

# MiscDiagramShape Class Cluster

## Class Cluster Description

The MiscDiagramShape class cluster manages shapes equivalent to the symbols found in Diagram(tm) from Lighthouse Design. The cluster's only public class, MiscDiagramShape , declares the programmatic interface for diagram shapes.

Shapes.eps ↪

The objects you create using this class are referred to as *shape objects*. Because of the nature of class clusters, shape objects are not actual instances of the MiscDiagramShape class but of one of their private subclasses. Although an shape object's class is private, its interface is public, as declared by this abstract superclass, MiscDiagramShape.

(See <sup>a</sup>Class Clusters<sup>o</sup> in the introduction to the Foundation Kit for more information on class clusters and creating subclasses within a cluster.)

You instantiate a shape object by sending one of the messages **createWithZone:shapeType:bounds:** or **shapeOfType:bounds:** to the MiscDiagramShape class object. The second method returns an autoreleased shape object.

The shape type of an MiscDiagramShape instance cannot

be changed. You have to instantiate a new instance with the appropriate shape and the bounds of the old instance.

## MiscDiagramShape

**Inherits From:** NSObject  
**Declared In:** MiscDiagramShape.h

### Class Description

A MiscDiagramShape instance can be moved with **moveTo:** and resized with **sizeTo:** and **innerSizeTo:**. The method **bounds** returns the shape bounds and **innerBounds** returns the bounds of a rectangle of maximum size inscribed in the shape. The class method **calcSizeForInnerSize:** gives the minimal size for a shape type such that a rectangle of a given size can be fully inscribed. The bounds and innerBounds are defined like this:

InnerBounds.eps ↪

Shapes can be drawn in a view with **drawOutline**, **drawFill** and **drawShadowWithDelta:**.

The method **hit** provides hit detection with a hit path and the method **calcIntersection:angle:toPoint:** calculates the intersection of a shape with a line running through middle point of the shape's bounds rectangle. The angle of the intersection is given in degrees and is defined like this:

Intersection.eps ↪

## Defined Types

SYNOPSIS

**MiscShapeType**

```
typedef enum {
```

```
MiscCircleShapeType,
```

```
MiscCylinderShapeType,
```

```
MiscDiamondShapeType,
```

```
MiscHexagonShapeType,
```

```
MiscHorizontalArrowSha  
peType,
```

```
MiscHorizontalTriangleSh  
apeType,
```

```
MiscParallelLinesShapeT  
ype,
```

```
MiscParallelogramShape  
Type,
```

MiscRectangleShapeType,  
e,

MiscRoundedRectShapeType,  
Type,

MiscVerticalArrowShapeType,  
ype,

MiscVerticalTriangleShapeType,  
eType,  
} **MiscShapeType**;

## Method Types

Allocating and initializing

+  
createWithZone:shapeType:bounds:  
+ shapeOfType:bounds:

Querying and manipulating shape

- bounds  
- innerBounds  
- shapeType  
- moveTo:  
- sizeTo:  
-  
calcIntersection:angle:toPoint:  
+  
calcSizeForInnerSize:shapeType:  
- reflectHorizontal  
- reflectVertical  
- reflectedHorizontal

	- reflectedVertical
	- setReflectHorizontal:
	- setReflectVertical:
Drawing	- drawOutline
	- drawFill
	- drawShadowWithDelta:
Hit detection	- hitOutline:
	- hitFill:

## Class Methods

### **calcSizeForInnerSize:shapeType:**

+ (NSSize)**calcSizeForInnerSize:**(NSSize)*aSize*  
**shapeType:**(MiscShapeType)*aShapeType*

Calculates the minimal size for a shape of type *aShapeType* such that a rectangle of size *aSize* can be inscribed in the shape.

### **createWithZone:shapeType:bounds:**

+ **createWithZone:**(NSZone \*)*aZone*  
**shapeType:**(MiscShapeType)*aShapeType*  
**bounds:**(NSRect)*aRect*

Creates and returns a new MiscDiagramShape instance of type *aShapeType* and bounds *aRect*. Note, the returned instance is not autoreleased. You are responsible for releasing it. Use **shapeOfType:bounds:** if you want an autoreleased shape.

### **shapeOfType:bounds:**

+ **shapeOfType:**(MiscShapeType)*aShapeType*  
**bounds:**(NSRect)*aRect*

Creates and returns a new MiscDiagramShape instance of type *aShapeType* and bounds *aRect*.

## Instance Methods

### **bounds**

- (NSRect)**bounds**

Returns the bounds of the receiver. See figure above for the difference between bounds and inner bounds.

### **calcIntersection:angle:toPoint:**

- (void)**calcIntersection:**(NSPoint \*)*ip*  
**angle:**(float \*)*alpha*  
**toPoint:**(NSPoint)*aPoint*

Calculates the intersection of a line running through *aPoint* and the middle point of the receiver's bounds with the receiver's outline. The resulting intersection point is placed in *ip* and the resulting intersection angle (given in degrees) in *alpha*. See figure above for a definition of the intersection angle.

### **drawFill**

- (void)**drawFill**

Draws the shape fill. The PostScript focus must be locked on a view when this method is invoked.

### **drawOutline**

- (void)**drawOutline**

Draws the shape outline. The PostScript focus must be locked on a view when this method is invoked.

### **drawShadowWithDelta:**

- (void)**drawShadowWithDelta:**(NSPoint)*aPos*

Draws the shape fill with the receiver displaced with delta

*aPos*. The PostScript focus must be locked on a view when this method is invoked.

### **hitFill:**

- (BOOL)**hitFill:**(MiscHitPath \*)*hitPath*

Returns whether the hit path *hitPath* intersects the fill of the receiver. The PostScript focus must be locked on a view when this method is invoked.

### **hitOutline:**

- (BOOL)**hitOutline:**(MiscHitPath \*)*hitPath*

Returns whether the hit path *hitPath* intersects the outline of the receiver. The PostScript focus must be locked on a view when this method is invoked.

### **innerBounds**

- (NSRect)**innerBounds**

Returns the inner bounds of the receiver. See figure above for the difference between bounds and inner bounds.

### **moveTo:**

- (void)**moveTo:**(NSPoint)*aPos*

Moves the receiver to point *aPos*.

### **reflectHorizontal**

- (void)**reflectHorizontal**

Reflects the receiver horizontally (with the vertical axis through the middle point as the reflection axis). This method affects only shapes which are asymmetrical with respect to the reflection axis.

### **reflectVertical**

- (void)**reflectVertical**

Reflects the receiver vertically (with the horizontal axis through the middle point as the reflection axis). This method affects only shapes which are asymmetrical with respect to the reflection axis.

### **reflectedHorizontal**

- (BOOL)**reflectedHorizontal**

Returns whether the receiver is reflected horizontally.

### **reflectedVertical**

- (BOOL)**reflectedVertical**

Returns whether the receiver is reflected vertically.

### **setReflectHorizontal:**

- (void)**setReflectHorizontal:(BOOL)aBool**

Sets the horizontal reflection state of the receiver.

### **setReflectVertical:**

- (void)**setReflectVertical:(BOOL)aBool**

Sets the vertical reflection state of the receiver.

### **shapeType**

- (MiscShapeType)**shapeType**

Returns the shape type of the receiver.

### **sizeTo:**

- (void)**sizeTo:(NSSize)aSize**

Changes size of receiver to *aSize*. See figure above for the difference between bounds and inner bounds.