| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Path2D.Double.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/Path2D.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Path2D.Float.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Path2D.Double.html)    [**NO FRAMES**](http://docs.google.com/Path2D.Double.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#3dy6vkm) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#2jxsxqh) |

## **java.awt.geom**

Class Path2D.Double

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.geom.Path2D](http://docs.google.com/java/awt/geom/Path2D.html)  
 **java.awt.geom.Path2D.Double**

**All Implemented Interfaces:** [Shape](http://docs.google.com/java/awt/Shape.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Enclosing class:**[Path2D](http://docs.google.com/java/awt/geom/Path2D.html)

public static class **Path2D.Double**extends [Path2D](http://docs.google.com/java/awt/geom/Path2D.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

The Double class defines a geometric path with coordinates stored in double precision floating point.

**Since:** 1.6 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.awt.geom.Path2D.Double)

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from class java.awt.geom.**[**Path2D**](http://docs.google.com/java/awt/geom/Path2D.html) |
| --- |
| [Path2D.Double](http://docs.google.com/java/awt/geom/Path2D.Double.html), [Path2D.Float](http://docs.google.com/java/awt/geom/Path2D.Float.html) |

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class java.awt.geom.**[**Path2D**](http://docs.google.com/java/awt/geom/Path2D.html) |
| --- |
| [WIND\_EVEN\_ODD](http://docs.google.com/java/awt/geom/Path2D.html#WIND_EVEN_ODD), [WIND\_NON\_ZERO](http://docs.google.com/java/awt/geom/Path2D.html#WIND_NON_ZERO) |

| **Constructor Summary** | |
| --- | --- |
| [**Path2D.Double**](http://docs.google.com/java/awt/geom/Path2D.Double.html#Path2D.Double())()            Constructs a new empty double precision Path2D object with a default winding rule of [Path2D.WIND\_NON\_ZERO](http://docs.google.com/java/awt/geom/Path2D.html#WIND_NON_ZERO). |
| [**Path2D.Double**](http://docs.google.com/java/awt/geom/Path2D.Double.html#Path2D.Double(int))(int rule)            Constructs a new empty double precision Path2D object with the specified winding rule to control operations that require the interior of the path to be defined. |
| [**Path2D.Double**](http://docs.google.com/java/awt/geom/Path2D.Double.html#Path2D.Double(int,%20int))(int rule, int initialCapacity)            Constructs a new empty double precision Path2D object with the specified winding rule and the specified initial capacity to store path segments. |
| [**Path2D.Double**](http://docs.google.com/java/awt/geom/Path2D.Double.html#Path2D.Double(java.awt.Shape))([Shape](http://docs.google.com/java/awt/Shape.html) s)            Constructs a new double precision Path2D object from an arbitrary [Shape](http://docs.google.com/java/awt/Shape.html) object. |
| [**Path2D.Double**](http://docs.google.com/java/awt/geom/Path2D.Double.html#Path2D.Double(java.awt.Shape,%20java.awt.geom.AffineTransform))([Shape](http://docs.google.com/java/awt/Shape.html) s, [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)            Constructs a new double precision Path2D object from an arbitrary [Shape](http://docs.google.com/java/awt/Shape.html) object, transformed by an [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) object. |

