Class java.awt.FontMetrics

```
java.lang.Object
   |
   +----java.awt.FontMetrics
```

---

public class **FontMetrics**
extends Object

A font metrics object. Note that the implementations of these methods are inefficient, they are usually overridden with more efficient toolkit specific implementations.

Version:
1.7 08/17/95
Author:
Jim Graham

Variable Index

o **font**
The actual font.

Constructor Index

o **FontMetrics**(Font)
Creates a new FontMetrics object with the specified font.

Method Index

o **bytesWidth**(byte[], int, int)
Returns the width of the specified array of bytes in this Font.
o **charWidth**(int)
Returns the width of the specified character in this Font.
o **charWidth**(char)
Returns the width of the specified character in this Font.
o **charsWidth**(char[], int, int)
Returns the width of the specified character array in this Font.

- **getAscent**()
  Gets the font ascent.
- **getDescent**()
  Gets the font descent.
- **getFont**()
  Gets the font.
- **getHeight**()
  Gets the total height of the font.
- **getLeading**()
  Gets the standard leading, or line spacing, for the font.
- **getMaxAdvance**()
  Gets the maximum advance width of any character in this Font.
- **getMaxAscent**()
  Gets the maximum ascent of all characters in this Font.
- **getMaxDecent**()
  Gets the maximum descent of all characters.
- **getWidths**()
  Gets the widths of the first 256 characters in the Font.
- **stringWidth**(String)
  Returns the width of the specified String in this Font.
- **toString**()
  Returns the String representation of this FontMetric's values.

Variables

- **font**

```
protected Font font
```

  The actual font.
  See Also:
    getFont

Constructors

- **FontMetrics**

```
protected FontMetrics(Font font)
```

  Creates a new FontMetrics object with the specified font.
  Parameters:
    font – the font
  See Also:
    Font

Methods

o **getFont**

```
public Font getFont()
```

Gets the font.

o **getLeading**

```
public int getLeading()
```

Gets the standard leading, or line spacing, for the font. This is the logical amount of space to be reserved between the descent of one line of text and the ascent of the next line. The height metric is calculated to include this extra space.

o **getAscent**

```
public int getAscent()
```

Gets the font ascent. The font ascent is the distance from the base line to the top of the characters.
See Also:
getMaxAscent

o **getDescent**

```
public int getDescent()
```

Gets the font descent. The font descent is the distance from the base line to the bottom of the characters.
See Also:
getMaxDescent

o **getHeight**

```
public int getHeight()
```

Gets the total height of the font. This is the distance between the baseline of adjacent lines of text. It is the sum of the leading + ascent + descent.

o **getMaxAscent**

```
public int getMaxAscent()
```

Gets the maximum ascent of all characters in this Font. No character will extend further above the baseline than this metric.
See Also:
getAscent

o **getMaxDecent**

```
public int getMaxDecent()
```

Gets the maximum descent of all characters. No character will descend futher below the baseline than this metric.

See Also:

getDescent

o **getMaxAdvance**

```
public int getMaxAdvance()
```

Gets the maximum advance width of any character in this Font.

Returns:

−1 if the max advance is not known.

o **charWidth**

```
public int charWidth(int ch)
```

Returns the width of the specified character in this Font.

Parameters:

ch − the specified font

See Also:

stringWidth

o **charWidth**

```
public int charWidth(char ch)
```

Returns the width of the specified character in this Font.

Parameters:

ch − the specified font

See Also:

stringWidth

o **stringWidth**

```
public int stringWidth(String str)
```

Returns the width of the specified String in this Font.

Parameters:

str − the String to be checked

See Also:

charsWidth, bytesWidth

o **charsWidth**

```
public int charsWidth(char data[],
                      int off,
                      int len)
```

Returns the width of the specified character array in this Font.

Parameters:

data – the data to be checked
off – the start offset of the data
len – the maximum number of bytes checked

See Also:

stringWidth, bytesWidth

o **bytesWidth**

```
public int bytesWidth(byte data[],
                      int off,
                      int len)
```

Returns the width of the specified array of bytes in this Font.

Parameters:

data – the data to be checked
off – the start offset of the data
len – the maximum number of bytes checked

See Also:

stringWidth, charsWidth

o **getWidths**

```
public int[] getWidths()
```

Gets the widths of the first 256 characters in the Font.

o **toString**

```
public String toString()
```

Returns the String representation of this FontMetric's values.

Overrides:

toString in class Object