| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ResolutionSyntax.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/PrintServiceAttributeSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/print/attribute/SetOfIntegerSyntax.html) | [**FRAMES**](http://docs.google.com/index.html?javax/print/attribute/ResolutionSyntax.html)    [**NO FRAMES**](http://docs.google.com/ResolutionSyntax.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **javax.print.attribute**

Class ResolutionSyntax

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

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

public abstract class **ResolutionSyntax**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 ResolutionSyntax is an abstract base class providing the common implementation of all attributes denoting a printer resolution.

A resolution attribute's value consists of two items, the cross feed direction resolution and the feed direction resolution. A resolution 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 resolution attribute's values, indicating the units in which the values are to be returned. The two most common resolution units are dots per inch (dpi) and dots per centimeter (dpcm), and exported constants [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) and [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM) are provided for indicating those units.

Once constructed, a resolution attribute's value is immutable.

**Design**

A resolution attribute's cross feed direction resolution and feed direction resolution values are stored internally using units of dots per 100 inches (dphi). Storing the values in dphi rather than, say, metric units allows precise integer arithmetic conversions between dpi and dphi and between dpcm and dphi: 1 dpi = 100 dphi, 1 dpcm = 254 dphi. Thus, the values can be stored into and retrieved back from a resolution attribute in either units with no loss of precision. This would not be guaranteed if a floating point representation were used. However, roundoff error will in general occur if a resolution attribute's values are created in one units and retrieved in different units; for example, 600 dpi will be rounded to 236 dpcm, whereas the true value (to five figures) is 236.22 dpcm.

Storing the values internally in common units of dphi lets two resolution attributes be compared without regard to the units in which they were created; for example, 300 dpcm will compare equal to 762 dpi, as they both are stored as 76200 dphi. In particular, a lookup service can match resolution attributes based on equality of their serialized representations regardless of the units in which they were created. Again, using integers for internal storage allows precise equality comparisons to be done, which would not be guaranteed if a floating point representation were used.

The exported constant [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) is actually the conversion factor by which to multiply a value in dpi to get the value in dphi. Likewise, the exported constant [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM) is the conversion factor by which to multiply a value in dpcm to get the value in dphi. A client can specify a resolution value in units other than dpi or dpcm by supplying its own conversion factor. However, since the internal units of dphi was chosen with supporting only the external units of dpi and dpcm 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.ResolutionSyntax)

| **Field Summary** | |
| --- | --- |
| static int | [**DPCM**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM)            Value to indicate units of dots per centimeter (dpcm). |
| static int | [**DPI**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI)            Value to indicate units of dots per inch (dpi). |

| **Constructor Summary** | |
| --- | --- |
| [**ResolutionSyntax**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#ResolutionSyntax(int,%20int,%20int))(int crossFeedResolution, int feedResolution, int units)            Construct a new resolution attribute from the given items. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) object)            Returns whether this resolution attribute is equivalent to the passed in object. |
| int | [**getCrossFeedResolution**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#getCrossFeedResolution(int))(int units)            Returns this resolution attribute's cross feed direction resolution in the given units. |
| protected  int | [**getCrossFeedResolutionDphi**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#getCrossFeedResolutionDphi())()            Returns this resolution attribute's cross feed direction resolution in units of dphi. |
| int | [**getFeedResolution**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#getFeedResolution(int))(int units)            Returns this resolution attribute's feed direction resolution in the given units. |
| protected  int | [**getFeedResolutionDphi**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#getFeedResolutionDphi())()            Returns this resolution attribute's feed direction resolution in units of dphi. |
| int[] | [**getResolution**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#getResolution(int))(int units)            Get this resolution attribute's resolution values in the given units. |
| int | [**hashCode**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#hashCode())()            Returns a hash code value for this resolution attribute. |
| boolean | [**lessThanOrEquals**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#lessThanOrEquals(javax.print.attribute.ResolutionSyntax))([ResolutionSyntax](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html) other)            Determine whether this resolution attribute's value is less than or equal to the given resolution attribute's value. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#toString())()            Returns a string version of this resolution attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#toString(int,%20java.lang.String))(int units, [String](http://docs.google.com/java/lang/String.html) unitsName)            Returns a string version of this resolution 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** |
| --- |

### DPI

public static final int **DPI**

Value to indicate units of dots per inch (dpi). It is actually the conversion factor by which to multiply dpi to yield dphi (100).

