| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Ellipse2D.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/Dimension2D.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Ellipse2D.Double.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Ellipse2D.html)    [**NO FRAMES**](http://docs.google.com/Ellipse2D.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

## **java.awt.geom**

Class Ellipse2D

[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.Ellipse2D**

**All Implemented Interfaces:** [Shape](http://docs.google.com/java/awt/Shape.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [Ellipse2D.Double](http://docs.google.com/java/awt/geom/Ellipse2D.Double.html), [Ellipse2D.Float](http://docs.google.com/java/awt/geom/Ellipse2D.Float.html)

public abstract class **Ellipse2D**extends [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html)

The Ellipse2D class describes an ellipse that is defined by a framing rectangle.

This class is only the abstract superclass for all objects which store a 2D ellipse. The actual storage representation of the coordinates is left to the subclass.

**Since:** 1.2

| **Nested Class Summary** | |
| --- | --- |
| static class | [**Ellipse2D.Double**](http://docs.google.com/java/awt/geom/Ellipse2D.Double.html)            The Double class defines an ellipse specified in double precision. |
| static class | [**Ellipse2D.Float**](http://docs.google.com/java/awt/geom/Ellipse2D.Float.html)            The Float class defines an ellipse specified in float precision. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Ellipse2D**](http://docs.google.com/java/awt/geom/Ellipse2D.html#Ellipse2D())()            This is an abstract class that cannot be instantiated directly. |

| **Method Summary** | |
| --- | --- |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Ellipse2D.html#contains(double,%20double))(double x, double y)            Tests if the specified coordinates are inside the boundary of the Shape. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Ellipse2D.html#contains(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Tests if the interior of the Shape entirely contains the specified rectangular area. |
| boolean | [**equals**](http://docs.google.com/java/awt/geom/Ellipse2D.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 Ellipse2D. |
| [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) | [**getPathIterator**](http://docs.google.com/java/awt/geom/Ellipse2D.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 this Ellipse2D. |
| int | [**hashCode**](http://docs.google.com/java/awt/geom/Ellipse2D.html#hashCode())()            Returns the hashcode for this Ellipse2D. |
| boolean | [**intersects**](http://docs.google.com/java/awt/geom/Ellipse2D.html#intersects(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Tests if the interior of the Shape intersects the interior of a specified rectangular area. |

| **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)), [contains](http://docs.google.com/java/awt/geom/RectangularShape.html#contains(java.awt.geom.Rectangle2D)), [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(double,%20double,%20double,%20double)), [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)) |

| **Methods inherited from interface java.awt.**[**Shape**](http://docs.google.com/java/awt/Shape.html) |
| --- |
| [getBounds2D](http://docs.google.com/java/awt/Shape.html#getBounds2D()) |

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

### Ellipse2D

protected **Ellipse2D**()

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.

**Since:** 1.2 **See Also:**[Ellipse2D.Float](http://docs.google.com/java/awt/geom/Ellipse2D.Float.html), [Ellipse2D.Double](http://docs.google.com/java/awt/geom/Ellipse2D.Double.html)

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

### contains

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

Tests if the specified coordinates are inside the boundary of the Shape.

**Parameters:**x - the specified X coordinate to be testedy - the specified Y coordinate to be tested **Returns:**true if the specified coordinates are inside the Shape boundary; false otherwise.**Since:** 1.2

### intersects

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

Tests if the interior of the Shape intersects the interior of a specified rectangular area. The rectangular area is considered to intersect the Shape if any point is contained in both the interior of the Shape and the specified rectangular area.

The Shape.intersects() method allows a Shape implementation to conservatively return true when:

* there is a high probability that the rectangular area and the Shape intersect, but
* the calculations to accurately determine this intersection are prohibitively expensive.

This means that for some Shapes this method might return true even though the rectangular area does not intersect the Shape. The [Area](http://docs.google.com/java/awt/geom/Area.html) class performs more accurate computations of geometric intersection than most Shape objects and therefore can be used if a more precise answer is required.

**Parameters:**x - the X coordinate of the upper-left corner of the specified rectangular areay - the Y coordinate of the upper-left corner of the specified rectangular areaw - the width of the specified rectangular areah - the height of the specified rectangular area **Returns:**true if the interior of the Shape and the interior of the rectangular area intersect, or are both highly likely to intersect and intersection calculations would be too expensive to perform; false otherwise.**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)

Tests if the interior of the Shape entirely contains the specified rectangular area. All coordinates that lie inside the rectangular area must lie within the Shape for the entire rectanglar area to be considered contained within the Shape.

The Shape.contains() method allows a Shape implementation to conservatively return false when:

* the intersect method returns true and
* the calculations to determine whether or not the Shape entirely contains the rectangular area are prohibitively expensive.

This means that for some Shapes this method might return false even though the Shape contains the rectangular area. The [Area](http://docs.google.com/java/awt/geom/Area.html) class performs more accurate geometric computations than most Shape objects and therefore can be used if a more precise answer is required.

**Parameters:**x - the X coordinate of the upper-left corner of the specified rectangular areay - the Y coordinate of the upper-left corner of the specified rectangular areaw - the width of the specified rectangular areah - the height of the specified rectangular area **Returns:**true if the interior of the Shape entirely contains the specified rectangular area; false otherwise or, if the Shape contains the rectangular area and the intersects method returns true and the containment calculations would be too expensive to perform.**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))

### 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 this Ellipse2D. The iterator for this class is multi-threaded safe, which means that this Ellipse2D class guarantees that modifications to the geometry of this Ellipse2D object 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 untransformed coordinates are desired **Returns:**the PathIterator object that returns the geometry of the outline of this Ellipse2D, one segment at a time.**Since:** 1.2

### hashCode

public int **hashCode**()

Returns the hashcode for this Ellipse2D.

**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 Ellipse2D.**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 Ellipse2D. The specified Object is equal to this Ellipse2D if it is an instance of Ellipse2D and if its location and size are the same as this Ellipse2D.

**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 Ellipse2D. **Returns:**true if obj is an instance of Ellipse2D 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/Ellipse2D.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/Dimension2D.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Ellipse2D.Double.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Ellipse2D.html)    [**NO FRAMES**](http://docs.google.com/Ellipse2D.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

[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).