| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MultiFileChooserUI.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/swing/plaf/multi/MultiDesktopPaneUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/multi/MultiInternalFrameUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/multi/MultiFileChooserUI.html)    [**NO FRAMES**](http://docs.google.com/MultiFileChooserUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **javax.swing.plaf.multi**

Class MultiFileChooserUI

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.plaf.ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)  
 [javax.swing.plaf.FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html)  
 **javax.swing.plaf.multi.MultiFileChooserUI**

public class **MultiFileChooserUI**extends [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html)

A multiplexing UI used to combine FileChooserUIs.

This file was automatically generated by AutoMulti.

| **Field Summary** | |
| --- | --- |
| protected  [Vector](http://docs.google.com/java/util/Vector.html) | [**uis**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#uis)            The vector containing the real UIs. |

| **Constructor Summary** | |
| --- | --- |
| [**MultiFileChooserUI**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#MultiFileChooserUI())() |

| **Method Summary** | |
| --- | --- |
| boolean | [**contains**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#contains(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a, int b, int c)            Invokes the contains method on each UI handled by this object. |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Returns a multiplexing UI instance if any of the auxiliary LookAndFeels supports this UI. |
| void | [**ensureFileIsVisible**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#ensureFileIsVisible(javax.swing.JFileChooser,%20java.io.File))([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a, [File](http://docs.google.com/java/io/File.html) b)            Invokes the ensureFileIsVisible method on each UI handled by this object. |
| [FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) | [**getAcceptAllFileFilter**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getAcceptAllFileFilter(javax.swing.JFileChooser))([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)            Invokes the getAcceptAllFileFilter method on each UI handled by this object. |
| [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) | [**getAccessibleChild**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getAccessibleChild(javax.swing.JComponent,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a, int b)            Invokes the getAccessibleChild method on each UI handled by this object. |
| int | [**getAccessibleChildrenCount**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getAccessibleChildrenCount(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Invokes the getAccessibleChildrenCount method on each UI handled by this object. |
| [String](http://docs.google.com/java/lang/String.html) | [**getApproveButtonText**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getApproveButtonText(javax.swing.JFileChooser))([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)            Invokes the getApproveButtonText method on each UI handled by this object. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDialogTitle**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getDialogTitle(javax.swing.JFileChooser))([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)            Invokes the getDialogTitle method on each UI handled by this object. |
| [FileView](http://docs.google.com/javax/swing/filechooser/FileView.html) | [**getFileView**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getFileView(javax.swing.JFileChooser))([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)            Invokes the getFileView method on each UI handled by this object. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getMaximumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Invokes the getMaximumSize method on each UI handled by this object. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getMinimumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Invokes the getMinimumSize method on each UI handled by this object. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Invokes the getPreferredSize method on each UI handled by this object. |
| [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)[] | [**getUIs**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#getUIs())()            Returns the list of UIs associated with this multiplexing UI. |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Invokes the installUI method on each UI handled by this object. |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) a, [JComponent](http://docs.google.com/javax/swing/JComponent.html) b)            Invokes the paint method on each UI handled by this object. |
| void | [**rescanCurrentDirectory**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#rescanCurrentDirectory(javax.swing.JFileChooser))([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)            Invokes the rescanCurrentDirectory method on each UI handled by this object. |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#uninstallUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)            Invokes the uninstallUI method on each UI handled by this object. |
| void | [**update**](http://docs.google.com/javax/swing/plaf/multi/MultiFileChooserUI.html#update(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) a, [JComponent](http://docs.google.com/javax/swing/JComponent.html) b)            Invokes the update method on each UI handled by this object. |

| **Methods inherited from class javax.swing.plaf.**[**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html) |
| --- |
| [getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)) |

| **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()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [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)) |

| **Field Detail** |
| --- |

### uis

protected [Vector](http://docs.google.com/java/util/Vector.html) **uis**

The vector containing the real UIs. This is populated in the call to createUI, and can be obtained by calling the getUIs method. The first element is guaranteed to be the real UI obtained from the default look and feel.

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

### MultiFileChooserUI

public **MultiFileChooserUI**()

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

### getUIs

public [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)[] **getUIs**()

Returns the list of UIs associated with this multiplexing UI. This allows processing of the UIs by an application aware of multiplexing UIs on components.

### getAcceptAllFileFilter

public [FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) **getAcceptAllFileFilter**([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)

Invokes the getAcceptAllFileFilter method on each UI handled by this object.

**Specified by:**[getAcceptAllFileFilter](http://docs.google.com/javax/swing/plaf/FileChooserUI.html#getAcceptAllFileFilter(javax.swing.JFileChooser)) in class [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html) **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel

### getFileView

public [FileView](http://docs.google.com/javax/swing/filechooser/FileView.html) **getFileView**([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)

Invokes the getFileView method on each UI handled by this object.

