What is SIFF?

SIFF stands for Scene Interchange File Format. It is an attempt at creating a reasonable interchange format for 3D data.

Why is it SIFF 1.0?

The 1.0 is specified to show that it is the first level SIFF format. A SIFF 2.0 is planned for the near future. It will accommodate higher order graphics (curves, NURBs, etc.).

Who did it?

SIFF was developed by Sega and a number of its vendors.

What is Sega's motivation?

We have a very real need for portable 3D data. Current interchange methods are either very cumbersome, or very damaging to the data in process. By establishing a common format and enlisting the support of major 3D package vendors, we can achieve this interchange transparently.

Is it limited to video game development?

No. Sega very deliberately made the format general purpose. If we limited the formats usefulness to video games, we would have difficulty in gaining support from major tool vendors. So limiting to format hinders its usefulness to us.

Why would Sega give away such an advantage?

Again, if we keep it to ourselves it becomes more difficult to garner support. By releasing the specification, we hopefully help the graphics community, which in turn will help us with a larger selection of tools.

What is being done to promote it?

Well, we're just getting underway. So, we're preparing sample code and data and working with a few vendors to establish support. Then we'll hit everybody.

What services are available?

Right now, just clarification of the format.

What services will be available?

In the near future, a chunk registry will be set up to maintain the chunk definitions. Also, we'll be able to arbitrate problems within the specification.

SIFF 1.0

Scene Interchange File Format

October 1996

Purpose

To provide a mechanism for the exchange of three-dimensional scene, animation, and model data for the general graphics community.

SIFF and the IFF Standard

The file format described is compliant with the IFF standard as specified by Electronic Arts.

Reference

Since voluminous documentation on the IFF standard exists, this document will not reiterate those descriptions. Most people working with file formats will already be familiar with IFF type file. The official format document is the *EA IFF 85 Standard for Interchange Format File*, Jerry Morrison (Electronic Arts), January 14, 1985.

FORM for SIFF

SIFF 1.0 is specifically defined as a "FORM" group chunk of type "SF1Q", thus the general file format it:

"FORM"
<size of group>
"SF1Q"
<chunk definition>
<chunk definition>
…
<chunk definition>

Alignment and Padding

The IFF standard requires chunks to be aligned on 32-bit boundaries with nulls used to pad out chunks. The size fields account only for actual data, not padding information.

Endian

IFF and SIFF assume the use of big-endian storage. (68000, not 80x86 style).

Ordering of chunks

SIFF chunks are designed to contain the necessary data of relationship with other chunks. As a result, ordering of chunks is arbitrary and no meaningful data should be inferred from such ordering.

SIFF General Notes

3-D Coordinate System

SIFF assumes a right-handed coordinate system with the positive x-axis directed towards the right; the positive y-axis directed down; and, the positive z-axis directed into the screen. Rotations about an axis are clockwise as viewed towards to positive directions.

UV Coordinate System

UV coordinates are used within SIFF to apply texture information to geometry.

Given a standard rectangular texture, the u-axis covers the range [0.0,1.0] with 0.0 indicating the left-edge of the left-most pixel of the texture; and 1.0 indicating the right-edge of the right-most pixel of the texture. The v-axis covers the range [0.0,1.0] with 0.0 indicating the top-edge of the top-most pixel of the texture; and 1.0 indicating the bottom-edge of the bottom-most pixel of the texture.

Since the UV coordinates are represented by floating-point values, they can be specified outside of the [0.0,1.0] range. This is accepted as legal and implies that the texture repeats infinitely in all directions without deviation.

Linear Algebra Conventions

Although implementation details can vary considerably, the following rules define SIFF expected mathematics. Only alternative methods and notation that yield identical result are considered compliant.

- Standard 4x4 homogenous matrices are used to represent transformations.
- Cumulative transformations are of the form: $T^* = T_0T_1T_2\ldots$, where T_N represents the transformation at depth N in the hierarchy.
- Points are transformed to global space with the following: $P^* = PT^*$, where P represents the original point in the form (X,Y,Z,1).

Types

The following data types are used within the SIFF specification. Note that these are more precise names than the identical types defined within the IFF specification.

Uint8	unsigned 8-bit integer (UBYTE in IFF)
Uint16	unsigned 16-bit integer (UWORD in IFF)
Uint32	unsigned 32-bit integer
float	32-bit IEEE single precision floating-point value
string	zero-terminated string

Names

Most of the SIFF chunks have associated "names" which are used as *unique* identifiers of the particular component that a chunk contains.

There is no restriction on the length of names or their contents. No restrictions are defined since naming schemes that a globally legal are very limited in ability. As a result, applications should use reasonable naming schemes. Ensuring legal names for the intended use is the responsibility of the application.

