The Infini-D™ 3.0 File Format

Specular R&D
October 10, 1995

Contents:

Contact information for updates and tech. support:

Chris Johnston
Director of 3D Products
Specular International
479 West St.
Amherst, MA, 01002

e-mail: chrisj@specular.com
Tel: (413) 253-3100
WWW: http://www.specular.com
AOL: keyword SPECULAR

1. Introduction

This document describes how to create, interpret, and modify Infini-D 3.0 and 3.1 scene files and object library files. A scene file describes either a static 3D scene or a 3D animation, and includes parameters for geometric objects, lights, cameras, surfaces, effects, and time-based transformation parameters. An object library file describes the geometry of a single SplineForm™ object, which describes a path, 4 rails, and 1 or more cross-sections to form a cubic Bezier-patch surface.

The Infini-D file format is not for casual consumption. We give a simple outline the Infini-D file format, and quickly refer you to the example programs where the real learning begins. Serious readers should be equipped with the following tools:

a) A Macintosh loaded with a licensed copy of Infini-D (to make example scene files, and test your own files, and understand the file parameters).
b) E-mail service and a telephone to contact us (see 1st page for contact information).
c) A "C" or "C++" compiler, and the example programs.
d) Resorcerer™ and the supplied "Elmo Templates" file for low-level examining and editing of files, or any other binary/hex file editor if not available. Resorcerer is a resource editor from Mathemæsthetics, Inc., PO Box 298, Boulder CO 80306-0298, Phone (303) 440 0707, Internet: resorcerer@aol.com.
e) Experience and/or good reference books on 3D graphics and animation, to understand the various file parameters.

2. Using the Examples and Templates.

The example C code is provides a basic structure for reading and writing Infini-D scene files. One example program is supplied, called "InfiniWriter." It first writes out a minimal scene file: 2 surfaces; 1 camera, 1 light, and 1 mesh object; the 6 standard views, and the 7 default bookmarks. It then reads the very same scene file and displays a basic listing of the blocks encountered. The code is intended to be a simple shell which can be extended by adding objects, surfaces, events, etc. It has a minimal I/O shell which uses the ANSI C Library for generality, plus a few Macintosh Toolbox calls were absolutely necessary. You'll certainly want to rewrite the I/O routines for better error handling and efficiency if you use the example code in a commercial product. The supplied source files come with a Think C version 7 project file, and have been tested in that environment.

If you are a user of the Resorcerer™ resource editor, you can examine Infini-D files using the supplied data-fork templates. Resorcerer with the supplied templates is highly recommended to ease debugging and understanding of scene file contents. To install the templates, copy the file "Elmo Templates" to the "Private Templates" folder inside the Resorcerer application folder and restart Resorcerer.

The <DF> template is for editing the data fork of a scene or object file. If the template conflicts with another <DF> template, open "Elmo Templates" and delete the TMPL resource named <DF>. The data fork can also be edited by changing the type of the '<DF>' resource to 'Elmo' temporarily; use the Info dialog.

The ElmB template is for viewing the block structure of Infini-D or other Elmo-based files. The block content is displayed in hex. You can use this template by temporarily changing a '<DF>' resource to 'ElmB' using the Info dialog.

3. The Elmo block structure.

Infini-D versions 3.0 and later use the Elmo file format. Elmo is a nested block structure which brings order to an otherwise chaotic collection of data. Elmo's main features are:

a) Each file is a list of 1 or more blocks.
b) Each block has a list of 0 or more subblocks after the main block data.
c) Each block has a header containing the block type, a unique block ID (the Tag), the block size, and the position of the first subblock.
d) Blocks have a fixed or variable size depending on the block type.
e) Any block can have extra data added by adding a subblock of a new type.
f) Any unrecognized block types are ignored, to allow backward compatibility.
g) Blocks are written in a standard order for efficiency of reading.
h) File readers can take advantage of the expected order for efficiency, but should not expect blocks in a particular order to allow for new and unrecognized block types.
i) Integers and other values are ordered with the most-significant byte first (big-endian), as is typical with Mactinosh data.

