| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Arc2D.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/geom/AffineTransform.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Arc2D.Double.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Arc2D.html)    [**NO FRAMES**](http://docs.google.com/Arc2D.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#tyjcwt) | [FIELD](#3dy6vkm) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: [FIELD](#3rdcrjn) | [CONSTR](#1ksv4uv) | [METHOD](#2jxsxqh) |

## **java.awt.geom**

Class Arc2D

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.geom.RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html)  
 **java.awt.geom.Arc2D**

**All Implemented Interfaces:** [Shape](http://docs.google.com/java/awt/Shape.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [Arc2D.Double](http://docs.google.com/java/awt/geom/Arc2D.Double.html), [Arc2D.Float](http://docs.google.com/java/awt/geom/Arc2D.Float.html)

public abstract class **Arc2D**extends [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html)

Arc2D is the abstract superclass for all objects that store a 2D arc defined by a framing rectangle, start angle, angular extent (length of the arc), and a closure type (OPEN, CHORD, or PIE).

The arc is a partial section of a full ellipse which inscribes the framing rectangle of its parent [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html).  The angles are specified relative to the non-square framing rectangle such that 45 degrees always falls on the line from the center of the ellipse to the upper right corner of the framing rectangle. As a result, if the framing rectangle is noticeably longer along one axis than the other, the angles to the start and end of the arc segment will be skewed farther along the longer axis of the frame.

The actual storage representation of the coordinates is left to the subclass.

**Since:** 1.2

| **Nested Class Summary** | |
| --- | --- |
| static class | [**Arc2D.Double**](http://docs.google.com/java/awt/geom/Arc2D.Double.html)            This class defines an arc specified in double precision. |
| static class | [**Arc2D.Float**](http://docs.google.com/java/awt/geom/Arc2D.Float.html)            This class defines an arc specified in float precision. |

| **Field Summary** | |
| --- | --- |
| static int | [**CHORD**](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD)            The closure type for an arc closed by drawing a straight line segment from the start of the arc segment to the end of the arc segment. |
| static int | [**OPEN**](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN)            The closure type for an open arc with no path segments connecting the two ends of the arc segment. |
| static int | [**PIE**](http://docs.google.com/java/awt/geom/Arc2D.html#PIE)            The closure type for an arc closed by drawing straight line segments from the start of the arc segment to the center of the full ellipse and from that point to the end of the arc segment. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Arc2D**](http://docs.google.com/java/awt/geom/Arc2D.html#Arc2D(int))(int type)            This is an abstract class that cannot be instantiated directly. |

