| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JFileChooser.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/JEditorPane.JEditorPaneAccessibleHypertextSupport.HTMLLink.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JFileChooser.AccessibleJFileChooser.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JFileChooser.html)    [**NO FRAMES**](http://docs.google.com/JFileChooser.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#37m2jsg) | [METHOD](#4k668n3) |

## **javax.swing**

Class JFileChooser

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.Component](http://docs.google.com/java/awt/Component.html)  
 [java.awt.Container](http://docs.google.com/java/awt/Container.html)  
 [javax.swing.JComponent](http://docs.google.com/javax/swing/JComponent.html)  
 **javax.swing.JFileChooser**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

public class **JFileChooser**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

JFileChooser provides a simple mechanism for the user to choose a file. For information about using JFileChooser, see [How to Use File Choosers](http://java.sun.com/docs/books/tutorial/uiswing/components/filechooser.html), a section in *The Java Tutorial*.

The following code pops up a file chooser for the user's home directory that sees only .jpg and .gif images:

JFileChooser chooser = new JFileChooser();  
 FileNameExtensionFilter filter = new FileNameExtensionFilter(  
 "JPG & GIF Images", "jpg", "gif");  
 chooser.setFileFilter(filter);  
 int returnVal = chooser.showOpenDialog(parent);  
 if(returnVal == JFileChooser.APPROVE\_OPTION) {  
 System.out.println("You chose to open this file: " +  
 chooser.getSelectedFile().getName());  
 }

**Warning:** Swing is not thread safe. For more information see [Swing's Threading Policy](http://docs.google.com/package-summary.html#threading).

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JFileChooser.AccessibleJFileChooser**](http://docs.google.com/javax/swing/JFileChooser.AccessibleJFileChooser.html)            This class implements accessibility support for the JFileChooser class. |

| **Nested classes/interfaces inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [JComponent.AccessibleJComponent](http://docs.google.com/javax/swing/JComponent.AccessibleJComponent.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**ACCEPT\_ALL\_FILE\_FILTER\_USED\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#ACCEPT_ALL_FILE_FILTER_USED_CHANGED_PROPERTY)            Identifies whether a the AcceptAllFileFilter is used or not. |
| protected  [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**accessibleContext**](http://docs.google.com/javax/swing/JFileChooser.html#accessibleContext) |
| static [String](http://docs.google.com/java/lang/String.html) | [**ACCESSORY\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#ACCESSORY_CHANGED_PROPERTY)            Says that a different accessory component is in use (for example, to preview files). |
| static [String](http://docs.google.com/java/lang/String.html) | [**APPROVE\_BUTTON\_MNEMONIC\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#APPROVE_BUTTON_MNEMONIC_CHANGED_PROPERTY)            Identifies change in the mnemonic for the approve (yes, ok) button. |
| static [String](http://docs.google.com/java/lang/String.html) | [**APPROVE\_BUTTON\_TEXT\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#APPROVE_BUTTON_TEXT_CHANGED_PROPERTY)            Identifies change in the text on the approve (yes, ok) button. |
| static [String](http://docs.google.com/java/lang/String.html) | [**APPROVE\_BUTTON\_TOOL\_TIP\_TEXT\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#APPROVE_BUTTON_TOOL_TIP_TEXT_CHANGED_PROPERTY)            Identifies change in the tooltip text for the approve (yes, ok) button. |
| static int | [**APPROVE\_OPTION**](http://docs.google.com/javax/swing/JFileChooser.html#APPROVE_OPTION)            Return value if approve (yes, ok) is chosen. |
| static [String](http://docs.google.com/java/lang/String.html) | [**APPROVE\_SELECTION**](http://docs.google.com/javax/swing/JFileChooser.html#APPROVE_SELECTION)            Instruction to approve the current selection (same as pressing yes or ok). |
| static int | [**CANCEL\_OPTION**](http://docs.google.com/javax/swing/JFileChooser.html#CANCEL_OPTION)            Return value if cancel is chosen. |
| static [String](http://docs.google.com/java/lang/String.html) | [**CANCEL\_SELECTION**](http://docs.google.com/javax/swing/JFileChooser.html#CANCEL_SELECTION)            Instruction to cancel the current selection. |
| static [String](http://docs.google.com/java/lang/String.html) | [**CHOOSABLE\_FILE\_FILTER\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#CHOOSABLE_FILE_FILTER_CHANGED_PROPERTY)            Identifies a change in the list of predefined file filters the user can choose from. |
| static [String](http://docs.google.com/java/lang/String.html) | [**CONTROL\_BUTTONS\_ARE\_SHOWN\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#CONTROL_BUTTONS_ARE_SHOWN_CHANGED_PROPERTY)            Instruction to display the control buttons. |
| static int | [**CUSTOM\_DIALOG**](http://docs.google.com/javax/swing/JFileChooser.html#CUSTOM_DIALOG)            Type value indicating that the JFileChooser supports a developer-specified file operation. |
| static [String](http://docs.google.com/java/lang/String.html) | [**DIALOG\_TITLE\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#DIALOG_TITLE_CHANGED_PROPERTY)            Identifies a change in the dialog title. |
| static [String](http://docs.google.com/java/lang/String.html) | [**DIALOG\_TYPE\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#DIALOG_TYPE_CHANGED_PROPERTY)            Identifies a change in the type of files displayed (files only, directories only, or both files and directories). |
| static int | [**DIRECTORIES\_ONLY**](http://docs.google.com/javax/swing/JFileChooser.html#DIRECTORIES_ONLY)            Instruction to display only directories. |
| static [String](http://docs.google.com/java/lang/String.html) | [**DIRECTORY\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#DIRECTORY_CHANGED_PROPERTY)            Identifies user's directory change. |
| static int | [**ERROR\_OPTION**](http://docs.google.com/javax/swing/JFileChooser.html#ERROR_OPTION)            Return value if an error occured. |
| static [String](http://docs.google.com/java/lang/String.html) | [**FILE\_FILTER\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#FILE_FILTER_CHANGED_PROPERTY)            User changed the kind of files to display. |
| static [String](http://docs.google.com/java/lang/String.html) | [**FILE\_HIDING\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#FILE_HIDING_CHANGED_PROPERTY)            Identifies a change in the display-hidden-files property. |
| static [String](http://docs.google.com/java/lang/String.html) | [**FILE\_SELECTION\_MODE\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#FILE_SELECTION_MODE_CHANGED_PROPERTY)            Identifies a change in the kind of selection (single, multiple, etc.). |
| static [String](http://docs.google.com/java/lang/String.html) | [**FILE\_SYSTEM\_VIEW\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#FILE_SYSTEM_VIEW_CHANGED_PROPERTY)            Says that a different object is being used to find available drives on the system. |
| static [String](http://docs.google.com/java/lang/String.html) | [**FILE\_VIEW\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#FILE_VIEW_CHANGED_PROPERTY)            Says that a different object is being used to retrieve file information. |
| static int | [**FILES\_AND\_DIRECTORIES**](http://docs.google.com/javax/swing/JFileChooser.html#FILES_AND_DIRECTORIES)            Instruction to display both files and directories. |
| static int | [**FILES\_ONLY**](http://docs.google.com/javax/swing/JFileChooser.html#FILES_ONLY)            Instruction to display only files. |
| static [String](http://docs.google.com/java/lang/String.html) | [**MULTI\_SELECTION\_ENABLED\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#MULTI_SELECTION_ENABLED_CHANGED_PROPERTY)            Enables multiple-file selections. |
| static int | [**OPEN\_DIALOG**](http://docs.google.com/javax/swing/JFileChooser.html#OPEN_DIALOG)            Type value indicating that the JFileChooser supports an "Open" file operation. |
| static int | [**SAVE\_DIALOG**](http://docs.google.com/javax/swing/JFileChooser.html#SAVE_DIALOG)            Type value indicating that the JFileChooser supports a "Save" file operation. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SELECTED\_FILE\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#SELECTED_FILE_CHANGED_PROPERTY)            Identifies change in user's single-file selection. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SELECTED\_FILES\_CHANGED\_PROPERTY**](http://docs.google.com/javax/swing/JFileChooser.html#SELECTED_FILES_CHANGED_PROPERTY)            Identifies change in user's multiple-file selection. |