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

### DPCM

public static final int **DPCM**

Value to indicate units of dots per centimeter (dpcm). It is actually the conversion factor by which to multiply dpcm to yield dphi (254).

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

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

### ResolutionSyntax

public **ResolutionSyntax**(int crossFeedResolution,  
 int feedResolution,  
 int units)

Construct a new resolution attribute from the given items.

**Parameters:**crossFeedResolution - Cross feed direction resolution.feedResolution - Feed direction resolution.units - Unit conversion factor, e.g. [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) or [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if crossFeedResolution < 1 or feedResolution < 1 or units < 1.

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

### getResolution

public int[] **getResolution**(int units)

Get this resolution attribute's resolution values in the given units. The values are rounded to the nearest integer.

**Parameters:**units - Unit conversion factor, e.g. [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) or [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM). **Returns:**A two-element array with the cross feed direction resolution at index 0 and the feed direction resolution at index 1. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### getCrossFeedResolution

public int **getCrossFeedResolution**(int units)

Returns this resolution attribute's cross feed direction resolution in the given units. The value is rounded to the nearest integer.

**Parameters:**units - Unit conversion factor, e.g. [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) or [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM). **Returns:**Cross feed direction resolution. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### getFeedResolution

public int **getFeedResolution**(int units)

Returns this resolution attribute's feed direction resolution in the given units. The value is rounded to the nearest integer.

**Parameters:**units - Unit conversion factor, e.g. [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) or [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM). **Returns:**Feed direction resolution. **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 resolution attribute in the given units. The string takes the form "*C*x*F* *U*", where *C* is the cross feed direction resolution, *F* is the feed direction resolution, and *U* is the units name. The values are rounded to the nearest integer.

**Parameters:**units - Unit conversion factor, e.g. [DPI](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPI) or [DPCM](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html#DPCM).unitsName - Units name string, e.g. "dpi" or "dpcm". If null, no units name is appended to the result. **Returns:**String version of this resolution attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - (unchecked exception) Thrown if units < 1.

### lessThanOrEquals

public boolean **lessThanOrEquals**([ResolutionSyntax](http://docs.google.com/javax/print/attribute/ResolutionSyntax.html) other)

Determine whether this resolution attribute's value is less than or equal to the given resolution attribute's value. This is true if all of the following conditions are true:

* This attribute's cross feed direction resolution is less than or equal to the other attribute's cross feed direction resolution.
* This attribute's feed direction resolution is less than or equal to the other attribute's feed direction resolution.

**Parameters:**other - Resolution attribute to compare with. **Returns:**True if this resolution attribute is less than or equal to the other resolution attribute, false otherwise. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - (unchecked exception) Thrown if other is null.

### equals

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

Returns whether this resolution 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 ResolutionSyntax.
3. This attribute's cross feed direction resolution is equal to object's cross feed direction resolution.
4. This attribute's feed direction resolution is equal to object's feed direction resolution.

**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 resolution 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 resolution 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 resolution attribute. The string takes the form "*C*x*F* dphi", where *C* is the cross feed direction resolution and *F* is the feed direction resolution. The values are reported in the internal units of dphi.

**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.

### getCrossFeedResolutionDphi

protected int **getCrossFeedResolutionDphi**()

Returns this resolution attribute's cross feed direction resolution in units of dphi. (For use in a subclass.)

**Returns:**Cross feed direction resolution.

### getFeedResolutionDphi

protected int **getFeedResolutionDphi**()

Returns this resolution attribute's feed direction resolution in units of dphi. (For use in a subclass.)

**Returns:**Feed direction resolution.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ResolutionSyntax.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/PrintServiceAttributeSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/print/attribute/SetOfIntegerSyntax.html) | [**FRAMES**](http://docs.google.com/index.html?javax/print/attribute/ResolutionSyntax.html)    [**NO FRAMES**](http://docs.google.com/ResolutionSyntax.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).