| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Size2DSyntax.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/javax/print/attribute/SetOfIntegerSyntax.html)   [**NEXT CLASS**](http://docs.google.com/javax/print/attribute/SupportedValuesAttribute.html) | [**FRAMES**](http://docs.google.com/index.html?javax/print/attribute/Size2DSyntax.html)    [**NO FRAMES**](http://docs.google.com/Size2DSyntax.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#lnxbz9) |

## **javax.print.attribute**

Class Size2DSyntax

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.print.attribute.Size2DSyntax**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [MediaSize](http://docs.google.com/javax/print/attribute/standard/MediaSize.html)

public abstract class **Size2DSyntax**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

Class Size2DSyntax is an abstract base class providing the common implementation of all attributes denoting a size in two dimensions.

A two-dimensional size attribute's value consists of two items, the X dimension and the Y dimension. A two-dimensional size attribute may be constructed by supplying the two values and indicating the units in which the values are measured. Methods are provided to return a two-dimensional size attribute's values, indicating the units in which the values are to be returned. The two most common size units are inches (in) and millimeters (mm), and exported constants [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) and [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM) are provided for indicating those units.

Once constructed, a two-dimensional size attribute's value is immutable.

**Design**

A two-dimensional size attribute's X and Y dimension values are stored internally as integers in units of micrometers (µm), where 1 micrometer = 10-6 meter = 1/1000 millimeter = 1/25400 inch. This permits dimensions to be represented exactly to a precision of 1/1000 mm (= 1 µm) or 1/100 inch (= 254 µm). If fractional inches are expressed in negative powers of two, this permits dimensions to be represented exactly to a precision of 1/8 inch (= 3175 µm) but not 1/16 inch (because 1/16 inch does not equal an integral number of µm).

Storing the dimensions internally in common units of µm lets two size attributes be compared without regard to the units in which they were created; for example, 8.5 in will compare equal to 215.9 mm, as they both are stored as 215900 µm. For example, a lookup service can match resolution attributes based on equality of their serialized representations regardless of the units in which they were created. Using integers for internal storage allows precise equality comparisons to be done, which would not be guaranteed if an internal floating point representation were used. Note that if you're looking for U.S. letter sized media in metric units, you have to search for a media size of 215.9 x 279.4 mm; rounding off to an integral 216 x 279 mm will not match.

The exported constant [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) is actually the conversion factor by which to multiply a value in inches to get the value in µm. Likewise, the exported constant [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM) is the conversion factor by which to multiply a value in mm to get the value in µm. A client can specify a resolution value in units other than inches or mm by supplying its own conversion factor. However, since the internal units of µm was chosen with supporting only the external units of inch and mm in mind, there is no guarantee that the conversion factor for the client's units will be an exact integer. If the conversion factor isn't an exact integer, resolution values in the client's units won't be stored precisely.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.print.attribute.Size2DSyntax)

| **Field Summary** | |
| --- | --- |
| static int | [**INCH**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH)            Value to indicate units of inches (in). |
| static int | [**MM**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM)            Value to indicate units of millimeters (mm). |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Size2DSyntax**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#Size2DSyntax(float,%20float,%20int))(float x, float y, int units)            Construct a new two-dimensional size attribute from the given floating-point values. |
| protected | [**Size2DSyntax**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#Size2DSyntax(int,%20int,%20int))(int x, int y, int units)            Construct a new two-dimensional size attribute from the given integer values. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) object)            Returns whether this two-dimensional size attribute is equivalent to the passed in object. |
| float[] | [**getSize**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#getSize(int))(int units)            Get this two-dimensional size attribute's dimensions in the given units as floating-point values. |
| float | [**getX**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#getX(int))(int units)            Returns this two-dimensional size attribute's X dimension in the given units as a floating-point value. |
| protected  int | [**getXMicrometers**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#getXMicrometers())()            Returns this two-dimensional size attribute's X dimension in units of micrometers (µm). |
| float | [**getY**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#getY(int))(int units)            Returns this two-dimensional size attribute's Y dimension in the given units as a floating-point value. |
| protected  int | [**getYMicrometers**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#getYMicrometers())()            Returns this two-dimensional size attribute's Y dimension in units of micrometers (µm). |
| int | [**hashCode**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#hashCode())()            Returns a hash code value for this two-dimensional size attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#toString())()            Returns a string version of this two-dimensional size attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#toString(int,%20java.lang.String))(int units, [String](http://docs.google.com/java/lang/String.html) unitsName)            Returns a string version of this two-dimensional size attribute in the given units. |