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

| **Fields inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [BOTTOM\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT), [CENTER\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT), [LEFT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT), [RIGHT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT), [TOP\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT) |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| [**JFileChooser**](http://docs.google.com/javax/swing/JFileChooser.html#JFileChooser())()            Constructs a JFileChooser pointing to the user's default directory. |
| [**JFileChooser**](http://docs.google.com/javax/swing/JFileChooser.html#JFileChooser(java.io.File))([File](http://docs.google.com/java/io/File.html) currentDirectory)            Constructs a JFileChooser using the given File as the path. |
| [**JFileChooser**](http://docs.google.com/javax/swing/JFileChooser.html#JFileChooser(java.io.File,%20javax.swing.filechooser.FileSystemView))([File](http://docs.google.com/java/io/File.html) currentDirectory, [FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)            Constructs a JFileChooser using the given current directory and FileSystemView. |
| [**JFileChooser**](http://docs.google.com/javax/swing/JFileChooser.html#JFileChooser(javax.swing.filechooser.FileSystemView))([FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)            Constructs a JFileChooser using the given FileSystemView. |
| [**JFileChooser**](http://docs.google.com/javax/swing/JFileChooser.html#JFileChooser(java.lang.String))([String](http://docs.google.com/java/lang/String.html) currentDirectoryPath)            Constructs a JFileChooser using the given path. |
| [**JFileChooser**](http://docs.google.com/javax/swing/JFileChooser.html#JFileChooser(java.lang.String,%20javax.swing.filechooser.FileSystemView))([String](http://docs.google.com/java/lang/String.html) currentDirectoryPath, [FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)            Constructs a JFileChooser using the given current directory path and FileSystemView. |