| **Method Summary** | |
| --- | --- |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Arc2D.html#contains(double,%20double))(double x, double y)            Determines whether or not the specified point is inside the boundary of the arc. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Arc2D.html#contains(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Determines whether or not the interior of the arc entirely contains the specified rectangle. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Arc2D.html#contains(java.awt.geom.Rectangle2D))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)            Determines whether or not the interior of the arc entirely contains the specified rectangle. |
| boolean | [**containsAngle**](http://docs.google.com/java/awt/geom/Arc2D.html#containsAngle(double))(double angle)            Determines whether or not the specified angle is within the angular extents of the arc. |
| boolean | [**equals**](http://docs.google.com/java/awt/geom/Arc2D.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Determines whether or not the specified Object is equal to this Arc2D. |
| abstract  double | [**getAngleExtent**](http://docs.google.com/java/awt/geom/Arc2D.html#getAngleExtent())()            Returns the angular extent of the arc. |
| abstract  double | [**getAngleStart**](http://docs.google.com/java/awt/geom/Arc2D.html#getAngleStart())()            Returns the starting angle of the arc. |
| int | [**getArcType**](http://docs.google.com/java/awt/geom/Arc2D.html#getArcType())()            Returns the arc closure type of the arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE). |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/geom/Arc2D.html#getBounds2D())()            Returns the high-precision framing rectangle of the arc. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getEndPoint**](http://docs.google.com/java/awt/geom/Arc2D.html#getEndPoint())()            Returns the ending point of the arc. |
| [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) | [**getPathIterator**](http://docs.google.com/java/awt/geom/Arc2D.html#getPathIterator(java.awt.geom.AffineTransform))([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)            Returns an iteration object that defines the boundary of the arc. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getStartPoint**](http://docs.google.com/java/awt/geom/Arc2D.html#getStartPoint())()            Returns the starting point of the arc. |
| int | [**hashCode**](http://docs.google.com/java/awt/geom/Arc2D.html#hashCode())()            Returns the hashcode for this Arc2D. |
| boolean | [**intersects**](http://docs.google.com/java/awt/geom/Arc2D.html#intersects(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Determines whether or not the interior of the arc intersects the interior of the specified rectangle. |
| protected abstract  [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**makeBounds**](http://docs.google.com/java/awt/geom/Arc2D.html#makeBounds(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Constructs a Rectangle2D of the appropriate precision to hold the parameters calculated to be the framing rectangle of this arc. |
| abstract  void | [**setAngleExtent**](http://docs.google.com/java/awt/geom/Arc2D.html#setAngleExtent(double))(double angExt)            Sets the angular extent of this arc to the specified double value. |
| void | [**setAngles**](http://docs.google.com/java/awt/geom/Arc2D.html#setAngles(double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2)            Sets the starting angle and angular extent of this arc using two sets of coordinates. |
| void | [**setAngles**](http://docs.google.com/java/awt/geom/Arc2D.html#setAngles(java.awt.geom.Point2D,%20java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p1, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p2)            Sets the starting angle and angular extent of this arc using two points. |
| abstract  void | [**setAngleStart**](http://docs.google.com/java/awt/geom/Arc2D.html#setAngleStart(double))(double angSt)            Sets the starting angle of this arc to the specified double value. |
| void | [**setAngleStart**](http://docs.google.com/java/awt/geom/Arc2D.html#setAngleStart(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p)            Sets the starting angle of this arc to the angle that the specified point defines relative to the center of this arc. |
| void | [**setArc**](http://docs.google.com/java/awt/geom/Arc2D.html#setArc(java.awt.geom.Arc2D))([Arc2D](http://docs.google.com/java/awt/geom/Arc2D.html) a)            Sets this arc to be the same as the specified arc. |
| abstract  void | [**setArc**](http://docs.google.com/java/awt/geom/Arc2D.html#setArc(double,%20double,%20double,%20double,%20double,%20double,%20int))(double x, double y, double w, double h, double angSt, double angExt, int closure)            Sets the location, size, angular extents, and closure type of this arc to the specified double values. |
| void | [**setArc**](http://docs.google.com/java/awt/geom/Arc2D.html#setArc(java.awt.geom.Point2D,%20java.awt.geom.Dimension2D,%20double,%20double,%20int))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) loc, [Dimension2D](http://docs.google.com/java/awt/geom/Dimension2D.html) size, double angSt, double angExt, int closure)            Sets the location, size, angular extents, and closure type of this arc to the specified values. |
| void | [**setArc**](http://docs.google.com/java/awt/geom/Arc2D.html#setArc(java.awt.geom.Rectangle2D,%20double,%20double,%20int))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) rect, double angSt, double angExt, int closure)            Sets the location, size, angular extents, and closure type of this arc to the specified values. |
| void | [**setArcByCenter**](http://docs.google.com/java/awt/geom/Arc2D.html#setArcByCenter(double,%20double,%20double,%20double,%20double,%20int))(double x, double y, double radius, double angSt, double angExt, int closure)            Sets the position, bounds, angular extents, and closure type of this arc to the specified values. |
| void | [**setArcByTangent**](http://docs.google.com/java/awt/geom/Arc2D.html#setArcByTangent(java.awt.geom.Point2D,%20java.awt.geom.Point2D,%20java.awt.geom.Point2D,%20double))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p1, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p2, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p3, double radius)            Sets the position, bounds, and angular extents of this arc to the specified value. |
| void | [**setArcType**](http://docs.google.com/java/awt/geom/Arc2D.html#setArcType(int))(int type)            Sets the closure type of this arc to the specified value: OPEN, CHORD, or PIE. |
| void | [**setFrame**](http://docs.google.com/java/awt/geom/Arc2D.html#setFrame(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Sets the location and size of the framing rectangle of this Shape to the specified rectangular values. |