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

### INCH

public static final int **INCH**

Value to indicate units of inches (in). It is actually the conversion factor by which to multiply inches to yield µm (25400).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.print.attribute.Size2DSyntax.INCH)

### MM

public static final int **MM**

Value to indicate units of millimeters (mm). It is actually the conversion factor by which to multiply mm to yield µm (1000).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.print.attribute.Size2DSyntax.MM)

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

### Size2DSyntax

protected **Size2DSyntax**(float x,  
 float y,  
 int units)

Construct a new two-dimensional size attribute from the given floating-point values.

**Parameters:**x - X dimension.y - Y dimension.units - Unit conversion factor, e.g. [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) or [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (Unchecked exception) Thrown if x < 0 or y < 0 or units < 1.

### Size2DSyntax

protected **Size2DSyntax**(int x,  
 int y,  
 int units)

Construct a new two-dimensional size attribute from the given integer values.

**Parameters:**x - X dimension.y - Y dimension.units - Unit conversion factor, e.g. [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) or [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (Unchecked exception) Thrown if x < 0 or y < 0 or units < 1.

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

### getSize

public float[] **getSize**(int units)

Get this two-dimensional size attribute's dimensions in the given units as floating-point values.

**Parameters:**units - Unit conversion factor, e.g. [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) or [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM). **Returns:**A two-element array with the X dimension at index 0 and the Y dimension at index 1. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### getX

public float **getX**(int units)

Returns this two-dimensional size attribute's X dimension in the given units as a floating-point value.

**Parameters:**units - Unit conversion factor, e.g. [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) or [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM). **Returns:**X dimension. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### getY

public float **getY**(int units)

Returns this two-dimensional size attribute's Y dimension in the given units as a floating-point value.

**Parameters:**units - Unit conversion factor, e.g. [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) or [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM). **Returns:**Y dimension. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**(int units,  
 [String](http://docs.google.com/java/lang/String.html) unitsName)

Returns a string version of this two-dimensional size attribute in the given units. The string takes the form "*X*x*Y* *U*", where *X* is the X dimension, *Y* is the Y dimension, and *U* is the units name. The values are displayed in floating point.

**Parameters:**units - Unit conversion factor, e.g. [INCH](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#INCH) or [MM](http://docs.google.com/javax/print/attribute/Size2DSyntax.html#MM).unitsName - Units name string, e.g. "in" or "mm". If null, no units name is appended to the result. **Returns:**String version of this two-dimensional size attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### equals

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

Returns whether this two-dimensional size attribute is equivalent to the passed in object. To be equivalent, all of the following conditions must be true:

1. object is not null.
2. object is an instance of class Size2DSyntax.
3. This attribute's X dimension is equal to object's X dimension.
4. This attribute's Y dimension is equal to object's Y dimension.

**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:**object - Object to compare to. **Returns:**True if object is equivalent to this two-dimensional size attribute, false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns a hash code value for this two-dimensional size attribute.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**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)

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string version of this two-dimensional size attribute. The string takes the form "*X*x*Y* um", where *X* is the X dimension and *Y* is the Y dimension. The values are reported in the internal units of micrometers.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of the object.

### getXMicrometers

protected int **getXMicrometers**()

Returns this two-dimensional size attribute's X dimension in units of micrometers (µm). (For use in a subclass.)

**Returns:**X dimension (µm).

### getYMicrometers

protected int **getYMicrometers**()

Returns this two-dimensional size attribute's Y dimension in units of micrometers (µm). (For use in a subclass.)

**Returns:**Y dimension (µm).

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Size2DSyntax.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/javax/print/attribute/SetOfIntegerSyntax.html)   [**NEXT CLASS**](http://docs.google.com/javax/print/attribute/SupportedValuesAttribute.html) | [**FRAMES**](http://docs.google.com/index.html?javax/print/attribute/Size2DSyntax.html)    [**NO FRAMES**](http://docs.google.com/Size2DSyntax.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#lnxbz9) |

[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).