| **Method Summary** | |
| --- | --- |
| boolean | [**accept**](http://docs.google.com/javax/swing/JFileChooser.html#accept(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Returns true if the file should be displayed. |
| void | [**addActionListener**](http://docs.google.com/javax/swing/JFileChooser.html#addActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Adds an ActionListener to the file chooser. |
| void | [**addChoosableFileFilter**](http://docs.google.com/javax/swing/JFileChooser.html#addChoosableFileFilter(javax.swing.filechooser.FileFilter))([FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) filter)            Adds a filter to the list of user choosable file filters. |
| void | [**approveSelection**](http://docs.google.com/javax/swing/JFileChooser.html#approveSelection())()            Called by the UI when the user hits the Approve button (labeled "Open" or "Save", by default). |
| void | [**cancelSelection**](http://docs.google.com/javax/swing/JFileChooser.html#cancelSelection())()            Called by the UI when the user chooses the Cancel button. |
| void | [**changeToParentDirectory**](http://docs.google.com/javax/swing/JFileChooser.html#changeToParentDirectory())()            Changes the directory to be set to the parent of the current directory. |
| protected  [JDialog](http://docs.google.com/javax/swing/JDialog.html) | [**createDialog**](http://docs.google.com/javax/swing/JFileChooser.html#createDialog(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) parent)            Creates and returns a new JDialog wrapping this centered on the parent in the parent's frame. |
| void | [**ensureFileIsVisible**](http://docs.google.com/javax/swing/JFileChooser.html#ensureFileIsVisible(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Makes sure that the specified file is viewable, and not hidden. |
| protected  void | [**fireActionPerformed**](http://docs.google.com/javax/swing/JFileChooser.html#fireActionPerformed(java.lang.String))([String](http://docs.google.com/java/lang/String.html) command)            Notifies all listeners that have registered interest for notification on this event type. |
| [FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) | [**getAcceptAllFileFilter**](http://docs.google.com/javax/swing/JFileChooser.html#getAcceptAllFileFilter())()            Returns the AcceptAll file filter. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JFileChooser.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JFileChooser. |
| [JComponent](http://docs.google.com/javax/swing/JComponent.html) | [**getAccessory**](http://docs.google.com/javax/swing/JFileChooser.html#getAccessory())()            Returns the accessory component. |
| [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] | [**getActionListeners**](http://docs.google.com/javax/swing/JFileChooser.html#getActionListeners())()            Returns an array of all the action listeners registered on this file chooser. |
| int | [**getApproveButtonMnemonic**](http://docs.google.com/javax/swing/JFileChooser.html#getApproveButtonMnemonic())()            Returns the approve button's mnemonic. |
| [String](http://docs.google.com/java/lang/String.html) | [**getApproveButtonText**](http://docs.google.com/javax/swing/JFileChooser.html#getApproveButtonText())()            Returns the text used in the ApproveButton in the FileChooserUI. |
| [String](http://docs.google.com/java/lang/String.html) | [**getApproveButtonToolTipText**](http://docs.google.com/javax/swing/JFileChooser.html#getApproveButtonToolTipText())()            Returns the tooltip text used in the ApproveButton. |
| [FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html)[] | [**getChoosableFileFilters**](http://docs.google.com/javax/swing/JFileChooser.html#getChoosableFileFilters())()            Gets the list of user choosable file filters. |
| boolean | [**getControlButtonsAreShown**](http://docs.google.com/javax/swing/JFileChooser.html#getControlButtonsAreShown())()            Returns the value of the controlButtonsAreShown property. |
| [File](http://docs.google.com/java/io/File.html) | [**getCurrentDirectory**](http://docs.google.com/javax/swing/JFileChooser.html#getCurrentDirectory())()            Returns the current directory. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDescription**](http://docs.google.com/javax/swing/JFileChooser.html#getDescription(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Returns the file description. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDialogTitle**](http://docs.google.com/javax/swing/JFileChooser.html#getDialogTitle())()            Gets the string that goes in the JFileChooser's titlebar. |
| int | [**getDialogType**](http://docs.google.com/javax/swing/JFileChooser.html#getDialogType())()            Returns the type of this dialog. |
| boolean | [**getDragEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#getDragEnabled())()            Gets the value of the dragEnabled property. |
| [FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) | [**getFileFilter**](http://docs.google.com/javax/swing/JFileChooser.html#getFileFilter())()            Returns the currently selected file filter. |
| int | [**getFileSelectionMode**](http://docs.google.com/javax/swing/JFileChooser.html#getFileSelectionMode())()            Returns the current file-selection mode. |
| [FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) | [**getFileSystemView**](http://docs.google.com/javax/swing/JFileChooser.html#getFileSystemView())()            Returns the file system view. |
| [FileView](http://docs.google.com/javax/swing/filechooser/FileView.html) | [**getFileView**](http://docs.google.com/javax/swing/JFileChooser.html#getFileView())()            Returns the current file view. |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getIcon**](http://docs.google.com/javax/swing/JFileChooser.html#getIcon(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Returns the icon for this file or type of file, depending on the system. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/javax/swing/JFileChooser.html#getName(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Returns the filename. |
| [File](http://docs.google.com/java/io/File.html) | [**getSelectedFile**](http://docs.google.com/javax/swing/JFileChooser.html#getSelectedFile())()            Returns the selected file. |
| [File](http://docs.google.com/java/io/File.html)[] | [**getSelectedFiles**](http://docs.google.com/javax/swing/JFileChooser.html#getSelectedFiles())()            Returns a list of selected files if the file chooser is set to allow multiple selection. |
| [String](http://docs.google.com/java/lang/String.html) | [**getTypeDescription**](http://docs.google.com/javax/swing/JFileChooser.html#getTypeDescription(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Returns the file type. |
| [FileChooserUI](http://docs.google.com/javax/swing/plaf/FileChooserUI.html) | [**getUI**](http://docs.google.com/javax/swing/JFileChooser.html#getUI())()            Gets the UI object which implements the L&F for this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JFileChooser.html#getUIClassID())()            Returns a string that specifies the name of the L&F class that renders this component. |
| boolean | [**isAcceptAllFileFilterUsed**](http://docs.google.com/javax/swing/JFileChooser.html#isAcceptAllFileFilterUsed())()            Returns whether the AcceptAll FileFilter is used. |
| boolean | [**isDirectorySelectionEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#isDirectorySelectionEnabled())()            Convenience call that determines if directories are selectable based on the current file selection mode. |
| boolean | [**isFileHidingEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#isFileHidingEnabled())()            Returns true if hidden files are not shown in the file chooser; otherwise, returns false. |
| boolean | [**isFileSelectionEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#isFileSelectionEnabled())()            Convenience call that determines if files are selectable based on the current file selection mode. |
| boolean | [**isMultiSelectionEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#isMultiSelectionEnabled())()            Returns true if multiple files can be selected. |
| boolean | [**isTraversable**](http://docs.google.com/javax/swing/JFileChooser.html#isTraversable(java.io.File))([File](http://docs.google.com/java/io/File.html) f)            Returns true if the file (directory) can be visited. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JFileChooser.html#paramString())()            Returns a string representation of this JFileChooser. |
| void | [**removeActionListener**](http://docs.google.com/javax/swing/JFileChooser.html#removeActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Removes an ActionListener from the file chooser. |
| boolean | [**removeChoosableFileFilter**](http://docs.google.com/javax/swing/JFileChooser.html#removeChoosableFileFilter(javax.swing.filechooser.FileFilter))([FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) f)            Removes a filter from the list of user choosable file filters. |
| void | [**rescanCurrentDirectory**](http://docs.google.com/javax/swing/JFileChooser.html#rescanCurrentDirectory())()            Tells the UI to rescan its files list from the current directory. |
| void | [**resetChoosableFileFilters**](http://docs.google.com/javax/swing/JFileChooser.html#resetChoosableFileFilters())()            Resets the choosable file filter list to its starting state. |
| void | [**setAcceptAllFileFilterUsed**](http://docs.google.com/javax/swing/JFileChooser.html#setAcceptAllFileFilterUsed(boolean))(boolean b)            Determines whether the AcceptAll FileFilter is used as an available choice in the choosable filter list. |
| void | [**setAccessory**](http://docs.google.com/javax/swing/JFileChooser.html#setAccessory(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) newAccessory)            Sets the accessory component. |
| void | [**setApproveButtonMnemonic**](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonMnemonic(char))(char mnemonic)            Sets the approve button's mnemonic using a character. |
| void | [**setApproveButtonMnemonic**](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonMnemonic(int))(int mnemonic)            Sets the approve button's mnemonic using a numeric keycode. |
| void | [**setApproveButtonText**](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) approveButtonText)            Sets the text used in the ApproveButton in the FileChooserUI. |
| void | [**setApproveButtonToolTipText**](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonToolTipText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) toolTipText)            Sets the tooltip text used in the ApproveButton. |
| void | [**setControlButtonsAreShown**](http://docs.google.com/javax/swing/JFileChooser.html#setControlButtonsAreShown(boolean))(boolean b)            Sets the property that indicates whether the *approve* and *cancel* buttons are shown in the file chooser. |
| void | [**setCurrentDirectory**](http://docs.google.com/javax/swing/JFileChooser.html#setCurrentDirectory(java.io.File))([File](http://docs.google.com/java/io/File.html) dir)            Sets the current directory. |
| void | [**setDialogTitle**](http://docs.google.com/javax/swing/JFileChooser.html#setDialogTitle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) dialogTitle)            Sets the string that goes in the JFileChooser window's title bar. |
| void | [**setDialogType**](http://docs.google.com/javax/swing/JFileChooser.html#setDialogType(int))(int dialogType)            Sets the type of this dialog. |
| void | [**setDragEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#setDragEnabled(boolean))(boolean b)            Sets the dragEnabled property, which must be true to enable automatic drag handling (the first part of drag and drop) on this component. |
| void | [**setFileFilter**](http://docs.google.com/javax/swing/JFileChooser.html#setFileFilter(javax.swing.filechooser.FileFilter))([FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) filter)            Sets the current file filter. |
| void | [**setFileHidingEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#setFileHidingEnabled(boolean))(boolean b)            Sets file hiding on or off. |
| void | [**setFileSelectionMode**](http://docs.google.com/javax/swing/JFileChooser.html#setFileSelectionMode(int))(int mode)            Sets the JFileChooser to allow the user to just select files, just select directories, or select both files and directories. |
| void | [**setFileSystemView**](http://docs.google.com/javax/swing/JFileChooser.html#setFileSystemView(javax.swing.filechooser.FileSystemView))([FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)            Sets the file system view that the JFileChooser uses for accessing and creating file system resources, such as finding the floppy drive and getting a list of root drives. |
| void | [**setFileView**](http://docs.google.com/javax/swing/JFileChooser.html#setFileView(javax.swing.filechooser.FileView))([FileView](http://docs.google.com/javax/swing/filechooser/FileView.html) fileView)            Sets the file view to used to retrieve UI information, such as the icon that represents a file or the type description of a file. |
| void | [**setMultiSelectionEnabled**](http://docs.google.com/javax/swing/JFileChooser.html#setMultiSelectionEnabled(boolean))(boolean b)            Sets the file chooser to allow multiple file selections. |
| void | [**setSelectedFile**](http://docs.google.com/javax/swing/JFileChooser.html#setSelectedFile(java.io.File))([File](http://docs.google.com/java/io/File.html) file)            Sets the selected file. |
| void | [**setSelectedFiles**](http://docs.google.com/javax/swing/JFileChooser.html#setSelectedFiles(java.io.File%5B%5D))([File](http://docs.google.com/java/io/File.html)[] selectedFiles)            Sets the list of selected files if the file chooser is set to allow multiple selection. |
| protected  void | [**setup**](http://docs.google.com/javax/swing/JFileChooser.html#setup(javax.swing.filechooser.FileSystemView))([FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) view)            Performs common constructor initialization and setup. |
| int | [**showDialog**](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))([Component](http://docs.google.com/java/awt/Component.html) parent, [String](http://docs.google.com/java/lang/String.html) approveButtonText)            Pops a custom file chooser dialog with a custom approve button. |
| int | [**showOpenDialog**](http://docs.google.com/javax/swing/JFileChooser.html#showOpenDialog(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) parent)            Pops up an "Open File" file chooser dialog. |
| int | [**showSaveDialog**](http://docs.google.com/javax/swing/JFileChooser.html#showSaveDialog(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) parent)            Pops up a "Save File" file chooser dialog. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JFileChooser.html#updateUI())()            Resets the UI property to a value from the current look and feel. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

| **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()), [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** |
| --- |