| **Methods inherited from class java.awt.geom.**[**RectangularShape**](http://docs.google.com/java/awt/geom/RectangularShape.html) |
| --- |
| [clone](http://docs.google.com/java/awt/geom/RectangularShape.html#clone()), [contains](http://docs.google.com/java/awt/geom/RectangularShape.html#contains(java.awt.geom.Point2D)), [getBounds](http://docs.google.com/java/awt/geom/RectangularShape.html#getBounds()), [getCenterX](http://docs.google.com/java/awt/geom/RectangularShape.html#getCenterX()), [getCenterY](http://docs.google.com/java/awt/geom/RectangularShape.html#getCenterY()), [getFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#getFrame()), [getHeight](http://docs.google.com/java/awt/geom/RectangularShape.html#getHeight()), [getMaxX](http://docs.google.com/java/awt/geom/RectangularShape.html#getMaxX()), [getMaxY](http://docs.google.com/java/awt/geom/RectangularShape.html#getMaxY()), [getMinX](http://docs.google.com/java/awt/geom/RectangularShape.html#getMinX()), [getMinY](http://docs.google.com/java/awt/geom/RectangularShape.html#getMinY()), [getPathIterator](http://docs.google.com/java/awt/geom/RectangularShape.html#getPathIterator(java.awt.geom.AffineTransform,%20double)), [getWidth](http://docs.google.com/java/awt/geom/RectangularShape.html#getWidth()), [getX](http://docs.google.com/java/awt/geom/RectangularShape.html#getX()), [getY](http://docs.google.com/java/awt/geom/RectangularShape.html#getY()), [intersects](http://docs.google.com/java/awt/geom/RectangularShape.html#intersects(java.awt.geom.Rectangle2D)), [isEmpty](http://docs.google.com/java/awt/geom/RectangularShape.html#isEmpty()), [setFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrame(java.awt.geom.Point2D,%20java.awt.geom.Dimension2D)), [setFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrame(java.awt.geom.Rectangle2D)), [setFrameFromCenter](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromCenter(double,%20double,%20double,%20double)), [setFrameFromCenter](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromCenter(java.awt.geom.Point2D,%20java.awt.geom.Point2D)), [setFrameFromDiagonal](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromDiagonal(double,%20double,%20double,%20double)), [setFrameFromDiagonal](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromDiagonal(java.awt.geom.Point2D,%20java.awt.geom.Point2D)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### OPEN

public static final int **OPEN**

The closure type for an open arc with no path segments connecting the two ends of the arc segment.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.geom.Arc2D.OPEN)

### CHORD

public static final int **CHORD**

The closure type for an arc closed by drawing a straight line segment from the start of the arc segment to the end of the arc segment.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.geom.Arc2D.CHORD)

### PIE

public static final int **PIE**

The closure type for an arc closed by drawing straight line segments from the start of the arc segment to the center of the full ellipse and from that point to the end of the arc segment.

**Since:** 1.2 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.geom.Arc2D.PIE)

| **Constructor Detail** |
| --- |

### Arc2D

protected **Arc2D**(int type)

This is an abstract class that cannot be instantiated directly. Type-specific implementation subclasses are available for instantiation and provide a number of formats for storing the information necessary to satisfy the various accessor methods below.

**Parameters:**type - The closure type of this arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE).**Since:** 1.2 **See Also:**[Arc2D.Float](http://docs.google.com/java/awt/geom/Arc2D.Float.html), [Arc2D.Double](http://docs.google.com/java/awt/geom/Arc2D.Double.html)

| **Method Detail** |
| --- |

### getAngleStart

public abstract double **getAngleStart**()

Returns the starting angle of the arc.

**Returns:**A double value that represents the starting angle of the arc in degrees.**Since:** 1.2 **See Also:**[setAngleStart(double)](http://docs.google.com/java/awt/geom/Arc2D.html#setAngleStart(double))

### getAngleExtent

public abstract double **getAngleExtent**()

Returns the angular extent of the arc.

**Returns:**A double value that represents the angular extent of the arc in degrees.**Since:** 1.2 **See Also:**[setAngleExtent(double)](http://docs.google.com/java/awt/geom/Arc2D.html#setAngleExtent(double))

### getArcType

public int **getArcType**()

Returns the arc closure type of the arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE).