**Specified by:**[getFileView](http://docs.google.com/javax/swing/plaf/FileChooserUI.html#getFileView(javax.swing.JFileChooser)) in class [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html) **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel

### getApproveButtonText

public [String](http://docs.google.com/java/lang/String.html) **getApproveButtonText**([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)

Invokes the getApproveButtonText method on each UI handled by this object.

**Specified by:**[getApproveButtonText](http://docs.google.com/javax/swing/plaf/FileChooserUI.html#getApproveButtonText(javax.swing.JFileChooser)) in class [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html) **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel

### getDialogTitle

public [String](http://docs.google.com/java/lang/String.html) **getDialogTitle**([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)

Invokes the getDialogTitle method on each UI handled by this object.

**Specified by:**[getDialogTitle](http://docs.google.com/javax/swing/plaf/FileChooserUI.html#getDialogTitle(javax.swing.JFileChooser)) in class [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html) **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel

### rescanCurrentDirectory

public void **rescanCurrentDirectory**([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a)

Invokes the rescanCurrentDirectory method on each UI handled by this object.

**Specified by:**[rescanCurrentDirectory](http://docs.google.com/javax/swing/plaf/FileChooserUI.html#rescanCurrentDirectory(javax.swing.JFileChooser)) in class [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html)

### ensureFileIsVisible

public void **ensureFileIsVisible**([JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html) a,  
 [File](http://docs.google.com/java/io/File.html) b)

Invokes the ensureFileIsVisible method on each UI handled by this object.

**Specified by:**[ensureFileIsVisible](http://docs.google.com/javax/swing/plaf/FileChooserUI.html#ensureFileIsVisible(javax.swing.JFileChooser,%20java.io.File)) in class [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html)

### contains

public boolean **contains**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a,  
 int b,  
 int c)

Invokes the contains method on each UI handled by this object.

**Overrides:**[contains](http://docs.google.com/javax/swing/plaf/ComponentUI.html#contains(javax.swing.JComponent,%20int,%20int)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the component where the *x,y* location is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple componentsb - the *x* coordinate of the pointc - the *y* coordinate of the point **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel**See Also:**[JComponent.contains(int, int)](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [Component.contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int))

### update

public void **update**([Graphics](http://docs.google.com/java/awt/Graphics.html) a,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) b)

Invokes the update method on each UI handled by this object.

**Overrides:**[update](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the Graphics context in which to paintb - the component being painted; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.paint(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)), [JComponent.paintComponent(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics))

### createUI

public static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **createUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Returns a multiplexing UI instance if any of the auxiliary LookAndFeels supports this UI. Otherwise, just returns the UI object obtained from the default LookAndFeel.

### installUI

public void **installUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Invokes the installUI method on each UI handled by this object.

**Overrides:**[installUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the component where this UI delegate is being installed**See Also:**[ComponentUI.uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)), [JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### uninstallUI

public void **uninstallUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Invokes the uninstallUI method on each UI handled by this object.

**Overrides:**[uninstallUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the component from which this UI delegate is being removed; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### paint

public void **paint**([Graphics](http://docs.google.com/java/awt/Graphics.html) a,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) b)

Invokes the paint method on each UI handled by this object.

**Overrides:**[paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the Graphics context in which to paintb - the component being painted; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.update(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent))

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Invokes the getPreferredSize method on each UI handled by this object.

**Overrides:**[getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the component whose preferred size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### getMinimumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMinimumSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Invokes the getMinimumSize method on each UI handled by this object.

**Overrides:**[getMinimumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the component whose minimum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel**See Also:**[JComponent.getMinimumSize()](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [LayoutManager.minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)), [ComponentUI.getPreferredSize(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent))

### getMaximumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMaximumSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Invokes the getMaximumSize method on each UI handled by this object.

**Overrides:**[getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**a - the component whose maximum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel**See Also:**[JComponent.getMaximumSize()](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [LayoutManager2.maximumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container))

### getAccessibleChildrenCount

public int **getAccessibleChildrenCount**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a)

Invokes the getAccessibleChildrenCount method on each UI handled by this object.

**Overrides:**[getAccessibleChildrenCount](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChildrenCount(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel**See Also:**[ComponentUI.getAccessibleChild(javax.swing.JComponent, int)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChild(javax.swing.JComponent,%20int))

### getAccessibleChild

public [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) **getAccessibleChild**([JComponent](http://docs.google.com/javax/swing/JComponent.html) a,  
 int b)

Invokes the getAccessibleChild method on each UI handled by this object.

**Overrides:**[getAccessibleChild](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChild(javax.swing.JComponent,%20int)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) b - zero-based index of child **Returns:**the value obtained from the first UI, which is the UI obtained from the default LookAndFeel**See Also:**[ComponentUI.getAccessibleChildrenCount(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChildrenCount(javax.swing.JComponent))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MultiFileChooserUI.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/swing/plaf/multi/MultiDesktopPaneUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/multi/MultiInternalFrameUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/multi/MultiFileChooserUI.html)    [**NO FRAMES**](http://docs.google.com/MultiFileChooserUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [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).