorithm | Reserved | Reserved | Reserved]

Optional Header

Name	MagicNumber	Length	Flags	Data
Length (Bytes)	4	2	1	Variable (See length field)
Purpose	Should be 0xEE88FF99	Length of the header	[Asymmetric Encryption Key | Symmetric Encryption Key | Securely Signed | Compression Algorithm | Serialization Algorithm]	Data described by flags

Optional Header Options Specification

This is the specification for each flag. The length field of each flag tell you how long the data will be

Flags	Asymmetric Key	Symmetric Key	Securely Signed	Compression Algorithm	Serialization Algorithm
Purpose	Public Key of the **receiver**	AES Key for the encrypted data	Do the equations have a digital signature	What compression to use	What serialization to use
Length	Variable	Variable	Variable	1	1

Compression Algorithms:

Name	Miniz
Value	0x01

Serialization Algorithms:

NOTE PROJECT 0 WILL BE BINARY ONLY

Name	JSON	BINARY
Value	0x01	0x02

Equation Format For Binary Serialization

This describes each equation. It is "serialized" into a custom format described below. Note the flags field will tell you if the numbers should be treated as floats or not. Note padding for an equation is out to 32-bytes and

Name	Equation ID	Flags	Equation/Solution	Padding
Length (Bytes)	4	1	variable	variable
Purpose	Unique ID for the equation	Numbers are [Float | Not float]	Equation to solve	Padding out to 32 bytes

Unsolved Equation Format

This is the format of an equation. Each operand and operator is seperated by 4 NULL bytes: operand + "\x00\x00\x00\x00" + operator + "\x00\x00\x00\x00" + operand.

Name	Operand	Operator	Operand
Length	8	1	8
Purpose	64-bit integer	User Defined	64-bit integer

Operators

```
        - Addition (0x01)
        - Subtraction (0x02)
        - Multiplication (0x03)
        - Division (0x04)
        - Modulo (0x05)
        - Left shift (0x06)
        - Right shift (0x07)
        - And (0x08)
        - Or (0x09)
        - XOR (0x0a)
        - Rotate Left (0x0b)
        - Rotate Right (0x0c)
        - 0x0d, 0x0e, and 0x0f are undefined and should report error if used
```

Solved Equation Format

This is format of a solution

Name	Equation ID	Solution
Length (Bytes)	4	8
Purpose	Unique ID for the equation	64-bit integer