**Returns:**One of the integer constant closure types defined in this class.**Since:** 1.2 **See Also:**[setArcType(int)](http://docs.google.com/java/awt/geom/Arc2D.html#setArcType(int))

### getStartPoint

public [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getStartPoint**()

Returns the starting point of the arc. This point is the intersection of the ray from the center defined by the starting angle and the elliptical boundary of the arc.

**Returns:**A Point2D object representing the x,y coordinates of the starting point of the arc.**Since:** 1.2

### getEndPoint

public [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getEndPoint**()

Returns the ending point of the arc. This point is the intersection of the ray from the center defined by the starting angle plus the angular extent of the arc and the elliptical boundary of the arc.

**Returns:**A Point2D object representing the x,y coordinates of the ending point of the arc.**Since:** 1.2

### setArc

public abstract void **setArc**(double x,  
 double y,  
 double w,  
 double h,  
 double angSt,  
 double angExt,  
 int closure)

Sets the location, size, angular extents, and closure type of this arc to the specified double values.

**Parameters:**x - The X coordinate of the upper-left corner of the arc.y - The Y coordinate of the upper-left corner of the arc.w - The overall width of the full ellipse of which this arc is a partial section.h - The overall height of the full ellipse of which this arc is a partial section.angSt - The starting angle of the arc in degrees.angExt - The angular extent of the arc in degrees.closure - The closure type for the arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE).**Since:** 1.2

### setArc

public void **setArc**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) loc,  
 [Dimension2D](http://docs.google.com/java/awt/geom/Dimension2D.html) size,  
 double angSt,  
 double angExt,  
 int closure)

Sets the location, size, angular extents, and closure type of this arc to the specified values.

**Parameters:**loc - The Point2D representing the coordinates of the upper-left corner of the arc.size - The Dimension2D representing the width and height of the full ellipse of which this arc is a partial section.angSt - The starting angle of the arc in degrees.angExt - The angular extent of the arc in degrees.closure - The closure type for the arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE).**Since:** 1.2

### setArc

public void **setArc**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) rect,  
 double angSt,  
 double angExt,  
 int closure)

Sets the location, size, angular extents, and closure type of this arc to the specified values.

**Parameters:**rect - The framing rectangle that defines the outer boundary of the full ellipse of which this arc is a partial section.angSt - The starting angle of the arc in degrees.angExt - The angular extent of the arc in degrees.closure - The closure type for the arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE).**Since:** 1.2

### setArc

public void **setArc**([Arc2D](http://docs.google.com/java/awt/geom/Arc2D.html) a)

Sets this arc to be the same as the specified arc.

**Parameters:**a - The Arc2D to use to set the arc's values.**Since:** 1.2

### setArcByCenter

public void **setArcByCenter**(double x,  
 double y,  
 double radius,  
 double angSt,  
 double angExt,  
 int closure)

Sets the position, bounds, angular extents, and closure type of this arc to the specified values. The arc is defined by a center point and a radius rather than a framing rectangle for the full ellipse.

**Parameters:**x - The X coordinate of the center of the arc.y - The Y coordinate of the center of the arc.radius - The radius of the arc.angSt - The starting angle of the arc in degrees.angExt - The angular extent of the arc in degrees.closure - The closure type for the arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE).**Since:** 1.2

### setArcByTangent

public void **setArcByTangent**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p1,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p2,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p3,  
 double radius)

Sets the position, bounds, and angular extents of this arc to the specified value. The starting angle of the arc is tangent to the line specified by points (p1, p2), the ending angle is tangent to the line specified by points (p2, p3), and the arc has the specified radius.

**Parameters:**p1 - The first point that defines the arc. The starting angle of the arc is tangent to the line specified by points (p1, p2).p2 - The second point that defines the arc. The starting angle of the arc is tangent to the line specified by points (p1, p2). The ending angle of the arc is tangent to the line specified by points (p2, p3).p3 - The third point that defines the arc. The ending angle of the arc is tangent to the line specified by points (p2, p3).radius - The radius of the arc.**Since:** 1.2

### setAngleStart

public abstract void **setAngleStart**(double angSt)

Sets the starting angle of this arc to the specified double value.