### OPEN\_DIALOG

public static final int **OPEN\_DIALOG**

Type value indicating that the JFileChooser supports an "Open" file operation.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.OPEN_DIALOG)

### SAVE\_DIALOG

public static final int **SAVE\_DIALOG**

Type value indicating that the JFileChooser supports a "Save" file operation.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.SAVE_DIALOG)

### CUSTOM\_DIALOG

public static final int **CUSTOM\_DIALOG**

Type value indicating that the JFileChooser supports a developer-specified file operation.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.CUSTOM_DIALOG)

### CANCEL\_OPTION

public static final int **CANCEL\_OPTION**

Return value if cancel is chosen.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.CANCEL_OPTION)

### APPROVE\_OPTION

public static final int **APPROVE\_OPTION**

Return value if approve (yes, ok) is chosen.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.APPROVE_OPTION)

### ERROR\_OPTION

public static final int **ERROR\_OPTION**

Return value if an error occured.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.ERROR_OPTION)

### FILES\_ONLY

public static final int **FILES\_ONLY**

Instruction to display only files.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILES_ONLY)

### DIRECTORIES\_ONLY

public static final int **DIRECTORIES\_ONLY**

Instruction to display only directories.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.DIRECTORIES_ONLY)

### FILES\_AND\_DIRECTORIES

public static final int **FILES\_AND\_DIRECTORIES**

Instruction to display both files and directories.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILES_AND_DIRECTORIES)

### CANCEL\_SELECTION

public static final [String](http://docs.google.com/java/lang/String.html) **CANCEL\_SELECTION**

Instruction to cancel the current selection.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.CANCEL_SELECTION)

### APPROVE\_SELECTION

public static final [String](http://docs.google.com/java/lang/String.html) **APPROVE\_SELECTION**

Instruction to approve the current selection (same as pressing yes or ok).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.APPROVE_SELECTION)

### APPROVE\_BUTTON\_TEXT\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **APPROVE\_BUTTON\_TEXT\_CHANGED\_PROPERTY**

Identifies change in the text on the approve (yes, ok) button.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.APPROVE_BUTTON_TEXT_CHANGED_PROPERTY)

### APPROVE\_BUTTON\_TOOL\_TIP\_TEXT\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **APPROVE\_BUTTON\_TOOL\_TIP\_TEXT\_CHANGED\_PROPERTY**

Identifies change in the tooltip text for the approve (yes, ok) button.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.APPROVE_BUTTON_TOOL_TIP_TEXT_CHANGED_PROPERTY)

### APPROVE\_BUTTON\_MNEMONIC\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **APPROVE\_BUTTON\_MNEMONIC\_CHANGED\_PROPERTY**

Identifies change in the mnemonic for the approve (yes, ok) button.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.APPROVE_BUTTON_MNEMONIC_CHANGED_PROPERTY)

### CONTROL\_BUTTONS\_ARE\_SHOWN\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **CONTROL\_BUTTONS\_ARE\_SHOWN\_CHANGED\_PROPERTY**

Instruction to display the control buttons.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.CONTROL_BUTTONS_ARE_SHOWN_CHANGED_PROPERTY)

### DIRECTORY\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **DIRECTORY\_CHANGED\_PROPERTY**

Identifies user's directory change.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.DIRECTORY_CHANGED_PROPERTY)

### SELECTED\_FILE\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **SELECTED\_FILE\_CHANGED\_PROPERTY**

Identifies change in user's single-file selection.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.SELECTED_FILE_CHANGED_PROPERTY)

### SELECTED\_FILES\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **SELECTED\_FILES\_CHANGED\_PROPERTY**

Identifies change in user's multiple-file selection.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.SELECTED_FILES_CHANGED_PROPERTY)

### MULTI\_SELECTION\_ENABLED\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **MULTI\_SELECTION\_ENABLED\_CHANGED\_PROPERTY**

Enables multiple-file selections.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.MULTI_SELECTION_ENABLED_CHANGED_PROPERTY)

### FILE\_SYSTEM\_VIEW\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **FILE\_SYSTEM\_VIEW\_CHANGED\_PROPERTY**

Says that a different object is being used to find available drives on the system.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILE_SYSTEM_VIEW_CHANGED_PROPERTY)

### FILE\_VIEW\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **FILE\_VIEW\_CHANGED\_PROPERTY**

Says that a different object is being used to retrieve file information.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILE_VIEW_CHANGED_PROPERTY)

### FILE\_HIDING\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **FILE\_HIDING\_CHANGED\_PROPERTY**

Identifies a change in the display-hidden-files property.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILE_HIDING_CHANGED_PROPERTY)

### FILE\_FILTER\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **FILE\_FILTER\_CHANGED\_PROPERTY**

User changed the kind of files to display.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILE_FILTER_CHANGED_PROPERTY)

### FILE\_SELECTION\_MODE\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **FILE\_SELECTION\_MODE\_CHANGED\_PROPERTY**

Identifies a change in the kind of selection (single, multiple, etc.).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.FILE_SELECTION_MODE_CHANGED_PROPERTY)

### ACCESSORY\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **ACCESSORY\_CHANGED\_PROPERTY**

Says that a different accessory component is in use (for example, to preview files).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.ACCESSORY_CHANGED_PROPERTY)

### ACCEPT\_ALL\_FILE\_FILTER\_USED\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **ACCEPT\_ALL\_FILE\_FILTER\_USED\_CHANGED\_PROPERTY**

Identifies whether a the AcceptAllFileFilter is used or not.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.ACCEPT_ALL_FILE_FILTER_USED_CHANGED_PROPERTY)

### DIALOG\_TITLE\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **DIALOG\_TITLE\_CHANGED\_PROPERTY**

Identifies a change in the dialog title.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.DIALOG_TITLE_CHANGED_PROPERTY)

### DIALOG\_TYPE\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **DIALOG\_TYPE\_CHANGED\_PROPERTY**

Identifies a change in the type of files displayed (files only, directories only, or both files and directories).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.DIALOG_TYPE_CHANGED_PROPERTY)

### CHOOSABLE\_FILE\_FILTER\_CHANGED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **CHOOSABLE\_FILE\_FILTER\_CHANGED\_PROPERTY**

Identifies a change in the list of predefined file filters the user can choose from.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFileChooser.CHOOSABLE_FILE_FILTER_CHANGED_PROPERTY)

### accessibleContext

protected [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **accessibleContext**

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

### JFileChooser

public **JFileChooser**()

Constructs a JFileChooser pointing to the user's default directory. This default depends on the operating system. It is typically the "My Documents" folder on Windows, and the user's home directory on Unix.

### JFileChooser

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

Constructs a JFileChooser using the given path. Passing in a null string causes the file chooser to point to the user's default directory. This default depends on the operating system. It is typically the "My Documents" folder on Windows, and the user's home directory on Unix.