A Single Elmo Block

Offset	Size in Bytes	Contents	Description
0	4	Elmo type	a 4-character code which determines the block contents, e.g. "scen" for scene files.
4	4	Elmo tag	unique identifying number for each block in a file.
8	4	block size	block size in bytes, including header and subblocks.
12	4	subblock offset	subblock offset: the number of bytes from the start of the block to the first subblock. If there are no subblocks then the offset equals the block size.
16	varies	block data	The contents here depends on the block type, and extends from byte 16 to (subblock_offset - 1)
subblock_offset	varies	1st subblock	(subblocks are optional, according to block type)
subblock_offset + 1st subblock block size	varies	2nd subblock	(subblocks are optional, according to block type)
subblock_offset + 2nd subblock block size	varies	3nd subblock	(subblocks are optional, according to block type)
...	...	...	...
block size			start of next block

4. Scene file structure.

Blocks types are listed in the order they usually appear in a scene file. Each level of indentation below indicates that the block type is a subblock of the block above.

```
elmo    -------file header
        scen    -------one scene block
        csrf    -------one block per composite surface, in a mixed list of basic and composite surfaces.
                csla    -------one block per composite surface layer
        surf    -------one block per basic surface, in a mixed list of basic and composite surfaces.  Basic surfaces have 1 or 2
                          surface mapping subblocks:
                -------one image mapping subblock from list below.
                rgb     -------RGB color parameter.
                tile    -------tiling parameters.
                frct    -------fractal parameters.
                nois    -------noise parameters.
                marb    -------marble parameters.
                wood    -------(old style) wood parameters.
                natw    -------natural wood parameters.
                imag    -------image map parameters.
                        alis    -------one file alias block.
                -------optionally one bump maping subblock from list below.
                nois    -------noise parameters.
                wave    -------wave parameters.
        lite    -------one per light
        obj     -------one per object, including cameras and lights, in a tree structure.
                terr    -------one terrain block for terrain objects
                        frct    -------one fractal data block for fractal terrains.
                        imag    -------one image block for image mapped terrains.
                                alis    -------one file alias block
                modl    -------one model block for polygonal mesh objects.
                        verl    -------one vertex list.
                        edgl    -------one edge list.
                        facl    -------one face list.
                                indl    -------multiple edge or neighbor index lists blocks.
                evtm    -------one for each eventmark of object. Event marks have different sets of parameters stored with
                                  the subblocks below:
                        afev    -------affine event data.
                        raev    -------rotation Affine event data.
                        caev    -------camera event data.
                        liev    -------light event data.
                        sfev    -------surface event data.
                        ppev    -------path-profile event data.
                                pmdl    -------one path model data block.
                                        ppro    -------one path-profile data block.
                                                ol3d    -------5 outline 3D data blocks (path & 4 rails).
                                                pf2d    -------one profile2D block for each cross-section.
                                                        ol2d    -------one outline2D block for each curve in a
                                                                                                  cross-section.
                        ptev    -------path-profile text event data.
                                pmdl    -------one path model data block (see above for subblocks)
                        trev    -------terrain event data.
                                frct    -------Fractal parameters.
        envv    -------one environment variables block.
        seqv    -------one sequence variables block.
        bkmk    -------one per bookmark, forming a linked list of blocks.
        view    -------one per view, forming a linked list of blocks.
        outl    -------one per outline object, forming a link list of blocks.
                pypt    -------5 polypoint lists per outline object.
```

```
                    pylp    -------one or more Polypoint Loop blocks per Polypoint block.
        sqin        -------one sequencer info block.
                    sqob    -------one per object listed in sequencer.
        end!        -------one EOF block.
```

5. Object Library file structure.

```
elmo    -------file header
        pmdl    -------one path model data block
                ppro    -------one path-profile data block
                        ol3d    -------5 outline 3D data blocks (path & 4 rails)
                        pf2d    -------one profile2D block for each cross-section
                                ol2d    -------one outline2D block for each curve in a cross-section
        end!    -------one EOF block
```

6. Block Type Summary

Infini-D 3.0 Block Types - Grouped and roughly in order of occurrence.