**Parameters:**angSt - The starting angle of the arc in degrees.**Since:** 1.2 **See Also:**[getAngleStart()](http://docs.google.com/java/awt/geom/Arc2D.html#getAngleStart())

### setAngleExtent

public abstract void **setAngleExtent**(double angExt)

Sets the angular extent of this arc to the specified double value.

**Parameters:**angExt - The angular extent of the arc in degrees.**Since:** 1.2 **See Also:**[getAngleExtent()](http://docs.google.com/java/awt/geom/Arc2D.html#getAngleExtent())

### setAngleStart

public void **setAngleStart**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p)

Sets the starting angle of this arc to the angle that the specified point defines relative to the center of this arc. The angular extent of the arc will remain the same.

**Parameters:**p - The Point2D that defines the starting angle.**Since:** 1.2 **See Also:**[getAngleStart()](http://docs.google.com/java/awt/geom/Arc2D.html#getAngleStart())

### setAngles

public void **setAngles**(double x1,  
 double y1,  
 double x2,  
 double y2)

Sets the starting angle and angular extent of this arc using two sets of coordinates. The first set of coordinates is used to determine the angle of the starting point relative to the arc's center. The second set of coordinates is used to determine the angle of the end point relative to the arc's center. The arc will always be non-empty and extend counterclockwise from the first point around to the second point.

**Parameters:**x1 - The X coordinate of the arc's starting point.y1 - The Y coordinate of the arc's starting point.x2 - The X coordinate of the arc's ending point.y2 - The Y coordinate of the arc's ending point.**Since:** 1.2

### setAngles

public void **setAngles**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p1,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p2)

Sets the starting angle and angular extent of this arc using two points. The first point is used to determine the angle of the starting point relative to the arc's center. The second point is used to determine the angle of the end point relative to the arc's center. The arc will always be non-empty and extend counterclockwise from the first point around to the second point.

**Parameters:**p1 - The Point2D that defines the arc's starting point.p2 - The Point2D that defines the arc's ending point.**Since:** 1.2

### setArcType

public void **setArcType**(int type)

Sets the closure type of this arc to the specified value: OPEN, CHORD, or PIE.

**Parameters:**type - The integer constant that represents the closure type of this arc: [OPEN](http://docs.google.com/java/awt/geom/Arc2D.html#OPEN), [CHORD](http://docs.google.com/java/awt/geom/Arc2D.html#CHORD), or [PIE](http://docs.google.com/java/awt/geom/Arc2D.html#PIE). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if type is not 0, 1, or 2.+**Since:** 1.2 **See Also:**[getArcType()](http://docs.google.com/java/awt/geom/Arc2D.html#getArcType())

### setFrame

public void **setFrame**(double x,  
 double y,  
 double w,  
 double h)

Sets the location and size of the framing rectangle of this Shape to the specified rectangular values. Note that the arc [partially inscribes](http://docs.google.com/Arc2D.html#inscribes) the framing rectangle of this RectangularShape.

**Specified by:**[setFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrame(double,%20double,%20double,%20double)) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Parameters:**x - the X coordinate of the upper-left corner of the specified rectangular shapey - the Y coordinate of the upper-left corner of the specified rectangular shapew - the width of the specified rectangular shapeh - the height of the specified rectangular shape**Since:** 1.2 **See Also:**[RectangularShape.getFrame()](http://docs.google.com/java/awt/geom/RectangularShape.html#getFrame())

### getBounds2D

public [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **getBounds2D**()

Returns the high-precision framing rectangle of the arc. The framing rectangle contains only the part of this Arc2D that is in between the starting and ending angles and contains the pie wedge, if this Arc2D has a PIE closure type.

This method differs from the [getBounds](http://docs.google.com/java/awt/geom/RectangularShape.html#getBounds()) in that the getBounds method only returns the bounds of the enclosing ellipse of this Arc2D without considering the starting and ending angles of this Arc2D.

**Returns:**the Rectangle2D that represents the arc's framing rectangle.**Since:** 1.2 **See Also:**[Shape.getBounds()](http://docs.google.com/java/awt/Shape.html#getBounds())

### makeBounds

protected abstract [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **makeBounds**(double x,  
 double y,  
 double w,  
 double h)

Constructs a Rectangle2D of the appropriate precision to hold the parameters calculated to be the framing rectangle of this arc.