**Parameters:**currentDirectoryPath - a String giving the path to a file or directory

### JFileChooser

public **JFileChooser**([File](http://docs.google.com/java/io/File.html) currentDirectory)

Constructs a JFileChooser using the given File as the path. Passing in a null file causes the file chooser to point to the user's default directory. This default depends on the operating system. It is typically the "My Documents" folder on Windows, and the user's home directory on Unix.

**Parameters:**currentDirectory - a File object specifying the path to a file or directory

### JFileChooser

public **JFileChooser**([FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)

Constructs a JFileChooser using the given FileSystemView.

### JFileChooser

public **JFileChooser**([File](http://docs.google.com/java/io/File.html) currentDirectory,  
 [FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)

Constructs a JFileChooser using the given current directory and FileSystemView.

### JFileChooser

public **JFileChooser**([String](http://docs.google.com/java/lang/String.html) currentDirectoryPath,  
 [FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)

Constructs a JFileChooser using the given current directory path and FileSystemView.

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

### setup

protected void **setup**([FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) view)

Performs common constructor initialization and setup.

### setDragEnabled

public void **setDragEnabled**(boolean b)

Sets the dragEnabled property, which must be true to enable automatic drag handling (the first part of drag and drop) on this component. The transferHandler property needs to be set to a non-null value for the drag to do anything. The default value of the dragEnabled property is false.

When automatic drag handling is enabled, most look and feels begin a drag-and-drop operation whenever the user presses the mouse button over an item and then moves the mouse a few pixels. Setting this property to true can therefore have a subtle effect on how selections behave.

Some look and feels might not support automatic drag and drop; they will ignore this property. You can work around such look and feels by modifying the component to directly call the exportAsDrag method of a TransferHandler.

**Parameters:**b - the value to set the dragEnabled property to **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if b is true and GraphicsEnvironment.isHeadless() returns true**Since:** 1.4 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [getDragEnabled()](http://docs.google.com/javax/swing/JFileChooser.html#getDragEnabled()), [JComponent.setTransferHandler(javax.swing.TransferHandler)](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html)

### getDragEnabled

public boolean **getDragEnabled**()

Gets the value of the dragEnabled property.

**Returns:**the value of the dragEnabled property**Since:** 1.4 **See Also:**[setDragEnabled(boolean)](http://docs.google.com/javax/swing/JFileChooser.html#setDragEnabled(boolean))

### getSelectedFile

public [File](http://docs.google.com/java/io/File.html) **getSelectedFile**()

Returns the selected file. This can be set either by the programmer via setFile or by a user action, such as either typing the filename into the UI or selecting the file from a list in the UI.

**Returns:**the selected file**See Also:**[setSelectedFile(java.io.File)](http://docs.google.com/javax/swing/JFileChooser.html#setSelectedFile(java.io.File))

### setSelectedFile

public void **setSelectedFile**([File](http://docs.google.com/java/io/File.html) file)

Sets the selected file. If the file's parent directory is not the current directory, changes the current directory to be the file's parent directory.

**Parameters:**file - the selected file**See Also:**[getSelectedFile()](http://docs.google.com/javax/swing/JFileChooser.html#getSelectedFile())

### getSelectedFiles

public [File](http://docs.google.com/java/io/File.html)[] **getSelectedFiles**()

Returns a list of selected files if the file chooser is set to allow multiple selection.

### setSelectedFiles

public void **setSelectedFiles**([File](http://docs.google.com/java/io/File.html)[] selectedFiles)

Sets the list of selected files if the file chooser is set to allow multiple selection.

### getCurrentDirectory

public [File](http://docs.google.com/java/io/File.html) **getCurrentDirectory**()

Returns the current directory.

**Returns:**the current directory**See Also:**[setCurrentDirectory(java.io.File)](http://docs.google.com/javax/swing/JFileChooser.html#setCurrentDirectory(java.io.File))

### setCurrentDirectory

public void **setCurrentDirectory**([File](http://docs.google.com/java/io/File.html) dir)

Sets the current directory. Passing in null sets the file chooser to point to the user's default directory. This default depends on the operating system. It is typically the "My Documents" folder on Windows, and the user's home directory on Unix. If the file passed in as currentDirectory is not a directory, the parent of the file will be used as the currentDirectory. If the parent is not traversable, then it will walk up the parent tree until it finds a traversable directory, or hits the root of the file system.

**Parameters:**dir - the current directory to point to**See Also:**[getCurrentDirectory()](http://docs.google.com/javax/swing/JFileChooser.html#getCurrentDirectory())

### changeToParentDirectory

public void **changeToParentDirectory**()

Changes the directory to be set to the parent of the current directory.

**See Also:**[getCurrentDirectory()](http://docs.google.com/javax/swing/JFileChooser.html#getCurrentDirectory())

### rescanCurrentDirectory

public void **rescanCurrentDirectory**()

Tells the UI to rescan its files list from the current directory.

### ensureFileIsVisible

public void **ensureFileIsVisible**([File](http://docs.google.com/java/io/File.html) f)

Makes sure that the specified file is viewable, and not hidden.

**Parameters:**f - a File object

### showOpenDialog

public int **showOpenDialog**([Component](http://docs.google.com/java/awt/Component.html) parent)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Pops up an "Open File" file chooser dialog. Note that the text that appears in the approve button is determined by the L&F.

**Parameters:**parent - the parent component of the dialog, can be null; see showDialog for details **Returns:**the return state of the file chooser on popdown:

* JFileChooser.CANCEL\_OPTION
* JFileChooser.APPROVE\_OPTION
* JFileChooser.ERROR\_OPTION if an error occurs or the dialog is dismissed

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [showDialog(java.awt.Component, java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))

### showSaveDialog

public int **showSaveDialog**([Component](http://docs.google.com/java/awt/Component.html) parent)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Pops up a "Save File" file chooser dialog. Note that the text that appears in the approve button is determined by the L&F.

**Parameters:**parent - the parent component of the dialog, can be null; see showDialog for details **Returns:**the return state of the file chooser on popdown:

* JFileChooser.CANCEL\_OPTION
* JFileChooser.APPROVE\_OPTION
* JFileChooser.ERROR\_OPTION if an error occurs or the dialog is dismissed

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [showDialog(java.awt.Component, java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))

### showDialog

public int **showDialog**([Component](http://docs.google.com/java/awt/Component.html) parent,  
 [String](http://docs.google.com/java/lang/String.html) approveButtonText)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Pops a custom file chooser dialog with a custom approve button. For example, the following code pops up a file chooser with a "Run Application" button (instead of the normal "Save" or "Open" button):

filechooser.showDialog(parentFrame, "Run Application");

Alternatively, the following code does the same thing:

JFileChooser chooser = new JFileChooser(null);  
 chooser.setApproveButtonText("Run Application");  
 chooser.showDialog(parentFrame, null);

The parent argument determines two things: the frame on which the open dialog depends and the component whose position the look and feel should consider when placing the dialog. If the parent is a Frame object (such as a JFrame) then the dialog depends on the frame and the look and feel positions the dialog relative to the frame (for example, centered over the frame). If the parent is a component, then the dialog depends on the frame containing the component, and is positioned relative to the component (for example, centered over the component). If the parent is null, then the dialog depends on no visible window, and it's placed in a look-and-feel-dependent position such as the center of the screen.