Type	Name	Parent	Child	Subblock offset	Page
elmo	File header	-	(many)	28	12
scen	Scene	elmo	surf	48	12
csrf	Composite surface	elmo	csla	100	13
csla	Composite surface layer	csrf	-	156	14
surf	Basic surface	elmo	rgb, tile, frct, nois, marb, wood, natw, imag	104	15
rgb	RGB color	surf	-	28	16
tile	Tile surface	surf	-	192	17
frct	Fractal surface	surf, terr	-	72	17
nois	Noise surface	surf	-	48	18
marb	Marble surface	surf	-	64	18
wood	Wood surface	surf	-	56	18
natw	Natural wood surface	surf	-	64	19
wave	Wave surface	surf	-	64	19
imag	Image map	surf, terr	alis	80	19
alis	File Alias	imag	-	varies	20
lite	Light	elmo	-	124	21
obj	Object	elmo	evtm	236	22
terr	Terrain data	obj	imag, frct	28	23
modl	Polygonal Mesh data	obj	verl, edgl, facl		24
verl	Vertex list	modl	-	varies	25
edgl	Edge list	modl	-	varies	25
facl	Face list	modl	indl	varies	25
indl	Index list	facl	-	varies	26
evtm	Event mark	obj	afev, raev, sfev, olev, txev, ptev, liev, caev, trev, ppev	varies	26
afev	Affine event mark	evtm	-	72	27
raev	Rotation Affine event mark	evtm	-	88	28
sfev	Surface event mark	evtm	-	28	29
olev	Outline event mark	evtm	-	28	29
txev	Text object event mark	evtm	-	varies	30
liev	Light object event mark	evtm	-	72	30
caev	Camera object event mark	evtm	-	28	31
trev	Terrain object event mark	evtm	frct	28	31
ppev	Path-profile obj. event mark	evtm	pmdl	28	32
ptev	Path-Profile text obj. event mark	evtm	pmdl	varies	32
pmdl	Path Model	ppev, ptev	ppro	40	33
ppro	Path Profile	pmdl	ol3d, pf2d	varies	33
pf2d	Cross Section (Profile 2D)	ppro	ol2d	varies	35
ol3d	3D Spline (Outline 3D)	ppro	-	varies	35
ol2d	2D Spline (Outline 2D)	pf2d	-	varies	36
envv	Environment Variables	elmo	-	64	37
seqv	Sequence Variables	elmo	-	64	37
bkmk	Bookmark	elmo	-	96	38
view	View of scene	elmo	-	196	39
outl	Outline Model (5-way Polyline)	elmo	pypt	58	41
pypt	Polypoint	outl	pylp	varies	42
pylp	Polypoint Loop	pypt	-	varies	42
sqin	Sequencer Info	elmo	sqob	20	43
sqob	Sequencer Object data	sqin	-	32	43
end!	End of file	elmo	-	16	44

Infini-D 3.0 Block Types - Alphabetical

Type	Name	Parent	Child	Subblock offset	Page
afev	Affine event mark	evtm	-	72	27
alis	File Alias	imag	-	varies	20
bkmk	Bookmark	elmo	-	96	38
caev	Camera object event mark	evtm	-	28	31
csla	Composite surface layer	csrf	-	156	14
csrf	Composite surface	elmo	csla	100	13
edgl	Edge list	modl	-	varies	25
elmo	File header	-	(many)	28	12
end!	End of file	elmo	-	16	44
envv	Environment Variables	elmo	-	64	37
evtm	Event mark	obj	afev, raev, sfev, olev, txev, ptev, liev, caev, trev, ppev	varies	26
facl	Face list	modl	indl	varies	25
frct	Fractal surface	surf, terr	-	72	17
imag	Image map	surf, terr	alis	80	19
indl	Index list	facl	-	varies	26
liev	Light object event mark	evtm	-	72	30
lite	Light	elmo	-	124	21
marb	Marble surface	surf	-	64	18
modl	Polygonal Mesh data	obj	verl, edgl, facl		24
natw	Natural wood surface	surf	-	64	19
nois	Noise surface	surf	-	48	18
obj	Object	elmo	evtm	236	22
ol2d	2D Spline (Outline 2D)	pf2d	-	varies	36
ol3d	3D Spline (Outline 3D)	ppro	-	varies	35
olev	Outline event mark	evtm	-	28	29
outl	Outline Model (5-way Polyline)	elmo	pypt	58	41
pf2d	Cross Section (Profile 2D)	ppro	ol2d	varies	35
pmdl	Path Model	ppev, ptev	ppro	40	33
ppev	Path-profile obj. event mark	evtm	pmdl	28	32
ppro	Path Profile	pmdl	ol3d, pf2d	varies	33
ptev	Path-Profile text obj. event mark	evtm	pmdl	varies	32
pylp	Polypoint Loop	pypt	-	varies	42
pypt	Polypoint	outl	pylp	varies	42
raev	Rotation Affine event mark	evtm	-	88	28
rgb	RGB color	surf	-	28	16
scen	Scene	elmo	surf	48	12
seqv	Sequence Variables	elmo	-	64	37
sfev	Surface event mark	evtm	-	28	29
sqin	Sequencer Info	elmo	sqob	20	43
sqob	Sequencer Object data	sqin	-	32	43
surf	Basic surface	elmo	rgb, tile, frct, nois, marb, wood, natw, imag	104	15
terr	Terrain data	obj	imag, frct	28	23
tile	Tile surface	surf	-	192	17
trev	Terrain object event mark	evtm	frct	28	31
txev	Text object event mark	evtm	-	varies	30
verl	Vertex list	modl	-	varies	25
view	View of scene	elmo	-	196	39
wave	Wave surface	surf	-	64	19
wood	Wood surface	surf	-	56	18

