

AlphaMask Effects Classes

Copyright 1995-2000 by AlphaMask, Inc. All rights reserved.

August 2, 2000

Contents

1	Base Classes	3
1.1	hsGBitmapShader	3
1.2	hsGImageFilter — A <i>hsGImageFilter</i> is attached to an <i>hsGAttributes</i> via <i>hsGAttribute::SetImageFilter</i>	4
1.3	hsGMaskFilter — For special effects such as blurring or embossing, the client may provide a subclass of <i>hsGMaskFilter</i>	4
1.4	hsGPathEffect — Clients may modify the geometry at draw time by providing a subclass of <i>hsGPathEffect</i>	5
1.4.1	Record	6
1.5	hsGRasterizer — Clients may also override the scan conversion process by providing a subclass of <i>hsGRasterizer</i>	6
1.6	hsGShader — <i>hsGShader</i> is a client-specified object that supplies per-pixel colors	7
1.6.4	TileMode — For subclasses that support tiling	8
2	Bitmap Shader	9
2.1	hsGBumpShader	9
3	Image Filters	10
3.1	hsGSparkleImageFilter	10
4	Mask Filters	11
4.1	hsGBlurMaskFilter	11
4.2	hsGEmbossMaskFilter	11
4.3	hsGEmbossRecord	12
4.4	hsGScatterMaskFilter	12
4.5	hsGShaderMaskFilter	13
5	Path Effects	14
5.1	hsGCrackPathEffect	14
5.2	hsGDashPathEffect	14
5.3	hsGWobbleEffectRecord	15
5.4	hsGWobblePathEffect	15
6	Rasterizer Effects	16
6.1	hsGBloaterizer	16
7	Shader Effects	17
7.1	hsGGradientShader	17
7.2	hsGLinearGradientShader	17
7.3	hsGRadialGradientShader	18
7.4	hsGTransparentShader	19
	Class Graph	20

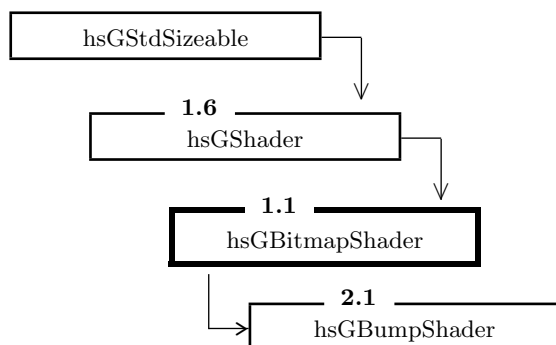
1 Base Classes

Names

1.1	class	hsGBitmapShader : public hsGShader.....	3
1.2	class	hsGImageFilter : public hsGStdSizeable <i>A hsGImageFilter is attached to an hsGAttribute via hsGAttribute::SetImageFilter .</i>	4
1.3	class	hsGMaskFilter : public hsGStdSizeable <i>For special effects such as blurring or embossing, the client may provide a subclass of hsGMaskFilter</i>	4
1.4	class	hsGPathEffect : public hsGStdSizeable <i>Clients may modify the geometry at draw time by providing a subclass of hsGPathEffect</i>	5
1.5	class	hsGRasterizer : public hsGStdSizeable <i>Clients may also override the scan conversion process by providing a subclass of hsGRasterizer</i>	
1.6	class	hsGShader : public hsGStdSizeable ⁶ <i>hsGShader is a client-specified object that supplies per-pixel colors</i>	7

1.1
class **hsGBitmapShader** : public hsGShader

Inheritance



Public Members

void **SetTileMode** (TileMode repeat)
void **SetDeviceCoords** (hsBool b)

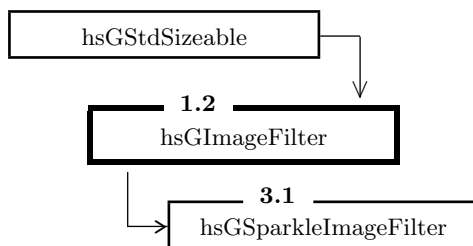
void **SetBitmap** (const hsGBitmap* bitmap, hsBool shaderOwnsPixels = false)