Tags

Many of the SIFF chunks have associated "tags" which contain *user-defined* (not application defined) data associated with a particular component. Tags are zero-terminated strings containing possibly multiple-field data. Semicolons are used to delimit fields within a single tag.

CHUNK TYPES

The following are the defined chunk types SIFF 1.0.

Geometry Chunk Types

The following chunk types define the geometric makeup of the scene. Hierarchy is inherent since each element has a specified parent. No element of the hierarchy is 'displayed' until it is associated with visual information (see Visual Chunk Types below).

All coordinates are local to the given object. The world space in inherited from the hierarchical parent, and can be changed with the transformation chunks. All transformations are cumulative within their hierarchies as would be expected, therefore transformations take place in the world space of their parent.

There are a limited number of restrictions or rules regarding usage of the hierarchy:

- Cycles within the data are restricted (as in two elements having each other as their respective parents).
- Elements that cannot reach global space through the parental chain are restricted. This is also a cyclical graph problem, but the intention of this rule is to ensure that elements can resolve their space in the global coordinate system.

NODE - Generic Node

A Generic Node is a placeholder or hook in the geometry intended to suit organizational purposes. It is also the base form for all of the geometric types.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.

MTRX - Matrix

Matrix data is used to incorporate standard homogenous matrix transforms of the form $\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & 1 \end{pmatrix}$.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
a00	float	
a01	float	
a02	float	
a03	float	
a10	float	
a11	float	
a12	float	
a13	float	
a20	float	
a21	float	
a22	float	
a23	float	
a30	float	
a31	float	
a32	float	

TRAN - Translate

Shorthand chunk for the specification of a translation-only transformation matrix.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
TransX	float	
TransY	float	
TransZ	float	

SCAL - Scale

Shorthand chunk for the specification of a scaling-only transformation matrix.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
ScaleX	float	
ScaleY	float	
ScaleZ	float	

ROTX - Rotate about the X-Axis

Shorthand chunk for the specification of an x-axis rotation-only transformation matrix.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
RotateX	float	

ROTY - Rotate about the Y-Axis

Shorthand chunk for the specification of a y-axis rotation-only transformation matrix.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
RotateY	float	

ROTZ - Rotate about the Z-Axis

Shorthand chunk for the specification of a z-axis rotation-only transformation matrix.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
RotateZ	float	

ROTQ - Rotate about Quarternion Defined Axis

The rotation component of a transformation matrix.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
RotateX	float	
RotateY	float	
RotateZ	float	

A quarternion of the form (s,(x,y,z)) can be converted to a homogenous matrix with the following formulae:

$$\begin{pmatrix} 1-2y^2-2z^2 & 2xy-2sz & 2xz+2sy & 0 \\ 2xy+2sz & 1-2x^2-2z^2 & 2yz-2sx & 0 \\ 2xz-2sy & 2yz+2sx & 1-2x^2-2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

VERT - Vertex

A vertex defines a point in space. It's location is specified in local coordinates; it's orientation is inherited from its parent.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.

Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
X	float	
Y	float	
Z	float	

EDGV - Edge (Value Specified)

Two points specified with discrete locations

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
X0	float	
Y0	float	
Z0	float	
X1	float	
Y1	float	
Z1	float	

EDGR - Edge (Reference Specified)

Two points specified by reference to points already transformed to global space.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
P0	string	Names of points to inherit from.
P1	string	

TRIV - Triangle (Value Specified)

Three points specified with discrete locations

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry.

		If NULL, then parent is global space.
X0	float	
Y0	float	
Z0	float	
X1	float	
Y1	float	
Z1	float	
X2	float	
Y2	float	
Z2	float	

TRIR - Triangle (Reference Specified)

Three points specified by reference to points already transformed to global space.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
P0	string	Names of points to inherit from.
P1	string	
P2	string	

QADV - Quadrilateral (Value Specified)

Four points specified with discrete locations

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
X0	float	
Y0	float	
Z0	float	
X1	float	
Y1	float	
Z1	float	
X2	float	
Y2	float	
Z2	float	
X3	float	
Y3	float	

Z3	float

QADR - Quadrilateral (Reference Specified)

Four points specified by reference to points already transformed to global space.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Names of points to inherit from.
P1	string	
P2	string	
P3	string	

PLYV - Polygon (General Case, Value Specified)

An n-gon of at least 1 point, specified with discrete value. It is a superset capable containing vertices, edges, triangles, and quadrilaterals, any of which may be specified here.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
N	float	Number of points
X0	float	
Y0	float	
Z0	float	
…	…	List points as necessary

PLYR - Polygon (General Case, Reference Specified)

An n-gon of at least 1 point, specified with referenced values. It is a superset capable containing vertices, edges, triangles, and quadrilaterals, any of which may be specified here.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.

