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

## **javax.swing**

Interface DesktopManager

**All Known Implementing Classes:** [DefaultDesktopManager](http://docs.google.com/javax/swing/DefaultDesktopManager.html)

public interface **DesktopManager**

DesktopManager objects are owned by a JDesktopPane object. They are responsible for implementing L&F specific behaviors for the JDesktopPane. JInternalFrame implementations should delegate specific behaviors to the DesktopManager. For instance, if a JInternalFrame was asked to iconify, it should try:

getDesktopPane().getDesktopManager().iconifyFrame(frame);

This delegation allows each L&F to provide custom behaviors for desktop-specific actions. (For example, how and where the internal frame's icon would appear.)

This class provides a policy for the various JInternalFrame methods, it is not meant to be called directly rather the various JInternalFrame methods will call into the DesktopManager.

**See Also:**[JDesktopPane](http://docs.google.com/javax/swing/JDesktopPane.html), [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html), [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html)

| **Method Summary** | |
| --- | --- |
| void | [**activateFrame**](http://docs.google.com/javax/swing/DesktopManager.html#activateFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, indicate that this frame has focus. |
| void | [**beginDraggingFrame**](http://docs.google.com/javax/swing/DesktopManager.html#beginDraggingFrame(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f)            This method is normally called when the user has indicated that they will begin dragging a component around. |
| void | [**beginResizingFrame**](http://docs.google.com/javax/swing/DesktopManager.html#beginResizingFrame(javax.swing.JComponent,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f, int direction)            This methods is normally called when the user has indicated that they will begin resizing the frame. |
| void | [**closeFrame**](http://docs.google.com/javax/swing/DesktopManager.html#closeFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, this call should remove the frame from it's parent. |
| void | [**deactivateFrame**](http://docs.google.com/javax/swing/DesktopManager.html#deactivateFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, indicate that this frame has lost focus. |
| void | [**deiconifyFrame**](http://docs.google.com/javax/swing/DesktopManager.html#deiconifyFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, remove any iconic representation that is present and restore the frame to it's original size and location. |
| void | [**dragFrame**](http://docs.google.com/javax/swing/DesktopManager.html#dragFrame(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f, int newX, int newY)            The user has moved the frame. |
| void | [**endDraggingFrame**](http://docs.google.com/javax/swing/DesktopManager.html#endDraggingFrame(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f)            This method signals the end of the dragging session. |
| void | [**endResizingFrame**](http://docs.google.com/javax/swing/DesktopManager.html#endResizingFrame(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f)            This method signals the end of the resize session. |
| void | [**iconifyFrame**](http://docs.google.com/javax/swing/DesktopManager.html#iconifyFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, remove this frame from it's parent and add an iconic representation. |
| void | [**maximizeFrame**](http://docs.google.com/javax/swing/DesktopManager.html#maximizeFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, the frame should be resized to match it's parents bounds. |
| void | [**minimizeFrame**](http://docs.google.com/javax/swing/DesktopManager.html#minimizeFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            Generally, this indicates that the frame should be restored to it's size and position prior to a maximizeFrame() call. |
| void | [**openFrame**](http://docs.google.com/javax/swing/DesktopManager.html#openFrame(javax.swing.JInternalFrame))([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)            If possible, display this frame in an appropriate location. |
| void | [**resizeFrame**](http://docs.google.com/javax/swing/DesktopManager.html#resizeFrame(javax.swing.JComponent,%20int,%20int,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f, int newX, int newY, int newWidth, int newHeight)            The user has resized the component. |
| void | [**setBoundsForFrame**](http://docs.google.com/javax/swing/DesktopManager.html#setBoundsForFrame(javax.swing.JComponent,%20int,%20int,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) f, int newX, int newY, int newWidth, int newHeight)            This is a primitive reshape method. |

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

### openFrame

void **openFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

If possible, display this frame in an appropriate location. Normally, this is not called, as the creator of the JInternalFrame will add the frame to the appropriate parent.

### closeFrame

void **closeFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, this call should remove the frame from it's parent.

### maximizeFrame

void **maximizeFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, the frame should be resized to match it's parents bounds.

### minimizeFrame

void **minimizeFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, this indicates that the frame should be restored to it's size and position prior to a maximizeFrame() call.

### iconifyFrame

void **iconifyFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, remove this frame from it's parent and add an iconic representation.

### deiconifyFrame

void **deiconifyFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, remove any iconic representation that is present and restore the frame to it's original size and location.

### activateFrame

void **activateFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, indicate that this frame has focus. This is usually called after the JInternalFrame's IS\_SELECTED\_PROPERTY has been set to true.

### deactivateFrame

void **deactivateFrame**([JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) f)

Generally, indicate that this frame has lost focus. This is usually called after the JInternalFrame's IS\_SELECTED\_PROPERTY has been set to false.

### beginDraggingFrame

void **beginDraggingFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f)

This method is normally called when the user has indicated that they will begin dragging a component around. This method should be called prior to any dragFrame() calls to allow the DesktopManager to prepare any necessary state. Normally **f** will be a JInternalFrame.

### dragFrame

void **dragFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f,  
 int newX,  
 int newY)

The user has moved the frame. Calls to this method will be preceded by calls to beginDraggingFrame(). Normally **f** will be a JInternalFrame.

### endDraggingFrame

void **endDraggingFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f)

This method signals the end of the dragging session. Any state maintained by the DesktopManager can be removed here. Normally **f** will be a JInternalFrame.

### beginResizingFrame

void **beginResizingFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f,  
 int direction)

This methods is normally called when the user has indicated that they will begin resizing the frame. This method should be called prior to any resizeFrame() calls to allow the DesktopManager to prepare any necessary state. Normally **f** will be a JInternalFrame.

### resizeFrame

void **resizeFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f,  
 int newX,  
 int newY,  
 int newWidth,  
 int newHeight)

The user has resized the component. Calls to this method will be preceded by calls to beginResizingFrame(). Normally **f** will be a JInternalFrame.

### endResizingFrame

void **endResizingFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f)

This method signals the end of the resize session. Any state maintained by the DesktopManager can be removed here. Normally **f** will be a JInternalFrame.

### setBoundsForFrame

void **setBoundsForFrame**([JComponent](http://docs.google.com/javax/swing/JComponent.html) f,  
 int newX,  
 int newY,  
 int newWidth,  
 int newHeight)

This is a primitive reshape method.

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

[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).