| **Method Summary** | |
| --- | --- |
| void | [**append**](http://docs.google.com/java/awt/geom/Path2D.Double.html#append(java.awt.geom.PathIterator,%20boolean))([PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) pi, boolean connect)            Appends the geometry of the specified [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) object to the path, possibly connecting the new geometry to the existing path segments with a line segment. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/awt/geom/Path2D.Double.html#clone())()            Creates a new object of the same class as this object. |
| void | [**curveTo**](http://docs.google.com/java/awt/geom/Path2D.Double.html#curveTo(double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double x3, double y3)            Adds a curved segment, defined by three new points, to the path by drawing a Bézier curve that intersects both the current coordinates and the specified coordinates (x3,y3), using the specified points (x1,y1) and (x2,y2) as Bézier control points. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/geom/Path2D.Double.html#getBounds2D())()            Returns a high precision and more accurate bounding box of the Shape than the getBounds method. |
| [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) | [**getPathIterator**](http://docs.google.com/java/awt/geom/Path2D.Double.html#getPathIterator(java.awt.geom.AffineTransform))([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)            Returns an iterator object that iterates along the Shape boundary and provides access to the geometry of the Shape outline. |
| void | [**lineTo**](http://docs.google.com/java/awt/geom/Path2D.Double.html#lineTo(double,%20double))(double x, double y)            Adds a point to the path by drawing a straight line from the current coordinates to the new specified coordinates specified in double precision. |
| void | [**moveTo**](http://docs.google.com/java/awt/geom/Path2D.Double.html#moveTo(double,%20double))(double x, double y)            Adds a point to the path by moving to the specified coordinates specified in double precision. |
| void | [**quadTo**](http://docs.google.com/java/awt/geom/Path2D.Double.html#quadTo(double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2)            Adds a curved segment, defined by two new points, to the path by drawing a Quadratic curve that intersects both the current coordinates and the specified coordinates (x2,y2), using the specified point (x1,y1) as a quadratic parametric control point. |
| void | [**transform**](http://docs.google.com/java/awt/geom/Path2D.Double.html#transform(java.awt.geom.AffineTransform))([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)            Transforms the geometry of this path using the specified [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html). |

| **Methods inherited from class java.awt.geom.**[**Path2D**](http://docs.google.com/java/awt/geom/Path2D.html) |
| --- |
| [append](http://docs.google.com/java/awt/geom/Path2D.html#append(java.awt.Shape,%20boolean)), [closePath](http://docs.google.com/java/awt/geom/Path2D.html#closePath()), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(double,%20double)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(double,%20double,%20double,%20double)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(java.awt.geom.PathIterator,%20double,%20double)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(java.awt.geom.PathIterator,%20double,%20double,%20double,%20double)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(java.awt.geom.PathIterator,%20java.awt.geom.Point2D)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(java.awt.geom.PathIterator,%20java.awt.geom.Rectangle2D)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(java.awt.geom.Point2D)), [contains](http://docs.google.com/java/awt/geom/Path2D.html#contains(java.awt.geom.Rectangle2D)), [createTransformedShape](http://docs.google.com/java/awt/geom/Path2D.html#createTransformedShape(java.awt.geom.AffineTransform)), [getBounds](http://docs.google.com/java/awt/geom/Path2D.html#getBounds()), [getCurrentPoint](http://docs.google.com/java/awt/geom/Path2D.html#getCurrentPoint()), [getPathIterator](http://docs.google.com/java/awt/geom/Path2D.html#getPathIterator(java.awt.geom.AffineTransform,%20double)), [getWindingRule](http://docs.google.com/java/awt/geom/Path2D.html#getWindingRule()), [intersects](http://docs.google.com/java/awt/geom/Path2D.html#intersects(double,%20double,%20double,%20double)), [intersects](http://docs.google.com/java/awt/geom/Path2D.html#intersects(java.awt.geom.PathIterator,%20double,%20double,%20double,%20double)), [intersects](http://docs.google.com/java/awt/geom/Path2D.html#intersects(java.awt.geom.PathIterator,%20java.awt.geom.Rectangle2D)), [intersects](http://docs.google.com/java/awt/geom/Path2D.html#intersects(java.awt.geom.Rectangle2D)), [reset](http://docs.google.com/java/awt/geom/Path2D.html#reset()), [setWindingRule](http://docs.google.com/java/awt/geom/Path2D.html#setWindingRule(int)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [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)) |

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

### Path2D.Double

public **Path2D.Double**()

Constructs a new empty double precision Path2D object with a default winding rule of [Path2D.WIND\_NON\_ZERO](http://docs.google.com/java/awt/geom/Path2D.html#WIND_NON_ZERO).

