

# Common Application Framework for Engineering Analysis (CAFEAN) Preprocessor Plugin API

## Appendix A Source Code Documentation

August 2005



---

## Package

# com.cafean.client.analysis

Provides the foundation classes for the ModelEditor.

Base classes are provided for representing [models](#), [components](#), [lists](#) of components and [connections](#) between components.

## com.cafean.client.analysis

### Class AbstractBeanComponent

```

java.lang.Object
|
+-com.cafean.client.analysis.GenericObject
|
|+-com.cafean.client.analysis.AbstractComponent
|
|+-com.cafean.client.analysis.AbstractBeanComponent

```

public abstract class **AbstractBeanComponent**  
 extends [AbstractComponent](#)

The base class for ModelEditor Components that are full fledged beans.

Fields inherited from class [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

### Constructor Summary

|        |  |
|--------|--|
| public | AbstractBeanComponent()<br>Create an AbstractBeanComponent object that does not belong to any model and has a display number of 0                                      |
| public | AbstractBeanComponent(AbstractModel model,int componentNumber)<br>Creates an AbstractBeanComponent object, adds it to model and sets the given component number on it. |

### Method Summary

|                                   |  |
|-----------------------------------|--|
| <a href="#">AbstractComponent</a> | copy(AbstractModel sm)<br>the following copy method produces a deep clone of a composite base so that it can be put in a copy model clipboard.   |
| void                              | copyFrom(GenericObject obj)  |
| boolean                           | dumpBlockParams(java.io.PrintWriter dumpFile)<br>A stub method implemented here to allow AbstractBeanComponent objects to be PibBlocks, stored and loaded directly to a PIB generated file format.                                 |
| static void                       | popupBeanDataDialog(AbstractComponent bean,java.awt.Window parent,boolean modal)<br>Creates a PropertySetDialog for the given object in the given model or toFront()'s an existing one.  |
| static void                       | popupBeanDataDialog(Object bean,AbstractModel model,AbstractComponent targetComponent,java.awt.Window parent,boolean modal)<br>Creates a PropertySetDialog for the given object in the given model or toFront()'s an existing one. |

[addComponentListener](#), [addConnection](#), [addMessage](#), [addMessage](#), [addToModel](#), [addToModel](#), [canConnectTo](#), [clearConnections](#), [clone](#), [complete](#), [connectTo](#), [connectTo](#), [copy](#), [createDrawnComponent](#), [createSourceData](#), [createTargetData](#), [DBTypeCode](#), [disconnect](#), [disconnectFrom](#), [fireComponentChanged](#), [fireComponentChanged](#), [fireComponentConnected](#), [fireComponentDeleted](#), [fireComponentDisconnected](#), [getCatCCCComparator](#), [getCategory](#), [getCCNumberComparator](#), [getComponent](#), [getComponentDependencies](#), [getConnectionCount](#), [getConnectionName](#), [getConnections](#), [getConnectionTypes](#), [getCustomPopupActions](#), [getCustomPopupItems](#), [getGroupedConnections](#), [getModel](#), [getName](#), [getNewCompIdent](#), [getOrder](#), [getOrderComparator](#), [getOwner](#), [getRealSize](#), [getSharedComponents](#), [includeInLoopcheck](#), [isOkayForExport](#), [isOkayForExport](#), [label](#), [popupDataDialog](#), [rebuildConnections](#), [reconnectImage](#), [removeComponentListener](#), [removeFromModel](#), [removeVerify](#), [restoreState](#), [setComponentNumber](#), [setDeleted](#), [setModel](#), [setOrder](#), [toString](#), [updateVersion](#), [writeName](#)

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#), [clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#), [deleteComment](#), [equals](#), [fixme](#), [getCCNumber](#), [getComment](#), [getComments](#), [getComments](#), [getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#), [getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#), [getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#), [popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#), [restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#), [setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setIdent](#), [setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#), [setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Constructors

(continued from last page)

## AbstractBeanComponent

```
public AbstractBeanComponent()
```

Create an AbstractBeanComponent object that does not belong to any model and has a display number of 0

## AbstractBeanComponent

```
public AbstractBeanComponent(AbstractModel model,  
                             int componentNumber)
```

Creates an AbstractBeanComponent object, adds it to model and sets the given component number on it.

### Parameters:

model - the AbstractModel to add this component to.

componentNumber - the component number to use; if 0 a new number will be given to the component.