1.2

```
class hsGImageFilter : public hsGStdSizeable
```

A hsGImageFilter is attached to an hsGAttributes via hsGAttribute::SetImageFilter

Inheritance



Public Members

hsBool **FilterBounds** (const hsIntRect* src, hsIntRect* dst)
 hsBool **FilterImage** (const hsGBitmap* src, hsGBitmap* dst)

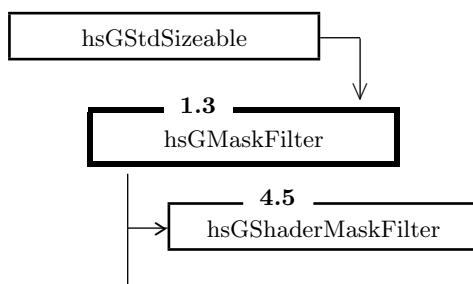
A hsGImageFilter is attached to an hsGAttributes via hsGAttribute::SetImageFilter.

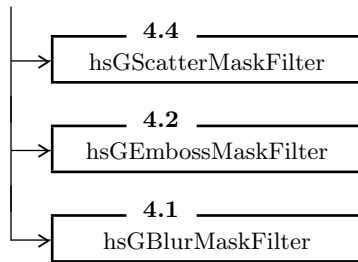
1.3

```
class hsGMaskFilter : public hsGStdSizeable
```

For special effects such as blurring or embossing, the client may provide a subclass of hsGMaskFilter

Inheritance





Public Members

void **Filter** (const hsMatrix33* matrix, const hsGMask* src, hsGMask* dst)

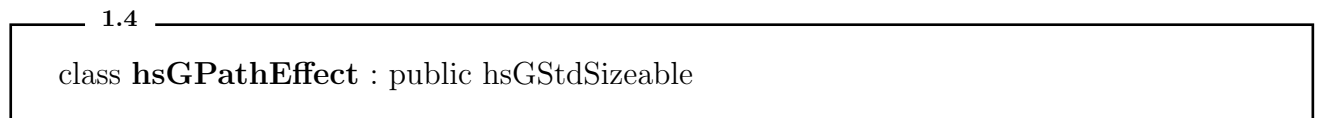
void **FilterBounds** (const hsMatrix33* matrix, hsIntRect* bounds)

void **Write** (hsStream* stream, UInt32 flags = 0)
 For streaming

void **FilterPath** (const RawDraw* rd, const hsPath* path,
 const hsMatrix33* matrix, hsGAttribute* attr)
 *This performs the filter and call the blitter with
 the new mask*

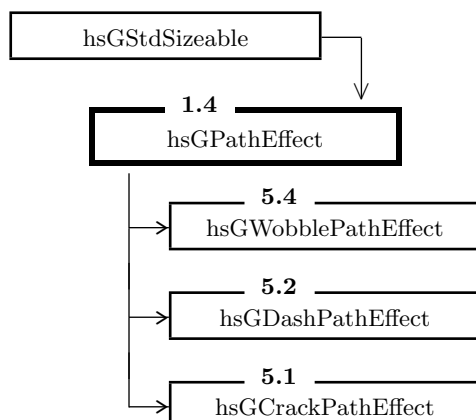
For special effects such as blurring or embossing, the client may provide a subclass of hsGMaskFilter. This object, when present, is called to modify the alpha mask of a drawing primitive.

A hsGMaskFilter is attached to an hsGAttributes via **hsGAttribute::SetMaskFilter**.



Clients may modify the geometry at draw time by providing a subclass of hsGPathEffect

Inheritance



Public Members

1.4.1	class	Record	6
	hsBool	Filter (const hsGPathEffect::Record* input, hsGPathEffect::Record* output)		

Clients may modify the geometry at draw time by providing a subclass of hsGPathEffect. This object is passed the original geometry, and may return a new one.

This class is reference counted.