**Since:** 1.6

### Path2D.Double

public **Path2D.Double**(int rule)

Constructs a new empty double precision Path2D object with the specified winding rule to control operations that require the interior of the path to be defined.

**Parameters:**rule - the winding rule**Since:** 1.6 **See Also:**[Path2D.WIND\_EVEN\_ODD](http://docs.google.com/java/awt/geom/Path2D.html#WIND_EVEN_ODD), [Path2D.WIND\_NON\_ZERO](http://docs.google.com/java/awt/geom/Path2D.html#WIND_NON_ZERO)

### Path2D.Double

public **Path2D.Double**(int rule,  
 int initialCapacity)

Constructs a new empty double precision Path2D object with the specified winding rule and the specified initial capacity to store path segments. This number is an initial guess as to how many path segments are in the path, but the storage is expanded as needed to store whatever path segments are added to this path.

**Parameters:**rule - the winding ruleinitialCapacity - the estimate for the number of path segments in the path**Since:** 1.6 **See Also:**[Path2D.WIND\_EVEN\_ODD](http://docs.google.com/java/awt/geom/Path2D.html#WIND_EVEN_ODD), [Path2D.WIND\_NON\_ZERO](http://docs.google.com/java/awt/geom/Path2D.html#WIND_NON_ZERO)

### Path2D.Double

public **Path2D.Double**([Shape](http://docs.google.com/java/awt/Shape.html) s)

Constructs a new double precision Path2D object from an arbitrary [Shape](http://docs.google.com/java/awt/Shape.html) object. All of the initial geometry and the winding rule for this path are taken from the specified Shape object.

**Parameters:**s - the specified Shape object**Since:** 1.6

### Path2D.Double

public **Path2D.Double**([Shape](http://docs.google.com/java/awt/Shape.html) s,  
 [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)

Constructs a new double precision Path2D object from an arbitrary [Shape](http://docs.google.com/java/awt/Shape.html) object, transformed by an [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) object. All of the initial geometry and the winding rule for this path are taken from the specified Shape object and transformed by the specified AffineTransform object.

**Parameters:**s - the specified Shape objectat - the specified AffineTransform object**Since:** 1.6

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

### moveTo

public final void **moveTo**(double x,  
 double y)

Adds a point to the path by moving to the specified coordinates specified in double precision.

**Specified by:**[moveTo](http://docs.google.com/java/awt/geom/Path2D.html#moveTo(double,%20double)) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Parameters:**x - the specified X coordinatey - the specified Y coordinate**Since:** 1.6

### lineTo

public final void **lineTo**(double x,  
 double y)

Adds a point to the path by drawing a straight line from the current coordinates to the new specified coordinates specified in double precision.

**Specified by:**[lineTo](http://docs.google.com/java/awt/geom/Path2D.html#lineTo(double,%20double)) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Parameters:**x - the specified X coordinatey - the specified Y coordinate**Since:** 1.6

### quadTo

public final void **quadTo**(double x1,  
 double y1,  
 double x2,  
 double y2)

Adds a curved segment, defined by two new points, to the path by drawing a Quadratic curve that intersects both the current coordinates and the specified coordinates (x2,y2), using the specified point (x1,y1) as a quadratic parametric control point. All coordinates are specified in double precision.

**Specified by:**[quadTo](http://docs.google.com/java/awt/geom/Path2D.html#quadTo(double,%20double,%20double,%20double)) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Parameters:**x1 - the X coordinate of the quadratic control pointy1 - the Y coordinate of the quadratic control pointx2 - the X coordinate of the final end pointy2 - the Y coordinate of the final end point**Since:** 1.6

### curveTo

public final void **curveTo**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double x3,  
 double y3)

Adds a curved segment, defined by three new points, to the path by drawing a Bézier curve that intersects both the current coordinates and the specified coordinates (x3,y3), using the specified points (x1,y1) and (x2,y2) as Bézier control points. All coordinates are specified in double precision.