**Parameters:**parent - the parent component of the dialog; can be nullapproveButtonText - the text of the ApproveButton **Returns:**the return state of the file chooser on popdown:

* JFileChooser.CANCEL\_OPTION
* JFileChooser.APPROVE\_OPTION
* JFileCHooser.ERROR\_OPTION if an error occurs or the dialog is dismissed

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### createDialog

protected [JDialog](http://docs.google.com/javax/swing/JDialog.html) **createDialog**([Component](http://docs.google.com/java/awt/Component.html) parent)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates and returns a new JDialog wrapping this centered on the parent in the parent's frame. This method can be overriden to further manipulate the dialog, to disable resizing, set the location, etc. Example:

class MyFileChooser extends JFileChooser {  
 protected JDialog createDialog(Component parent) throws HeadlessException {  
 JDialog dialog = super.createDialog(parent);  
 dialog.setLocation(300, 200);  
 dialog.setResizable(false);  
 return dialog;  
 }  
 }

**Parameters:**parent - the parent component of the dialog; can be null **Returns:**a new JDialog containing this instance **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**Since:** 1.4 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### getControlButtonsAreShown

public boolean **getControlButtonsAreShown**()

Returns the value of the controlButtonsAreShown property.

**Returns:**the value of the controlButtonsAreShown property**Since:** 1.3 **See Also:**[setControlButtonsAreShown(boolean)](http://docs.google.com/javax/swing/JFileChooser.html#setControlButtonsAreShown(boolean))

### setControlButtonsAreShown

public void **setControlButtonsAreShown**(boolean b)

Sets the property that indicates whether the *approve* and *cancel* buttons are shown in the file chooser. This property is true by default. Look and feels that always show these buttons will ignore the value of this property. This method fires a property-changed event, using the string value of CONTROL\_BUTTONS\_ARE\_SHOWN\_CHANGED\_PROPERTY as the name of the property.

**Parameters:**b - false if control buttons should not be shown; otherwise, true**Since:** 1.3 **See Also:**[getControlButtonsAreShown()](http://docs.google.com/javax/swing/JFileChooser.html#getControlButtonsAreShown()), [CONTROL\_BUTTONS\_ARE\_SHOWN\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/JFileChooser.html#CONTROL_BUTTONS_ARE_SHOWN_CHANGED_PROPERTY)

### getDialogType

public int **getDialogType**()

Returns the type of this dialog. The default is JFileChooser.OPEN\_DIALOG.

**Returns:**the type of dialog to be displayed:

* JFileChooser.OPEN\_DIALOG
* JFileChooser.SAVE\_DIALOG
* JFileChooser.CUSTOM\_DIALOG

**See Also:**[setDialogType(int)](http://docs.google.com/javax/swing/JFileChooser.html#setDialogType(int))

### setDialogType

public void **setDialogType**(int dialogType)

Sets the type of this dialog. Use OPEN\_DIALOG when you want to bring up a file chooser that the user can use to open a file. Likewise, use SAVE\_DIALOG for letting the user choose a file for saving. Use CUSTOM\_DIALOG when you want to use the file chooser in a context other than "Open" or "Save". For instance, you might want to bring up a file chooser that allows the user to choose a file to execute. Note that you normally would not need to set the JFileChooser to use CUSTOM\_DIALOG since a call to setApproveButtonText does this for you. The default dialog type is JFileChooser.OPEN\_DIALOG.

**Parameters:**dialogType - the type of dialog to be displayed:

* JFileChooser.OPEN\_DIALOG
* JFileChooser.SAVE\_DIALOG
* JFileChooser.CUSTOM\_DIALOG

**Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dialogType is not legal**See Also:**[getDialogType()](http://docs.google.com/javax/swing/JFileChooser.html#getDialogType()), [setApproveButtonText(java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonText(java.lang.String))

### setDialogTitle

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

Sets the string that goes in the JFileChooser window's title bar.

**Parameters:**dialogTitle - the new String for the title bar**See Also:**[getDialogTitle()](http://docs.google.com/javax/swing/JFileChooser.html#getDialogTitle())

### getDialogTitle

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

Gets the string that goes in the JFileChooser's titlebar.

**See Also:**[setDialogTitle(java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#setDialogTitle(java.lang.String))

### setApproveButtonToolTipText

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

Sets the tooltip text used in the ApproveButton. If null, the UI object will determine the button's text.

**Parameters:**toolTipText - the tooltip text for the approve button**See Also:**[setApproveButtonText(java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonText(java.lang.String)), [setDialogType(int)](http://docs.google.com/javax/swing/JFileChooser.html#setDialogType(int)), [showDialog(java.awt.Component, java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))

### getApproveButtonToolTipText

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

Returns the tooltip text used in the ApproveButton. If null, the UI object will determine the button's text.

**Returns:**the tooltip text used for the approve button**See Also:**[setApproveButtonText(java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonText(java.lang.String)), [setDialogType(int)](http://docs.google.com/javax/swing/JFileChooser.html#setDialogType(int)), [showDialog(java.awt.Component, java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))

### getApproveButtonMnemonic

public int **getApproveButtonMnemonic**()

Returns the approve button's mnemonic.

**Returns:**an integer value for the mnemonic key**See Also:**[setApproveButtonMnemonic(int)](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonMnemonic(int))

### setApproveButtonMnemonic

public void **setApproveButtonMnemonic**(int mnemonic)

Sets the approve button's mnemonic using a numeric keycode.

**Parameters:**mnemonic - an integer value for the mnemonic key**See Also:**[getApproveButtonMnemonic()](http://docs.google.com/javax/swing/JFileChooser.html#getApproveButtonMnemonic())

### setApproveButtonMnemonic

public void **setApproveButtonMnemonic**(char mnemonic)

Sets the approve button's mnemonic using a character.

**Parameters:**mnemonic - a character value for the mnemonic key**See Also:**[getApproveButtonMnemonic()](http://docs.google.com/javax/swing/JFileChooser.html#getApproveButtonMnemonic())

### setApproveButtonText

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

Sets the text used in the ApproveButton in the FileChooserUI.

**Parameters:**approveButtonText - the text used in the ApproveButton**See Also:**[getApproveButtonText()](http://docs.google.com/javax/swing/JFileChooser.html#getApproveButtonText()), [setDialogType(int)](http://docs.google.com/javax/swing/JFileChooser.html#setDialogType(int)), [showDialog(java.awt.Component, java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))

### getApproveButtonText

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

Returns the text used in the ApproveButton in the FileChooserUI. If null, the UI object will determine the button's text. Typically, this would be "Open" or "Save".

**Returns:**the text used in the ApproveButton**See Also:**[setApproveButtonText(java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#setApproveButtonText(java.lang.String)), [setDialogType(int)](http://docs.google.com/javax/swing/JFileChooser.html#setDialogType(int)), [showDialog(java.awt.Component, java.lang.String)](http://docs.google.com/javax/swing/JFileChooser.html#showDialog(java.awt.Component,%20java.lang.String))

### getChoosableFileFilters

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

Gets the list of user choosable file filters.

