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

## **javax.imageio**

Interface IIOParamController

public interface **IIOParamController**

An interface to be implemented by objects that can determine the settings of an IIOParam object, either by putting up a GUI to obtain values from a user, or by other means. This interface merely specifies a generic activate method that invokes the controller, without regard for how the controller obtains values (*i.e.*, whether the controller puts up a GUI or merely computes a set of values is irrelevant to this interface).

Within the activate method, a controller obtains initial values by querying the IIOParam object's get methods, modifies values by whatever means, then invokes the IIOParam object's set methods to modify the appropriate settings. Normally, these set methods will be invoked all at once at a final commit in order that a cancel operation not disturb existing values. In general, applications may expect that when the activate method returns true, the IIOParam object is ready for use in a read or write operation.

Vendors may choose to provide GUIs for the IIOParam subclasses they define for a particular plug-in. These can be set up as default controllers in the corresponding IIOParam subclasses.

Applications may override any default GUIs and provide their own controllers embedded in their own framework. All that is required is that theactivate method behave modally (not returning until either cancelled or committed), though it need not put up an explicitly modal dialog. Such a non-modal GUI component would be coded roughly as follows:

class MyGUI extends SomeComponent implements IIOParamController {  
  
 public MyGUI() {  
 // ...  
 setEnabled(false);  
 }  
  
 public boolean activate(IIOParam param) {  
 // disable other components if desired  
 setEnabled(true);  
 // go to sleep until either cancelled or committed  
 boolean ret = false;  
 if (!cancelled) {  
 // set values on param  
 ret = true;  
 }  
 setEnabled(false);  
 // enable any components disabled above  
 return ret;  
 }

Alternatively, an algorithmic process such as a database lookup or the parsing of a command line could be used as a controller, in which case the activate method would simply look up or compute the settings, call the IIOParam.setXXX methods, and return true.

**See Also:**[IIOParam.setController(javax.imageio.IIOParamController)](http://docs.google.com/javax/imageio/IIOParam.html#setController(javax.imageio.IIOParamController)), [IIOParam.getController()](http://docs.google.com/javax/imageio/IIOParam.html#getController()), [IIOParam.getDefaultController()](http://docs.google.com/javax/imageio/IIOParam.html#getDefaultController()), [IIOParam.hasController()](http://docs.google.com/javax/imageio/IIOParam.html#hasController()), [IIOParam.activateController()](http://docs.google.com/javax/imageio/IIOParam.html#activateController())

| **Method Summary** | |
| --- | --- |
| boolean | [**activate**](http://docs.google.com/javax/imageio/IIOParamController.html#activate(javax.imageio.IIOParam))([IIOParam](http://docs.google.com/javax/imageio/IIOParam.html) param)            Activates the controller. |

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

### activate

boolean **activate**([IIOParam](http://docs.google.com/javax/imageio/IIOParam.html) param)

Activates the controller. If true is returned, all settings in the IIOParam object should be ready for use in a read or write operation. If false is returned, no settings in the IIOParam object will be disturbed (*i.e.*, the user canceled the operation).

**Parameters:**param - the IIOParam object to be modified. **Returns:**true if the IIOParam has been modified, false otherwise. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if param is null or is not an instance of the correct class.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IIOParamController.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/imageio/IIOParam.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/ImageIO.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/IIOParamController.html)    [**NO FRAMES**](http://docs.google.com/IIOParamController.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).