1.4.1

```
class Record
```

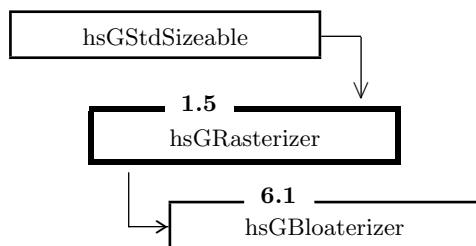
Public Members

UInt32	fFlags
hsScalar	fWidth
hsPath*	fPath

1.5

```
class hsGRasterizer : public hsGStdSizeable
```

Clients may also override the scan conversion process by providing a subclass of hsGRasterizer

Inheritance**Public Members**

void	GetBounds (const hsPath* path, const hsMatrix33* matrix, hsIntRect* bounds)
void	Rasterize (const hsPath* path, const hsMatrix33* matrix, const hsScanRegion* clip, hsGMask* mask)

Clients may also override the scan conversion process by providing a subclass of `hsGRasterizer`. This object is passed a path, and returns an alpha mask.

A `hsGRasterizer` is attached to an `hsGAttribute` via `hsGAttribute::SetRasterizer`.

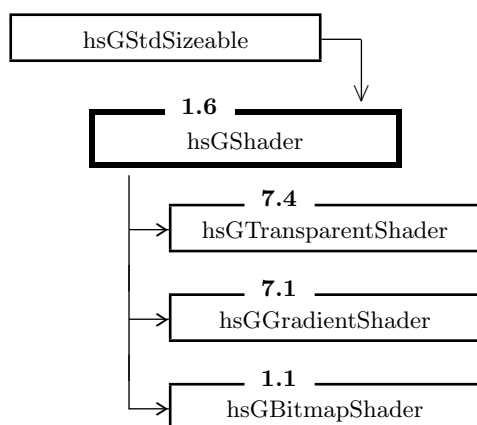
This class is reference counted.

1.6

```
class hsGShader : public hsGStdSizeable
```

hsGShader is a client-specified object that supplies per-pixel colors

Inheritance



Public Members

1.6.1	void	SetContext (const <code>hsGBitmap*</code> device, <code>hsGAttribute*</code> attr, const <code>hsMatrix33*</code> matrix)	<i>This is called before each draw, giving the shader the current attribute and matrix</i>	8
1.6.2	<code>hsBool</code>	IsOpaque ()	<i>This may be called by the blitter</i>	8
1.6.3	void	ShadeSpan (int y, int x, int count, <code>hsColor32</code> src[])	<i>Subclasses must implement this</i>	8
	void	ShadeSpanAlpha (int y, int x, int count, <code>UInt8</code> alpha[])	<i>If this isn't overridden, it calls ShadeSpan.</i>	
	void	ShadeGray4 (int y, int x, int count, <code>hsGAlphaGray44</code> src[])	<i>If this isn't overridden, it calls ShadeSpan.</i>	
	void	SetLocalMatrix (const <code>hsMatrix33*</code> mat)		
	void	GetLocalMatrix (<code>hsMatrix33 *</code> mat) const		
1.6.4		TileMode	<i>For subclasses that support tiling</i>	8

`hsGShader` is a client-specified object that supplies per-pixel colors. It is called for each scanline of the primitive being drawn.

A hsGShader is attached to an hsGAttributes via `hsGAttribute::SetShader`.

1.6.1

```
void SetContext (const hsGBitmap* device, hsGAttribute* attr, const hsMatrix33* matrix)
```

This is called before each draw, giving the shader the current attribute and matrix

This is called before each draw, giving the shader the current attribute and matrix.

1.6.2

```
hsBool IsOpaque ()
```

This may be called by the blitter

This may be called by the blitter. Return `true` if all colors have alpha == 0xFF.

1.6.3

```
void ShadeSpan (int y, int x, int count, hsColor32 src[])
```

Subclasses must implement this

Subclasses must implement this.

This is called by the blitter, where x and y are in device coordinates.