7. Data Types.

Integers are ordered with the most-significant byte first (big-endian), as is typical with Mactinosh data. All floating-point values are in standard IEEE format.

General Types		
Name	*Bytes*	*Description*
ElmoUInt8	1	8 bit unsigned integer
ElmoUInt16	2	16 bit unsigned integer
ElmoUInt32	4	32 bit unsigned integer
ElmoInt8	1	8 bit signed integer
ElmoInt16	2	16 bit signed integer
ElmoInt32	4	32 bit signed integer
ElmoFloat32	4	32 bit floating point number (IEEE single precision)
ElmoBoolean	1	8 bit boolean value. 1 = TRUE, 0 = FALSE.
ElmoPString	varies	A Macintosh “Pascal” string, where the first byte is the number of characters (0-255) in the string that follows. There is no terminating character. Note that the maximum string length is one less than the number of bytes available.

Block header types		
Name	*Bytes*	*Description*
ElmoType	4	4-character code, e.g. 'elmo'. Characters must be in the range 32-216 ($20-$D8).
ElmoTag	4	An identifying number unique to each block in an elmo file. A 32 bit unsigned value. Reserved block numbers are: kNotAnElmoTag = 0 — No block or end of block list. kElmoEOFTag = -1 — Tag of “end!” block. Elmo file header = 1 — Tag of file header block.
ElmoBlockHeader	16	The standard header for each Elmo block: 4 ElmoType — a 4-character code which determines the block contents, e.g. "scen" for scene files. Characters must be in the range 32-2216 ($20-$D8). 4 ElmoTag — unique identifying number for each block in a file. 4 ElmoUInt32 — block size in bytes, including header and subblocks. 4 ElmoUInt32 — subblock offset: the number of bytes from the start of the block to the first subblock. If there are no subblocks then the offset equals the block size.

Geometric types		
Name	*Bytes*	*Description*
ElmoPoint2D	8	A 2D point (ElmoFloat32: x, y)
ElmoIntPoint2D	4	A 2D Macintosh-style point (ElmoInt16: x, y)
ElmoPointST	8	A point for texture mapping (ElmoFloat32: s, t)
ElmoPoint3D	12	a 3D point or 3D vector (x, y, z)
ElmoPoint4D	16	a 4D point or 4D vector (x, y, z, w)
ElmoQuaternion	16	a quaternion (a useful way to represent rotation: c, x, y, z)
ElmoRect	16	a rectangle (left, top, right, bottom)
ElmoIntRect	8	an integer, Macintosh-style rect (top, left, bottom, right)

Color types				
Name	*Bytes*	*Description*		
ElmoRGBIntColor	6	A Macintosh-style RGB color		
		2 ElmoUInt16	red	
		2 ElmoUInt16	green	
		2 ElmoUInt16	blue	
ElmoRGBColor	12	an RGB color (each component ranges from 0.0..1.0)		
		4 ElmoFloat32	red	
		4 ElmoFloat32	green	
		4 ElmoFloat32	blue	
ElmoHSVColor	12	an HSV color (each component ranges from 0.0..1.0)		
		4 ElmoFloat32	hue	
		4 ElmoFloat32	saturation	
		4 ElmoFloat32	value	
ElmoColorTransition	28	an RGB or HSV transition from one color to another		
		2 ElmoUInt16	padding (reserved, set to 0)	
		2 ElmoUInt16	transition type (ElmoRGBTransition = 0, ElmoHSVTransition = 1)	
		12 ElmoRGBColor / ElmoHSVColor	start	
		12 ElmoRGBColor / ElmoHSVColor	end	
ElmoAlphaMode	1	What kind of alpha channel for an image?		
		kElmoNoAlpha = 0		
		kElmoStraightAlpha = 1		
		kElmoMultipliedAlpha = 2		