**Returns:**a FileFilter array containing all the choosable file filters**See Also:**[addChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#addChoosableFileFilter(javax.swing.filechooser.FileFilter)), [removeChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#removeChoosableFileFilter(javax.swing.filechooser.FileFilter)), [resetChoosableFileFilters()](http://docs.google.com/javax/swing/JFileChooser.html#resetChoosableFileFilters())

### addChoosableFileFilter

public void **addChoosableFileFilter**([FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) filter)

Adds a filter to the list of user choosable file filters. For information on setting the file selection mode, see [setFileSelectionMode](http://docs.google.com/javax/swing/JFileChooser.html#setFileSelectionMode(int)).

**Parameters:**filter - the FileFilter to add to the choosable file filter list**See Also:**[getChoosableFileFilters()](http://docs.google.com/javax/swing/JFileChooser.html#getChoosableFileFilters()), [removeChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#removeChoosableFileFilter(javax.swing.filechooser.FileFilter)), [resetChoosableFileFilters()](http://docs.google.com/javax/swing/JFileChooser.html#resetChoosableFileFilters()), [setFileSelectionMode(int)](http://docs.google.com/javax/swing/JFileChooser.html#setFileSelectionMode(int))

### removeChoosableFileFilter

public boolean **removeChoosableFileFilter**([FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) f)

Removes a filter from the list of user choosable file filters. Returns true if the file filter was removed.

**See Also:**[addChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#addChoosableFileFilter(javax.swing.filechooser.FileFilter)), [getChoosableFileFilters()](http://docs.google.com/javax/swing/JFileChooser.html#getChoosableFileFilters()), [resetChoosableFileFilters()](http://docs.google.com/javax/swing/JFileChooser.html#resetChoosableFileFilters())

### resetChoosableFileFilters

public void **resetChoosableFileFilters**()

Resets the choosable file filter list to its starting state. Normally, this removes all added file filters while leaving the AcceptAll file filter.

**See Also:**[addChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#addChoosableFileFilter(javax.swing.filechooser.FileFilter)), [getChoosableFileFilters()](http://docs.google.com/javax/swing/JFileChooser.html#getChoosableFileFilters()), [removeChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#removeChoosableFileFilter(javax.swing.filechooser.FileFilter))

### getAcceptAllFileFilter

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

Returns the AcceptAll file filter. For example, on Microsoft Windows this would be All Files (\*.\*).

### isAcceptAllFileFilterUsed

public boolean **isAcceptAllFileFilterUsed**()

Returns whether the AcceptAll FileFilter is used.

**Returns:**true if the AcceptAll FileFilter is used**Since:** 1.3 **See Also:**[setAcceptAllFileFilterUsed(boolean)](http://docs.google.com/javax/swing/JFileChooser.html#setAcceptAllFileFilterUsed(boolean))

### setAcceptAllFileFilterUsed

public void **setAcceptAllFileFilterUsed**(boolean b)

Determines whether the AcceptAll FileFilter is used as an available choice in the choosable filter list. If false, the AcceptAll file filter is removed from the list of available file filters. If true, the AcceptAll file filter will become the the actively used file filter.

**Since:** 1.3 **See Also:**[isAcceptAllFileFilterUsed()](http://docs.google.com/javax/swing/JFileChooser.html#isAcceptAllFileFilterUsed()), [getAcceptAllFileFilter()](http://docs.google.com/javax/swing/JFileChooser.html#getAcceptAllFileFilter()), [setFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#setFileFilter(javax.swing.filechooser.FileFilter))

### getAccessory

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

Returns the accessory component.

**Returns:**this JFileChooser's accessory component, or null**See Also:**[setAccessory(javax.swing.JComponent)](http://docs.google.com/javax/swing/JFileChooser.html#setAccessory(javax.swing.JComponent))

### setAccessory

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

Sets the accessory component. An accessory is often used to show a preview image of the selected file; however, it can be used for anything that the programmer wishes, such as extra custom file chooser controls.

Note: if there was a previous accessory, you should unregister any listeners that the accessory might have registered with the file chooser.

### setFileSelectionMode

public void **setFileSelectionMode**(int mode)

Sets the JFileChooser to allow the user to just select files, just select directories, or select both files and directories. The default is JFilesChooser.FILES\_ONLY.

**Parameters:**mode - the type of files to be displayed:

* JFileChooser.FILES\_ONLY
* JFileChooser.DIRECTORIES\_ONLY
* JFileChooser.FILES\_AND\_DIRECTORIES

**Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if mode is an illegal file selection mode**See Also:**[getFileSelectionMode()](http://docs.google.com/javax/swing/JFileChooser.html#getFileSelectionMode())

### getFileSelectionMode

public int **getFileSelectionMode**()

Returns the current file-selection mode. The default is JFilesChooser.FILES\_ONLY.

**Returns:**the type of files to be displayed, one of the following:

* JFileChooser.FILES\_ONLY
* JFileChooser.DIRECTORIES\_ONLY
* JFileChooser.FILES\_AND\_DIRECTORIES

**See Also:**[setFileSelectionMode(int)](http://docs.google.com/javax/swing/JFileChooser.html#setFileSelectionMode(int))

### isFileSelectionEnabled

public boolean **isFileSelectionEnabled**()

Convenience call that determines if files are selectable based on the current file selection mode.

**See Also:**[setFileSelectionMode(int)](http://docs.google.com/javax/swing/JFileChooser.html#setFileSelectionMode(int)), [getFileSelectionMode()](http://docs.google.com/javax/swing/JFileChooser.html#getFileSelectionMode())

### isDirectorySelectionEnabled

public boolean **isDirectorySelectionEnabled**()

Convenience call that determines if directories are selectable based on the current file selection mode.

**See Also:**[setFileSelectionMode(int)](http://docs.google.com/javax/swing/JFileChooser.html#setFileSelectionMode(int)), [getFileSelectionMode()](http://docs.google.com/javax/swing/JFileChooser.html#getFileSelectionMode())

### setMultiSelectionEnabled

public void **setMultiSelectionEnabled**(boolean b)

Sets the file chooser to allow multiple file selections.

**Parameters:**b - true if multiple files may be selected**See Also:**[isMultiSelectionEnabled()](http://docs.google.com/javax/swing/JFileChooser.html#isMultiSelectionEnabled())

### isMultiSelectionEnabled

public boolean **isMultiSelectionEnabled**()

Returns true if multiple files can be selected.

**Returns:**true if multiple files can be selected**See Also:**[setMultiSelectionEnabled(boolean)](http://docs.google.com/javax/swing/JFileChooser.html#setMultiSelectionEnabled(boolean))

### isFileHidingEnabled

public boolean **isFileHidingEnabled**()

Returns true if hidden files are not shown in the file chooser; otherwise, returns false.

**Returns:**the status of the file hiding property**See Also:**[setFileHidingEnabled(boolean)](http://docs.google.com/javax/swing/JFileChooser.html#setFileHidingEnabled(boolean))

### setFileHidingEnabled

public void **setFileHidingEnabled**(boolean b)

Sets file hiding on or off. If true, hidden files are not shown in the file chooser. The job of determining which files are shown is done by the FileView.

**Parameters:**b - the boolean value that determines whether file hiding is turned on**See Also:**[isFileHidingEnabled()](http://docs.google.com/javax/swing/JFileChooser.html#isFileHidingEnabled())

### setFileFilter

public void **setFileFilter**([FileFilter](http://docs.google.com/javax/swing/filechooser/FileFilter.html) filter)

Sets the current file filter. The file filter is used by the file chooser to filter out files from the user's view.