1.6.4

```
TileMode
```

For subclasses that support tiling

Members

```
kClampTile  
kWrapTile  
kMirrorTile  
kDecalTile
```


2

Bitmap Shader

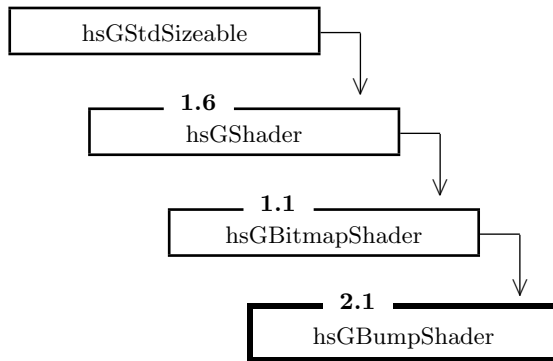
Names

2.1	class	hsGBumpShader : public hsGBitmapShader	9
	void	hsGConvertBitmapToBumpMap (const hsGBitmap* input, hsGBitmap* output, hsScalar mag)	

2.1

```
class hsGBumpShader : public hsGBitmapShader
```

Inheritance



Public Members

	hsGBumpShader ()
void	SetLightingInfo (hsScalar ka, hsScalar x, hsScalar y, hsScalar z) <i>Local Data for lighting stuff</i>

3

Image Filters

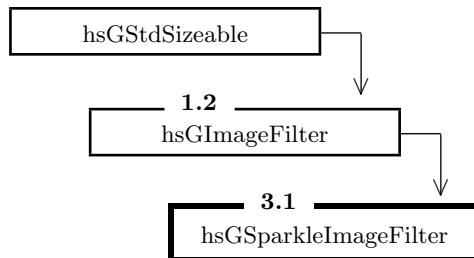
Names

3.1 class **hsGSparkleImageFilter** : public hsGImageFilter..... 10

3.1

class **hsGSparkleImageFilter** : public hsGImageFilter

Inheritance



Public Members

hsGSparkleImageFilter (UInt8 range = 64)

4

Mask Filters

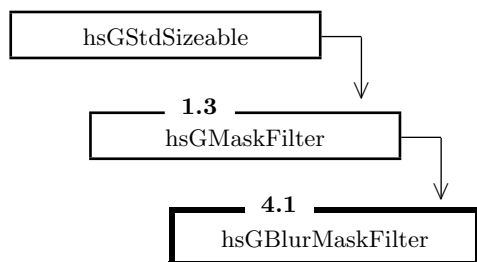
Names

4.1	class	hsGBlurMaskFilter : public hsGMaskFilter	11
4.2	class	hsGEmbossMaskFilter : public hsGMaskFilter	11
4.3		hsGEmbossRecord	12
4.4	class	hsGScatterMaskFilter : public hsGMaskFilter	12
4.5	class	hsGShaderMaskFilter : public hsGMaskFilter	13

4.1

```
class hsGBlurMaskFilter : public hsGMaskFilter
```

Inheritance



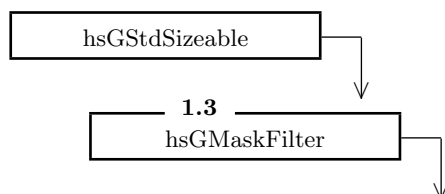
Public Members

```
hsGBlurMaskFilter (hsScalar innerRadius, hsScalar outerRadius,  
hsBool doFill, hsScalar stdSize)
```

4.2

```
class hsGEmbossMaskFilter : public hsGMaskFilter
```

Inheritance



4.2

hsGEmbossMaskFilter

Public Members

hsGEmbossMaskFilter (hsScalar stdSize)

hsGEmbossMaskFilter (const hsGEmbossRecord* rec, hsScalar stdSize)

4.3

hsGEmbossRecord

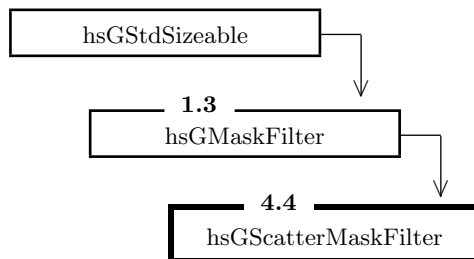
Members

hsScalar	fRadius	<i>amount to emboss</i>
hsScalar	fLight [3]	<i>0, 0, 1 is "regular"</i>
hsScalar	fKs	<i>0..1</i>
hsScalar	fKd	<i>0..1</i>
void	EmbossAlpha (const hsGBitmap* src, hsGBitmap* dst)	