**Parameters:**x - The X coordinate of the upper-left corner of the framing rectangle.y - The Y coordinate of the upper-left corner of the framing rectangle.w - The width of the framing rectangle.h - The height of the framing rectangle. **Returns:**a Rectangle2D that is the framing rectangle of this arc.**Since:** 1.2

### containsAngle

public boolean **containsAngle**(double angle)

Determines whether or not the specified angle is within the angular extents of the arc.

**Parameters:**angle - The angle to test. **Returns:**true if the arc contains the angle, false if the arc doesn't contain the angle.**Since:** 1.2

### contains

public boolean **contains**(double x,  
 double y)

Determines whether or not the specified point is inside the boundary of the arc.

**Parameters:**x - The X coordinate of the point to test.y - The Y coordinate of the point to test. **Returns:**true if the point lies within the bound of the arc, false if the point lies outside of the arc's bounds.**Since:** 1.2

### intersects

public boolean **intersects**(double x,  
 double y,  
 double w,  
 double h)

Determines whether or not the interior of the arc intersects the interior of the specified rectangle.

**Parameters:**x - The X coordinate of the rectangle's upper-left corner.y - The Y coordinate of the rectangle's upper-left corner.w - The width of the rectangle.h - The height of the rectangle. **Returns:**true if the arc intersects the rectangle, false if the arc doesn't intersect the rectangle.**Since:** 1.2 **See Also:**[Area](http://docs.google.com/java/awt/geom/Area.html)

### contains

public boolean **contains**(double x,  
 double y,  
 double w,  
 double h)

Determines whether or not the interior of the arc entirely contains the specified rectangle.

**Parameters:**x - The X coordinate of the rectangle's upper-left corner.y - The Y coordinate of the rectangle's upper-left corner.w - The width of the rectangle.h - The height of the rectangle. **Returns:**true if the arc contains the rectangle, false if the arc doesn't contain the rectangle.**Since:** 1.2 **See Also:**[Area](http://docs.google.com/java/awt/geom/Area.html), [Shape.intersects(double, double, double, double)](http://docs.google.com/java/awt/Shape.html#intersects(double,%20double,%20double,%20double))

### contains

public boolean **contains**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)

Determines whether or not the interior of the arc entirely contains the specified rectangle.

**Specified by:**[contains](http://docs.google.com/java/awt/Shape.html#contains(java.awt.geom.Rectangle2D)) in interface [Shape](http://docs.google.com/java/awt/Shape.html)**Overrides:**[contains](http://docs.google.com/java/awt/geom/RectangularShape.html#contains(java.awt.geom.Rectangle2D)) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Parameters:**r - The Rectangle2D to test. **Returns:**true if the arc contains the rectangle, false if the arc doesn't contain the rectangle.**Since:** 1.2 **See Also:**[Shape.contains(double, double, double, double)](http://docs.google.com/java/awt/Shape.html#contains(double,%20double,%20double,%20double))

### getPathIterator

public [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) **getPathIterator**([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)

Returns an iteration object that defines the boundary of the arc. This iterator is multithread safe. Arc2D guarantees that modifications to the geometry of the arc do not affect any iterations of that geometry that are already in process.

**Parameters:**at - an optional AffineTransform to be applied to the coordinates as they are returned in the iteration, or null if the untransformed coordinates are desired. **Returns:**A PathIterator that defines the arc's boundary.**Since:** 1.2

### hashCode

public int **hashCode**()

Returns the hashcode for this Arc2D.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the hashcode for this Arc2D.**Since:** 1.6 **See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Determines whether or not the specified Object is equal to this Arc2D. The specified Object is equal to this Arc2D if it is an instance of Arc2D and if its location, size, arc extents and type are the same as this Arc2D.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - an Object to be compared with this Arc2D. **Returns:**true if obj is an instance of Arc2D and has the same values; false otherwise.**Since:** 1.6 **See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Arc2D.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/geom/AffineTransform.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Arc2D.Double.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Arc2D.html)    [**NO FRAMES**](http://docs.google.com/Arc2D.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#tyjcwt) | [FIELD](#3dy6vkm) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: [FIELD](#3rdcrjn) | [CONSTR](#1ksv4uv) | [METHOD](#2jxsxqh) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).