N	float	Number of points
P0	string	Names of points to inherit from
...	...	List points as necessary

PLXV - Polygon (Convex Case, Value Specified)

A special case polygon which is entirely convex. The differentiation in chunks is to serve as a hint for optimal processing of the data.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
N	float	Number of points
X0	float	
Y0	float	
Z0	float	
...	...	List points as necessary

PLXR - Polygon (Convex Case, Reference Specified)

A special case polygon which is entirely convex. The differentiation in chunks is to serve as a hint for optimal processing of the data.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
N	float	Number of points
P0	string	Names of points to inherit from
...	...	List points as necessary

INST - Instance of Geometry Data

Declares an instance of data.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
Original	string	Name of original geometry to copy (can be a hierarchy)

Visual Chunk Types

The following chunk types define the visual properties of the scene.

COLR - Color

Defines a color in 16-bit *per channel* form, including alpha. Since most systems are not using 16-bits per channel, the data for lower bit depths should be aligned to the most-significant-bit of the data space. Additionally, IFF specifies the repetition of data bits to the lower order unused space. For example,

bit-depth	value (hex)	store as (hex)
4	F	FFFF
8	4F	4F4F
12	34F	34F3

Name	string	Unique identification string for visual data.
Tag	string	User defined information.
Red	Uint16	Red Channel
Green	Uint16	Green Channel
Blue	Uint16	Blue Channel
Alpha	Uint16	Alpha Channel (transparency) – 0 is transparent, FFFF is opaque

PALT - Palette

Defines a series of colors in 16-bit per channel form, including alpha. The same conventions regarding bit-use apply as in the COLR chunk.

Name	string	Unique identification string for visual data.
Tag	string	User defined information.

N	Uint16	Number of colors in this palette
Red0	Uint16	
Green0	Uint16	
Blue0	Uint16	
Alpha0	Uint16	
...	…	List colors as necessary

TXTP - Texture using Palette Data

Defines a texture map using palette data. Data is listed row-by-row from left-to-right. Each pixel gets is own byte (if 256 or fewer colors are used) or 16-bit word (if more than 256 are used). Palette indices are enumerated beginning with zero.

Name	string	Unique identification string for visual data.
Tag	string	User defined information.
Palette	string	Name of the palette to use
Offset	Uint16	Offset within palette to starting color
Size	Uint16	Number of colors assumed to be in palette
Width	Uint16	in Pixels
Height	Uint16	in Pixels
data	Uint8 | Uint16	Row-by-row dump of palette indices.
...	…	List entries as necessary

TXTR - Texture using Raw Data

Indicates a texture map using raw data.

Supported are 8-bit per channel and 16-bit per channel data, including alpha. Color entries follow the same rules as in the COLR chunk. Data is stored row-by-row, left-to-right.

Name	string	Unique identification string for visual data.
Tag	string	User defined information.

Depth	Uint16	Color depth per channel.
Width	Uint16	in Pixels
Height	Uint16	in Pixels
Red00	Uint8 | Uint16	Red channel.
Green00	Uint8 | Uint16	Green channel.
Blue00	Uint8 | Uint16	Blue channel.
Alpha00	Uint8 | Uint16	Alpha channel.
…	…	List entries as necessary

APLY - Apply to Object

Applies visual information to geometry.

In general, any visual information can be applied to any geometry. Some of these are appropriate; other make little sense. Specified results are as follows:

- COLR applied to <any> ‡ <any> is solid-colored with specified color.
- PALT applied to <any> ‡ <any>'s points are colored with a series of colors from the palette. The indices of the palettes are specified by the U coordinate of each point. This is to allow for Gouraud shading.
- TXT* applied to <any> ‡ <any>'s surface area is mapped with the specified texture data linearly mapped in 3-space.

A visual chunk can be applied to multiple pieces of geometry. A piece of geometry can have multiple visual chunks applied to it. If an object has multiple visual attributes, they are assumed to mix equally.

Textures are specified with UV coordinates. As many UV coordinates as the target geometry has are specified (1 for VERT, 2 for EDG*, 3 for TRI*, 4 for QAD*, N for PL*).

Name	string	Unique identification string for this chunk
Tag	string	User defined information.
Visual	string	Name of visual chunk to apply
Geometry	string	Name of geometry chunk to apply to.
Points	Uint32	Number of points expected in target geometry. (32 bits are used to maintain alignment)

U0	float	U coordinate for first point
V0	float	V coordinate for first point
…	…	List UV entries as necessary

GRUP - Declare a Group of Chunks

Identifies a set of chunks as a named set. The named set could be used for group selection, or operation. The primary intent of this feature is to allow a group of visual information to be applied to a group of geometry information.