4.4

class **hsGScatterMaskFilter** : public hsGMaskFilter

Inheritance



Public Members

4.4.1	hsGScatterMaskFilter (hsScalar radius, UInt32 flags, hsScalar stdSize)	13
	hsScalar GetRadius () const	
	void SetRadius (hsScalar radius)	

4.4.1

hsGScatterMaskFilter (hsScalar radius, UInt32 flags, hsScalar stdSize)

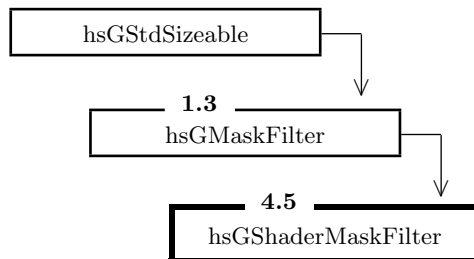
Arguments

kNoGrow_Flag

4.5

class **hsGShaderMaskFilter** : public hsGMaskFilter

Inheritance



Public Members

hsGShaderMaskFilter (hsGShader* shader, hsScalar stdSize)

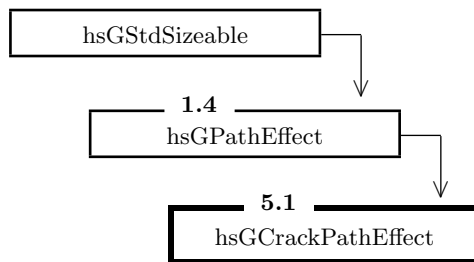
5

Path Effects**Names**

5.1	class	hsGCrackPathEffect : public hsGPathEffect.....	14
5.2	class	hsGDashPathEffect : public hsGPathEffect.....	14
5.3		hsGWobbleEffectRecord	15
5.4	class	hsGWobblePathEffect : public hsGPathEffect.....	15

5.1

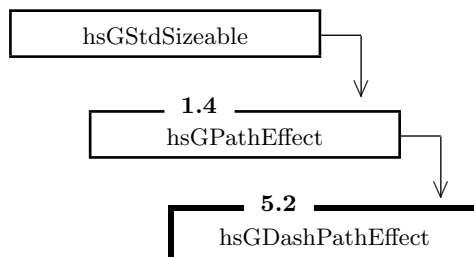
```
class hsGCrackPathEffect : public hsGPathEffect
```

Inheritance**Public Members**

hsGCrackPathEffect (hsScalar period, hsScalar depth, hsScalar gap,
hsScalar rand, hsScalar stdSize)

5.2

```
class hsGDashPathEffect : public hsGPathEffect
```

Inheritance

Public Members

```

hsGDashPathEffect (int count, const hsScalar intervals[],
                    hsScalar stdSize)
    the array is ordered <ON, OFF, ON, OFF, ...>

```

5.3**hsGWobbleEffectRecord****Members**

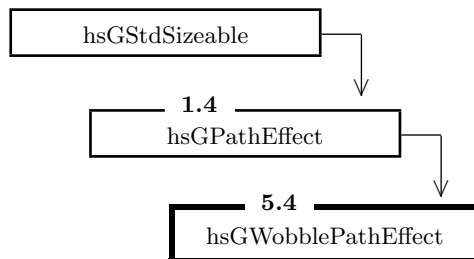
hsScalar	fPeriod
hsScalar	fDeviation
hsScalar	fRandom
hsScalar	fSmooth

5.4

```

class hsGWobblePathEffect : public hsGPathEffect

```

Inheritance**Public Members**

```

hsGWobblePathEffect (const hsGWobbleEffectRecord* record,
                      hsScalar stdSize)

void SetWobble (const hsGWobbleEffectRecord* record)

```

6

Rasterizer Effects

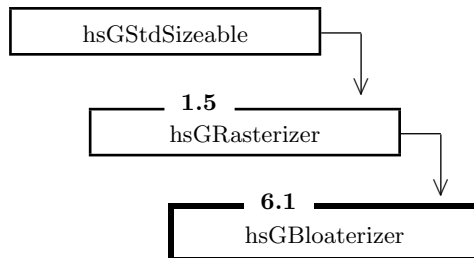
Names

6.1 class **hsGBloaterizer** : public hsGRasterizer 16

6.1

```
class hsGBloaterizer : public hsGRasterizer
```

Inheritance



Public Members

hsGBloaterizer (hsScalar radius, hsGColorValue faceAlpha,
hsGColorValue sideAlpha, hsScalar stdSize)