Object transform types				
Name	*Bytes*	*Description*		
ElmoAffine	72	An object's affine data.		
		12 ElmoVector3D	scale	local scale; does not affect children
		12 ElmoVector3D	offset	local offset; does not affect children
		12 ElmoVector3D	tree_scale	“uniform scale”; passed down to children
		12 ElmoVector3D	rotation	passed down to children
		12 ElmoVector3D	shear	passed down to children
		12 ElmoVector3D	position	passed down to children
ElmoConstraint3D	28	An object's constraint information (rotation, scale, or position).		
		12 ElmoVector3D	min	
		12 ElmoVector3D	max	
		1 ElmoBoolean	xLocked	
		1 ElmoBoolean	yLocked	
		1 ElmoBoolean	zLocked	
		1 ElmoBoolean	reserved	

Bezier spline types				
Name	*Bytes*	*Description*		
PtNode2D_Elmo_Struct	28	A point node on a 2D cubic Bezier spline.		
		2 ElmoUInt16	reserved	
		1 ElmoUInt8	reserved	
		1 ElmoUInt8	kind	kPlainKind=0
				kCornerKind=1
				kCurveKind=2
				kSmoothKind=3
		8 ElmoPoint2D	position	
		8 ElmoPoint2D	LControl	
		8 ElmoPoint2D	RControl	
PtNode3D_Elmo_Struct	40	A point node on a 3D cubic Bezier spline.		
		2 ElmoUInt16	reserved	
		1 ElmoUInt8	reserved	
		1 ElmoUInt8	kind	kPlainKind=0
				kCornerKind=1
				kCurveKind=2
				kSmoothKind=3
		12 ElmoPoint3D	position	
		12 ElmoPoint3D	LControl	
		12 ElmoPoint3D	RControl	

Polygonal Model types				
Name	*Bytes*	*Description*		
ElmoModelIndex	4	An index into a Polygonal model's face, edge, or vertex list (an unsigned integer)		
ElmoEdge	8	An edge of a polygonal mesh model.		
		4 ElmoModelIndex	index1	Index into vertex list; vertex has coordinates of 1st endpoint.
		4 ElmoModelIndex	index2	Index into vertex list; vertex has coordinates of 2nd endpoint.
ElmoFace	40	A "face", a 3 or more sided 3D polygon; a facet of a polygonal mesh model.		
		4 ElmoUInt16	flags	kElmoFlatFace = 0 (face is planar) kElmoInterpFace = 1 (face not planar)
		4 ElmoModelIndex	edgeCount	Number of edges
		16 ElmoIndexUnion	edgeList	4 edge indexes or Tag of (edge) index list, in CCW order.
		16 ElmoIndexUnion	neighborList	4 neighboring face indexes or Tag of (neighboring face) index list, in CCW order and corresponding to the above edgeList.
ElmoIndexUnion	16	A list of up to 4 indices *OR* the tag of a subblock with a list of 5 or more indicies. Used for a face's edge and neighbor lists - since most faces have 3 or 4 edges and 3 or 4 neighbors, we optimize for that case. Edges and Faces must be listed in counter-clockwise order (CCW from the ouside perspective). If a face does not exist, or smoothing towards the neighboring face is not desired, the constant K_no_neighboring_face is used. K_no_neighboring_face is 0xFFFFFFFF for version 3.1 or later, 0xFFFF for version 3.0 or 3.0.1.		
		(if ElmoFace.edgeCount is 4 or less)		
		4 ElmoModelIndex	index[0]	Index of the first edge or neighbor
		4 ElmoModelIndex	index[1]	Index of the second edge or neighbor
		4 ElmoModelIndex	index[2]	Index of the third edge or neighbor
		4 ElmoModelIndex	index[3]	Index of the fourth edge or neighbor
		(or, if ElmoFace.edgeCount is 5 or more)		
		4 ElmoTag	indexListTag	Tag of Index List subblock (holds edge list or face list).

Outline Model types				
Name	*Bytes*	*Description*		
Elmo_Polypoint_Node	8	A node of a polypoint loop		
		4 ElmoPoint2D	pt	
		2 ElmoUInt16	padding	
		2 ElmoInt16	mode	K_straight_segment = 0 K_interpolated_segment = 1 K_bezier_segment = 2 (cubic spline) K_bezier_2_segment = 3 (quadratic spline)