**Specified by:**[curveTo](http://docs.google.com/java/awt/geom/Path2D.html#curveTo(double,%20double,%20double,%20double,%20double,%20double)) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Parameters:**x1 - the X coordinate of the first Bézier control pointy1 - the Y coordinate of the first Bézier control pointx2 - the X coordinate of the second Bézier control pointy2 - the Y coordinate of the second Bézier control pointx3 - the X coordinate of the final end pointy3 - the Y coordinate of the final end point**Since:** 1.6

### append

public final void **append**([PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) pi,  
 boolean connect)

Appends the geometry of the specified [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) object to the path, possibly connecting the new geometry to the existing path segments with a line segment. If the connect parameter is true and the path is not empty then any initial moveTo in the geometry of the appended Shape is turned into a lineTo segment. If the destination coordinates of such a connecting lineTo segment match the ending coordinates of a currently open subpath then the segment is omitted as superfluous. The winding rule of the specified Shape is ignored and the appended geometry is governed by the winding rule specified for this path.

**Specified by:**[append](http://docs.google.com/java/awt/geom/Path2D.html#append(java.awt.geom.PathIterator,%20boolean)) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Parameters:**pi - the PathIterator whose geometry is appended to this pathconnect - a boolean to control whether or not to turn an initial moveTo segment into a lineTo segment to connect the new geometry to the existing path**Since:** 1.6

### transform

public final void **transform**([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)

Transforms the geometry of this path using the specified [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html). The geometry is transformed in place, which permanently changes the boundary defined by this object.

**Specified by:**[transform](http://docs.google.com/java/awt/geom/Path2D.html#transform(java.awt.geom.AffineTransform)) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Parameters:**at - the AffineTransform used to transform the area**Since:** 1.6

### getBounds2D

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

Returns a high precision and more accurate bounding box of the Shape than the getBounds method. Note that there is no guarantee that the returned [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) is the smallest bounding box that encloses the Shape, only that the Shape lies entirely within the indicated Rectangle2D. The bounding box returned by this method is usually tighter than that returned by the getBounds method and never fails due to overflow problems since the return value can be an instance of the Rectangle2D that uses double precision values to store the dimensions.

**Specified by:**[getBounds2D](http://docs.google.com/java/awt/Shape.html#getBounds2D()) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Returns:**an instance of Rectangle2D that is a high-precision bounding box of the Shape.**Since:** 1.6 **See Also:**[Shape.getBounds()](http://docs.google.com/java/awt/Shape.html#getBounds())

### 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 iterator object that iterates along the Shape boundary and provides access to the geometry of the Shape outline. If an optional [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) is specified, the coordinates returned in the iteration are transformed accordingly.

Each call to this method returns a fresh PathIterator object that traverses the geometry of the Shape object independently from any other PathIterator objects in use at the same time.

It is recommended, but not guaranteed, that objects implementing the Shape interface isolate iterations that are in process from any changes that might occur to the original object's geometry during such iterations.

The iterator for this class is not multi-threaded safe, which means that the Path2D class does not guarantee that modifications to the geometry of this Path2D object do not affect any iterations of that geometry that are already in process.

**Specified by:**[getPathIterator](http://docs.google.com/java/awt/Shape.html#getPathIterator(java.awt.geom.AffineTransform)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**at - an AffineTransform **Returns:**a new PathIterator that iterates along the boundary of this Shape and provides access to the geometry of this Shape's outline**Since:** 1.6

### clone

public final [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Creates a new object of the same class as this object.

**Specified by:**[clone](http://docs.google.com/java/awt/geom/Path2D.html#clone()) in class [Path2D](http://docs.google.com/java/awt/geom/Path2D.html) **Returns:**a clone of this instance. **Throws:** [OutOfMemoryError](http://docs.google.com/java/lang/OutOfMemoryError.html) - if there is not enough memory.**Since:** 1.6 **See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.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/Path2D.Double.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/Path2D.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Path2D.Float.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Path2D.Double.html)    [**NO FRAMES**](http://docs.google.com/Path2D.Double.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#3dy6vkm) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [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).