7 Shader Effects

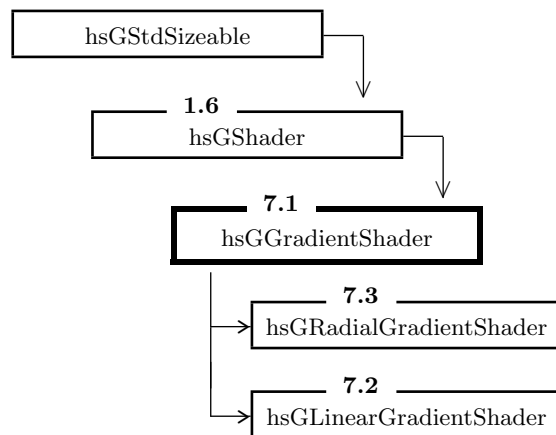
Names

7.1	class	hsGGradientShader : public hsGShader	17
7.2	class	hsGLinearGradientShader : public hsGGradientShader	17
7.3	class	hsGRadialGradientShader : public hsGGradientShader	18
7.4	class	hsGTransparentShader : public hsGShader	19

7.1

```
class hsGGradientShader : public hsGShader
```

Inheritance



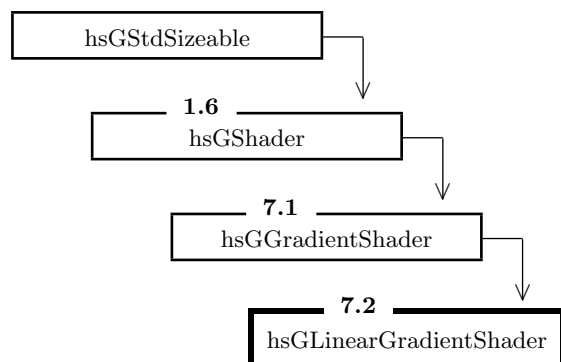
Public Members

	hsGGradientShader (hsScalar stdSize = 0)
int	GetGradient (hsGColor colors[], hsScalar intervals[], TileMode* repeat)
void	SetGradient (int count, const hsGColor colors[], const hsScalar intervals[], TileMode repeat)

7.2

```
class hsGLinearGradientShader : public hsGGradientShader
```

Inheritance



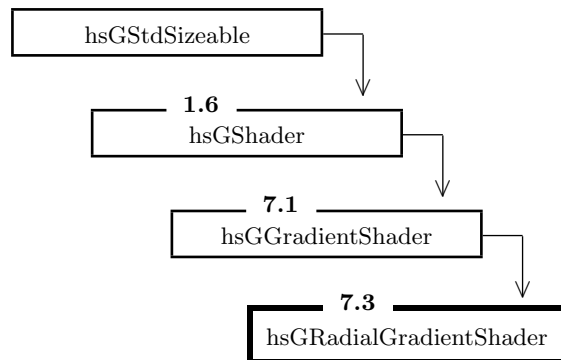
Public Members

void **GetPoints** (hsPoint2* start, hsPoint2* stop) const
 void **SetPoints** (const hsPoint2* start, const hsPoint2* stop)

7.3

```
class hsGRadialGradientShader : public hsGGradientShader
```

Inheritance

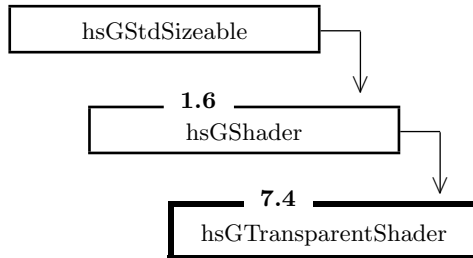


Public Members

hsScalar **GetRadial** (hsPoint2* center) const
 void **SetRadial** (const hsPoint2* center, hsScalar radius)

7.4

```
class hsGTransparentShader : public hsGShader
```

Inheritance**Public Members**

hsGTransparentShader ()

Class Graph

