| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/java/awt/font/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/awt/im/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package java.awt.geom

Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**PathIterator**](http://docs.google.com/java/awt/geom/PathIterator.html) | The PathIterator interface provides the mechanism for objects that implement the [Shape](http://docs.google.com/java/awt/Shape.html) interface to return the geometry of their boundary by allowing a caller to retrieve the path of that boundary a segment at a time. |

| **Class Summary** | |
| --- | --- |
| [**AffineTransform**](http://docs.google.com/java/awt/geom/AffineTransform.html) | The AffineTransform class represents a 2D affine transform that performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelness" of lines. |
| [**Arc2D**](http://docs.google.com/java/awt/geom/Arc2D.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). |
| [**Arc2D.Double**](http://docs.google.com/java/awt/geom/Arc2D.Double.html) | This class defines an arc specified in double precision. |
| [**Arc2D.Float**](http://docs.google.com/java/awt/geom/Arc2D.Float.html) | This class defines an arc specified in float precision. |
| [**Area**](http://docs.google.com/java/awt/geom/Area.html) | An Area object stores and manipulates a resolution-independent description of an enclosed area of 2-dimensional space. |
| [**CubicCurve2D**](http://docs.google.com/java/awt/geom/CubicCurve2D.html) | The CubicCurve2D class defines a cubic parametric curve segment in (x,y) coordinate space. |
| [**CubicCurve2D.Double**](http://docs.google.com/java/awt/geom/CubicCurve2D.Double.html) | A cubic parametric curve segment specified with double coordinates. |
| [**CubicCurve2D.Float**](http://docs.google.com/java/awt/geom/CubicCurve2D.Float.html) | A cubic parametric curve segment specified with float coordinates. |
| [**Dimension2D**](http://docs.google.com/java/awt/geom/Dimension2D.html) | The Dimension2D class is to encapsulate a width and a height dimension. |
| [**Ellipse2D**](http://docs.google.com/java/awt/geom/Ellipse2D.html) | The Ellipse2D class describes an ellipse that is defined by a framing rectangle. |
| [**Ellipse2D.Double**](http://docs.google.com/java/awt/geom/Ellipse2D.Double.html) | The Double class defines an ellipse specified in double precision. |
| [**Ellipse2D.Float**](http://docs.google.com/java/awt/geom/Ellipse2D.Float.html) | The Float class defines an ellipse specified in float precision. |
| [**FlatteningPathIterator**](http://docs.google.com/java/awt/geom/FlatteningPathIterator.html) | The FlatteningPathIterator class returns a flattened view of another [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) object. |
| [**GeneralPath**](http://docs.google.com/java/awt/geom/GeneralPath.html) | The GeneralPath class represents a geometric path constructed from straight lines, and quadratic and cubic (Bézier) curves. |
| [**Line2D**](http://docs.google.com/java/awt/geom/Line2D.html) | This Line2D represents a line segment in (x,y) coordinate space. |
| [**Line2D.Double**](http://docs.google.com/java/awt/geom/Line2D.Double.html) | A line segment specified with double coordinates. |
| [**Line2D.Float**](http://docs.google.com/java/awt/geom/Line2D.Float.html) | A line segment specified with float coordinates. |
| [**Path2D**](http://docs.google.com/java/awt/geom/Path2D.html) | The Path2D class provides a simple, yet flexible shape which represents an arbitrary geometric path. |
| [**Path2D.Double**](http://docs.google.com/java/awt/geom/Path2D.Double.html) | The Double class defines a geometric path with coordinates stored in double precision floating point. |
| [**Path2D.Float**](http://docs.google.com/java/awt/geom/Path2D.Float.html) | The Float class defines a geometric path with coordinates stored in single precision floating point. |
| [**Point2D**](http://docs.google.com/java/awt/geom/Point2D.html) | The Point2D class defines a point representing a location in (x,y) coordinate space. |
| [**Point2D.Double**](http://docs.google.com/java/awt/geom/Point2D.Double.html) | The Double class defines a point specified in double precision. |
| [**Point2D.Float**](http://docs.google.com/java/awt/geom/Point2D.Float.html) | The Float class defines a point specified in float precision. |
| [**QuadCurve2D**](http://docs.google.com/java/awt/geom/QuadCurve2D.html) | The QuadCurve2D class defines a quadratic parametric curve segment in (x,y) coordinate space. |
| [**QuadCurve2D.Double**](http://docs.google.com/java/awt/geom/QuadCurve2D.Double.html) | A quadratic parametric curve segment specified with double coordinates. |
| [**QuadCurve2D.Float**](http://docs.google.com/java/awt/geom/QuadCurve2D.Float.html) | A quadratic parametric curve segment specified with float coordinates. |
| [**Rectangle2D**](http://docs.google.com/java/awt/geom/Rectangle2D.html) | The Rectangle2D class describes a rectangle defined by a location (x,y) and dimension (w x h). |
| [**Rectangle2D.Double**](http://docs.google.com/java/awt/geom/Rectangle2D.Double.html) | The Double class defines a rectangle specified in double coordinates. |
| [**Rectangle2D.Float**](http://docs.google.com/java/awt/geom/Rectangle2D.Float.html) | The Float class defines a rectangle specified in float coordinates. |
| [**RectangularShape**](http://docs.google.com/java/awt/geom/RectangularShape.html) | RectangularShape is the base class for a number of [Shape](http://docs.google.com/java/awt/Shape.html) objects whose geometry is defined by a rectangular frame. |
| [**RoundRectangle2D**](http://docs.google.com/java/awt/geom/RoundRectangle2D.html) | The RoundRectangle2D class defines a rectangle with rounded corners defined by a location (x,y), a dimension (w x h), and the width and height of an arc with which to round the corners. |
| [**RoundRectangle2D.Double**](http://docs.google.com/java/awt/geom/RoundRectangle2D.Double.html) | The Double class defines a rectangle with rounded corners all specified in double coordinates. |
| [**RoundRectangle2D.Float**](http://docs.google.com/java/awt/geom/RoundRectangle2D.Float.html) | The Float class defines a rectangle with rounded corners all specified in float coordinates. |

| **Exception Summary** | |
| --- | --- |
| [**IllegalPathStateException**](http://docs.google.com/java/awt/geom/IllegalPathStateException.html) | The IllegalPathStateException represents an exception that is thrown if an operation is performed on a path that is in an illegal state with respect to the particular operation being performed, such as appending a path segment to a [GeneralPath](http://docs.google.com/java/awt/geom/GeneralPath.html) without an initial moveto. |
| [**NoninvertibleTransformException**](http://docs.google.com/java/awt/geom/NoninvertibleTransformException.html) | The NoninvertibleTransformException class represents an exception that is thrown if an operation is performed requiring the inverse of an [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) object but the AffineTransform is in a non-invertible state. |

## Package java.awt.geom Description

Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry. Some important features of the package include:

* classes for manipulating geometry, such as AffineTransform and the PathIterator interface which is implemented by all Shape objects.
* classes that implement the Shape interface, such as CubicCurve2D, Ellipse2D, Line2D, Rectangle2D, and GeneralShape.
* the Area class which provides mechanisms for add (union), subtract, intersect, and exclusiveOR operations on other Shape objects.

**Since:** 1.2

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/java/awt/font/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/awt/im/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).