**Parameters:**filter - the new current file filter to use**See Also:**[getFileFilter()](http://docs.google.com/javax/swing/JFileChooser.html#getFileFilter())

### getFileFilter

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

Returns the currently selected file filter.

**Returns:**the current file filter**See Also:**[setFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#setFileFilter(javax.swing.filechooser.FileFilter)), [addChoosableFileFilter(javax.swing.filechooser.FileFilter)](http://docs.google.com/javax/swing/JFileChooser.html#addChoosableFileFilter(javax.swing.filechooser.FileFilter))

### setFileView

public void **setFileView**([FileView](http://docs.google.com/javax/swing/filechooser/FileView.html) fileView)

Sets the file view to used to retrieve UI information, such as the icon that represents a file or the type description of a file.

**See Also:**[getFileView()](http://docs.google.com/javax/swing/JFileChooser.html#getFileView())

### getFileView

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

Returns the current file view.

**See Also:**[setFileView(javax.swing.filechooser.FileView)](http://docs.google.com/javax/swing/JFileChooser.html#setFileView(javax.swing.filechooser.FileView))

### getName

public [String](http://docs.google.com/java/lang/String.html) **getName**([File](http://docs.google.com/java/io/File.html) f)

Returns the filename.

**Parameters:**f - the File **Returns:**the String containing the filename for f**See Also:**[FileView.getName(java.io.File)](http://docs.google.com/javax/swing/filechooser/FileView.html#getName(java.io.File))

### getDescription

public [String](http://docs.google.com/java/lang/String.html) **getDescription**([File](http://docs.google.com/java/io/File.html) f)

Returns the file description.

**Parameters:**f - the File **Returns:**the String containing the file description for f**See Also:**[FileView.getDescription(java.io.File)](http://docs.google.com/javax/swing/filechooser/FileView.html#getDescription(java.io.File))

### getTypeDescription

public [String](http://docs.google.com/java/lang/String.html) **getTypeDescription**([File](http://docs.google.com/java/io/File.html) f)

Returns the file type.

**Parameters:**f - the File **Returns:**the String containing the file type description for f**See Also:**[FileView.getTypeDescription(java.io.File)](http://docs.google.com/javax/swing/filechooser/FileView.html#getTypeDescription(java.io.File))

### getIcon

public [Icon](http://docs.google.com/javax/swing/Icon.html) **getIcon**([File](http://docs.google.com/java/io/File.html) f)

Returns the icon for this file or type of file, depending on the system.

**Parameters:**f - the File **Returns:**the Icon for this file, or type of file**See Also:**[FileView.getIcon(java.io.File)](http://docs.google.com/javax/swing/filechooser/FileView.html#getIcon(java.io.File))

### isTraversable

public boolean **isTraversable**([File](http://docs.google.com/java/io/File.html) f)

Returns true if the file (directory) can be visited. Returns false if the directory cannot be traversed.

**Parameters:**f - the File **Returns:**true if the file/directory can be traversed, otherwise false**See Also:**[FileView.isTraversable(java.io.File)](http://docs.google.com/javax/swing/filechooser/FileView.html#isTraversable(java.io.File))

### accept

public boolean **accept**([File](http://docs.google.com/java/io/File.html) f)

Returns true if the file should be displayed.

**Parameters:**f - the File **Returns:**true if the file should be displayed, otherwise false**See Also:**[FileFilter.accept(java.io.File)](http://docs.google.com/javax/swing/filechooser/FileFilter.html#accept(java.io.File))

### setFileSystemView

public void **setFileSystemView**([FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) fsv)

Sets the file system view that the JFileChooser uses for accessing and creating file system resources, such as finding the floppy drive and getting a list of root drives.

**Parameters:**fsv - the new FileSystemView**See Also:**[FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html)

### getFileSystemView

public [FileSystemView](http://docs.google.com/javax/swing/filechooser/FileSystemView.html) **getFileSystemView**()

Returns the file system view.

**Returns:**the FileSystemView object**See Also:**[setFileSystemView(javax.swing.filechooser.FileSystemView)](http://docs.google.com/javax/swing/JFileChooser.html#setFileSystemView(javax.swing.filechooser.FileSystemView))

### approveSelection

public void **approveSelection**()

Called by the UI when the user hits the Approve button (labeled "Open" or "Save", by default). This can also be called by the programmer. This method causes an action event to fire with the command string equal to APPROVE\_SELECTION.

**See Also:**[APPROVE\_SELECTION](http://docs.google.com/javax/swing/JFileChooser.html#APPROVE_SELECTION)

### cancelSelection

public void **cancelSelection**()

Called by the UI when the user chooses the Cancel button. This can also be called by the programmer. This method causes an action event to fire with the command string equal to CANCEL\_SELECTION.

**See Also:**[CANCEL\_SELECTION](http://docs.google.com/javax/swing/JFileChooser.html#CANCEL_SELECTION)

### addActionListener

public void **addActionListener**([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)

Adds an ActionListener to the file chooser.

**Parameters:**l - the listener to be added**See Also:**[approveSelection()](http://docs.google.com/javax/swing/JFileChooser.html#approveSelection()), [cancelSelection()](http://docs.google.com/javax/swing/JFileChooser.html#cancelSelection())

### removeActionListener

public void **removeActionListener**([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)

Removes an ActionListener from the file chooser.

**Parameters:**l - the listener to be removed**See Also:**[addActionListener(java.awt.event.ActionListener)](http://docs.google.com/javax/swing/JFileChooser.html#addActionListener(java.awt.event.ActionListener))

### getActionListeners

public [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] **getActionListeners**()

Returns an array of all the action listeners registered on this file chooser.

**Returns:**all of this file chooser's ActionListeners or an empty array if no action listeners are currently registered**Since:** 1.4 **See Also:**[addActionListener(java.awt.event.ActionListener)](http://docs.google.com/javax/swing/JFileChooser.html#addActionListener(java.awt.event.ActionListener)), [removeActionListener(java.awt.event.ActionListener)](http://docs.google.com/javax/swing/JFileChooser.html#removeActionListener(java.awt.event.ActionListener))

### fireActionPerformed

protected void **fireActionPerformed**([String](http://docs.google.com/java/lang/String.html) command)

Notifies all listeners that have registered interest for notification on this event type. The event instance is lazily created using the command parameter.

**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### updateUI

public void **updateUI**()

Resets the UI property to a value from the current look and feel.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

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

Returns a string that specifies the name of the L&F class that renders this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "FileChooserUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### getUI

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

Gets the UI object which implements the L&F for this component.

**Returns:**the FileChooserUI object that implements the FileChooserUI L&F

### paramString

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

Returns a string representation of this JFileChooser. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

**Overrides:**[paramString](http://docs.google.com/javax/swing/JComponent.html#paramString()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a string representation of this JFileChooser

### getAccessibleContext

public [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **getAccessibleContext**()

Gets the AccessibleContext associated with this JFileChooser. For file choosers, the AccessibleContext takes the form of an AccessibleJFileChooser. A new AccessibleJFileChooser instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/javax/swing/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJFileChooser that serves as the AccessibleContext of this JFileChooser

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JFileChooser.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/JEditorPane.JEditorPaneAccessibleHypertextSupport.HTMLLink.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JFileChooser.AccessibleJFileChooser.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JFileChooser.html)    [**NO FRAMES**](http://docs.google.com/JFileChooser.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#37m2jsg) | [METHOD](#4k668n3) |

[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).