**Overview  Package  Class  Use  Tree  Deprecated  Index  Help**

**PREV CLASS   NEXT CLASS**
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES    NO FRAMES
DETAIL: FIELD | CONSTR | METHOD

*Java™ Platform*
*Standard Ed. 6*

**javax.swing.text**
# Class TabStop

java.lang.Object
 └ **javax.swing.text.TabStop**

**All Implemented Interfaces:**
>  Serializable

---

public class **TabStop**
extends Object
implements Serializable

This class encapsulates a single tab stop (basically as tab stops are thought of by RTF). A tab stop is at a specified distance from the left margin, aligns text in a specified way, and has a specified leader. TabStops are immutable, and usually contained in TabSets.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeans™ has been added to the java.beans package. Please see XMLEncoder.

---

## Field Summary

| | |
|---|---|
| static int | **ALIGN_BAR** |
| static int | **ALIGN_CENTER**<br>        Characters following tab are positioned such that all following characters up to next tab/newline are centered around the tabs location. |
| static int | **ALIGN_DECIMAL**<br>        Characters following tab are aligned such that next decimal/tab/newline is at the tab location, very similar to RIGHT_TAB, just includes decimal as additional character to look for. |
| static int | **ALIGN_LEFT**<br>        Character following tab is positioned at location. |
| static int | **ALIGN_RIGHT**<br>        Characters following tab are positioned such that all following characters up to next tab/newline end at location. |
| static int | **LEAD_DOTS** |
| static int | **LEAD_EQUALS** |
| static int | **LEAD_HYPHENS** |
| static int | **LEAD_NONE** |
| static int | **LEAD_THICKLINE** |
| static int | **LEAD_UNDERLINE** |

## Constructor Summary

| |
|---|
| **TabStop**(float pos)<br>        Creates a tab at position pos with a default alignment and default leader. |
| **TabStop**(float pos, int align, int leader)<br>        Creates a tab with the specified position pos, alignment align and leader leader. |

## Method Summary

| | |
|---|---|
| boolean | **equals**(Object other)<br>        Returns true if the tabs are equal. |
| int | **getAlignment**()<br>        Returns the alignment, as an integer, of the tab. |
| int | **getLeader**()<br>        Returns the leader of the tab. |
| float | **getPosition**()<br>        Returns the position, as a float, of the tab. |
| int | **hashCode**()<br>        Returns the hashCode for the object. |
| String | **toString**()<br>        Returns a string representation of the object. |

**Methods inherited from class java.lang.Object**

clone, finalize, getClass, notify, notifyAll, wait, wait, wait

---

## Field Detail

### ALIGN_LEFT

`public static final int ALIGN_LEFT`

Character following tab is positioned at location.

**See Also:**
Constant Field Values

### ALIGN_RIGHT

`public static final int ALIGN_RIGHT`

Characters following tab are positioned such that all following characters up to next tab/newline end at location.

**See Also:**
Constant Field Values

### ALIGN_CENTER

`public static final int ALIGN_CENTER`

Characters following tab are positioned such that all following characters up to next tab/newline are centered around the tabs location.

**See Also:**
Constant Field Values

### ALIGN_DECIMAL

`public static final int ALIGN_DECIMAL`

Characters following tab are aligned such that next decimal/tab/newline is at the tab location, very similar to RIGHT_TAB, just includes decimal as additional character to look for.

**See Also:**
Constant Field Values

### ALIGN_BAR

`public static final int ALIGN_BAR`

**See Also:**
Constant Field Values

### LEAD_NONE

`public static final int LEAD_NONE`

**See Also:**
Constant Field Values

### LEAD_DOTS

`public static final int LEAD_DOTS`

**See Also:**
Constant Field Values

### LEAD_HYPHENS

`public static final int LEAD_HYPHENS`

**See Also:**
Constant Field Values

### LEAD_UNDERLINE

`public static final int LEAD_UNDERLINE`

**See Also:**
Constant Field Values

### LEAD_THICKLINE

`public static final int LEAD_THICKLINE`

**See Also:**
Constant Field Values

**LEAD_EQUALS**

```
public static final int LEAD_EQUALS
```

**See Also:**
> Constant Field Values

# Constructor Detail

### TabStop

```
public TabStop(float pos)
```

> Creates a tab at position `pos` with a default alignment and default leader.

---

### TabStop

```
public TabStop(float pos,
               int align,
               int leader)
```

> Creates a tab with the specified position `pos`, alignment `align` and leader `leader`.

# Method Detail

### getPosition

```
public float getPosition()
```

> Returns the position, as a float, of the tab.
>
> **Returns:**
> > the position of the tab

---

### getAlignment

```
public int getAlignment()
```

> Returns the alignment, as an integer, of the tab.
>
> **Returns:**
> > the alignment of the tab

---

### getLeader

```
public int getLeader()
```

> Returns the leader of the tab.
>
> **Returns:**
> > the leader of the tab

---

### equals

```
public boolean equals(Object other)
```

> Returns true if the tabs are equal.
>
> **Overrides:**
> > equals in class Object
>
> **Parameters:**
> > other - the reference object with which to compare.
>
> **Returns:**
> > true if the tabs are equal, otherwise false
>
> **See Also:**
> > Object.hashCode(), Hashtable

---

### hashCode

```
public int hashCode()
```

> Returns the hashCode for the object. This must be defined here to ensure 100% pure.
>
> **Overrides:**
> > hashCode in class Object
>
> **Returns:**
> > the hashCode for the object
>
> **See Also:**
> > Object.equals(java.lang.Object), Hashtable

---

### toString

```
public String toString()
```

> **Description copied from class: `Object`**
> Returns a string representation of the object. In general, the `toString` method returns a string that "textually represents" this object. The result should be a concise but informative representation that is easy for a person to read. It is recommended that all subclasses override this method.
>
> The `toString` method for class `Object` returns a string consisting of the name of the class of which the object is an instance, the at-sign character `@', and the unsigned hexadecimal representation of the hash code of the object. In other words, this method returns a string equal to the value of:
>
> ```
> getClass().getName() + '@' + Integer.toHexString(hashCode())
> ```
>
> **Overrides:**
> > `toString` in class `Object`
> **Returns:**
> > a string representation of the object.

---

**Overview  Package  Class  Use  Tree  Deprecated  Index  Help**

**PREV CLASS  NEXT CLASS**
SUMMARY: NESTED | FIELD | CONSTR | METHOD

**FRAMES   NO FRAMES**
DETAIL: FIELD | CONSTR | METHOD

*Java™ Platform*
*Standard Ed. 6*

Submit a bug or feature
For further API reference and developer documentation, see Java SE Developer Documentation. That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.