There are many situations where this feature is undesirable – as in the parent of a geometry item. As a result this chunk is only legally named as part of the APLY chunk, or by itself.

Name	string	Unique identification string of this group.
Tag	string	User defined information.
N	Uint32	Number of members in this group
Member0	string	Name of members to be included
…	string	List members as necessary

Animation Chunk Types

ANML - Animate Linear

Straight linear interpolated animation (or key-framed data)

Name	string	Unique identification string for this chunk
Tag	string	User defined information.
Points	Uint16	Number of control points in function.
Time0	float	Time of first control point.
Param0	float	Value at first control point.
…	…	List Time/Parameter pairs as necessary.

ANIM - Animate Parameter

Applies animation function to an arbitrary parameter. Due to the nature of variable length records, the context of the target needs to be understood for the animation data to be correctly placed and converted to integer if necessary.

Name	string	Unique identification string for this chunk
Tag	string	User defined information.
Anim	string	Name of animation function to apply
Target	string	Name of chunk to apply to.
Param	Uint32	Parameter index of animated value (staring with zero for 'Name')

VALU - Value

This chunk is a container for a single value. It is used for evaluation of animation.

Name	string	Unique identification string for this chunk
Tag	string	User defined information.
Param	float	The value

ASGN - Assign Value to Parameter

Copies a value to an arbitrary parameter. Due to the nature of variable length records, the context of the target needs to be understood for the animation data to be correctly placed and converted to integer if necessary.

Name	string	Unique identification string for this chunk
Tag	string	User defined information.
Value	string	Name of value to copy
Target	string	Name of chunk to apply to.
Param	Uint32	Parameter index of animated value (staring with zero for 'Name')

ARTH - Arithmetic function, simple

Performs simple three component arithmetic (two sources, one destination). It operates exclusively on VALU chunks. The hierarchy exists to force ordering when necessary – processing is expected to be depth-first.

Name	string	Unique identification string for this chunk
Tag	string	User defined information.
Parent	string	Identification string of parent arithmetic. If NULL, then function is top-level.
Oper	string	Function identification
A	string	Name of source value 1
B	string	Name of source value 2
Result	string	Name of result value

Oper	**Result**
+	A+B
-	A-B
*	A*B
/	A/B
%	A%B
abs	as defined by "C" language
acos	as defined by "C" language
asin	as defined by "C" language
atan	as defined by "C" language
cos	as defined by "C" language
cosh	as defined by "C" language
exp	as defined by "C" language
log	as defined by "C" language
log10	as defined by "C" language
pow	as defined by "C" language
sin	as defined by "C" language
sinh	as defined by "C" language
tan	as defined by "C" language
tanh	as defined by "C" language
sqrt	as defined by "C" language

Order of Animation evaluation

Order of evaluation of animation data is as follows:

- determination of Time
- ANIM/ANML chunks
- ARTH chunks
- ASGN chunks

- evaluation of Geometry

Miscellaneous Chunk Types

LTAM - Light, Ambient

Defines an ambient light source

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
R	float	Color of light (1.0 being full intensity)
G	float	
B	float	

LTPT - Light, Point

Defines a point light in space. It's location is specified in local coordinates; it's orientation is inherited from its parent.

Name	string	Unique identification string of this light.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
R	float	Color of light (1.0 being full intensity)
G	float	
B	float	
X	float	Coordinate location
Y	float	
Z	float	

LTDR - Light, Directional

Defines a directional light in space. It's location is specified in local coordinates; it's orientation is inherited from its parent.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.

Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
R	float	Color of light (1.0 being full intensity)
G	float	
B	float	
X	float	Coordinate location
Y	float	
Z	float	

CAMR - Camera

Defines a simple pinpoint camera in space. It's location is specified in local coordinates; it's orientation is inherited from its parent.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry. If NULL, then parent is global space.
Dist	float	distance to view plane
X	float	Coordinate location
Y	float	
Z	float	

USR* - Application defined usage

Any chunk beginning with "USR" is assumed to be application defined. Though not official sanctioned as a chunk, it is included for application writers who are unable to or do not have time to register their chunk usage and definition, and for development stage work.

NORM - Defines a Surface Normal

Define a surface normal which can be attached to a piece of geometry. The parent is defined at having this normal. This chunk is not considered a core geometry type since normals are very application specific. Therefore, specification and usage of these normals is application defined. Applications that use this data should have an option to ignore this data.

Name	string	Unique identification string of this piece of geometry.
Tag	string	User defined information.
Parent	string	Identification string of parent geometry.

		If NULL, then parent is global space.
NX	float	Surface normal value
NY	float	
NZ	float	