## Methods

### copyFrom

```
public void copyFrom(GenericObject obj)
```

### dumpBlockParams

```
public boolean dumpBlockParams(java.io.PrintWriter dumpFile)
```

A stub method implemented here to allow AbstractBeanComponent objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

### writeBlockParams

```
public boolean writeBlockParams(com.apt.xdr.PibFile pibFile)
```

A stub method implemented here to allow AbstractBeanComponent objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

### readBlockParams

```
public boolean readBlockParams(com.apt.xdr.PibFile pibFile,  
                               int[] blockparm)
```

A stub method implemented here to allow AbstractBeanComponent objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

### popupBeanDataDialog

```
public static void popupBeanDataDialog(AbstractComponent bean,  
                                       java.awt.Window parent,  
                                       boolean modal)
```

Creates a PropertySetDialog for the given object in the given model or toFront()'s an existing one.

### Parameters:

bean - the AbstractComponent to show properties for

parent - the Window to set as the parent of the newly created dialog

modal - if true, the resulting dialog will be modal and block this Thread.

(continued on next page)

(continued from last page)

## popupBeanDataDialog

```
public static void popupBeanDataDialog(Object bean,  
    AbstractModel model,  
    AbstractComponent targetComponent,  
    java.awt.Window parent,  
    boolean modal)
```

Creates a PropertySetDialog for the given object in the given model or toFront()'s an existing one.

### Parameters:

bean - the Object to show properties for  
model - the AbstractModel containing the given bean and it's target component  
targetComponent - the AbstractComponent owner of the given bean  
parent - the Window to set as the parent of the newly created dialog  
modal - if true, the resulting dialog will be modal and block this Thread.

---

## popupDataDialog

```
public void popupDataDialog(java.awt.Window parent,  
    boolean modal)
```

Creates a new bean editing dialog for this object or resets and refreshes this object's current editing dialog.

---

## setComponentNumberConstrained

```
public void setComponentNumberConstrained(int val_)
```

Sets this bean-based component's component number to the given number. Conflicting numbers are reported to the user via MainFrame#addMessage.

### Parameters:

val\_ - the component number to use for this component

---

## copy

```
public AbstractComponent copy(AbstractModel sm)
```

the following copy method produces a deep clone of a composite base so that it can be put in a copy model clipboard. Connections to objects in the Vector which is the argument are preserved. Connections to all other objects are removed.  
NOTE: COPY/PASTE should use this in place of clone()

## com.cafean.client.analysis

### Class AbstractComponent

```

java.lang.Object
|
+-com.cafean.client.analysis.GenericObject
|
+-com.cafean.client.analysis.AbstractComponent

```

#### All Implemented Interfaces:

Checkable, [ComponentElement](#), Cloneable, [IdentHolder](#), StateEditable, Cloneable

#### Direct Known Subclasses:

[Connection](#), [ViewComponent](#), [AbstractBeanComponent](#)

public abstract class **AbstractComponent**

extends [GenericObject](#)

implements Cloneable, StateEditable, [IdentHolder](#), Cloneable, [ComponentElement](#), Checkable

The base class for ModelEditor Components.

#### See Also:

ComponentList, ComponentListener

**Fields inherited from class** [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

## Constructor Summary

|        |  |
|--------|--|
| public | AbstractComponent()<br>Create a AbstractComponent object that does not belong to any model and has a display number of 0                                       |
| public | AbstractComponent(AbstractModel model,int componentNumber)<br>Creates an AbstractComponent object, adds it to model and sets the given component number on it. |

## Method Summary

|      |  |
|------|--|
| void | addComponentListener(ComponentListener listener)<br>Adds a new component listener to this abstract component.                            |
| void | addConnection(Connection con)<br>The highest level of addConnection must be called after the connection has been added to the component. |
| void | addMessage(String text,ComponentElement obj,int severityCode)<br>Adds the given message as type severityCode                             |
| void | addMessage(String text,int severityCode)<br>Adds the given message as type severityCode  |



|                                   |  |
|-----------------------------------|--|
| void                              | addToModel( <code>AbstractModel model</code> )<br>This method adds this <code>AbstractComponent</code> to <code>model</code> .   |
| void                              | addToModel( <code>AbstractModel model,boolean adjustIdent</code> )<br>This method adds this <code>AbstractComponent</code> to <code>model</code> .   |
| boolean                           | canConnectTo( <code>AbstractComponent target</code> )<br>This method checks to see if it is allowable for this <code>AbstractComponent</code> to connect to the given <code>target</code> .  |
| void                              | clearConnections()<br>   |
| Object                            | clone()<br>this clone method copies all primitive data types.  |
| void                              | complete()<br>This completes this object's initialization in response to it's creation from a UI event.  |
| <a href="#">Connection</a>        | connectTo( <code>AbstractComponent target,ConnectionData tData,ConnectionData sData</code> )<br>This method connects this component to the given target component in the way described by the two <code>ConnectionData</code> objects.           |
| <a href="#">Connection</a>        | connectTo( <code>AbstractComponent target,ConnectionData tData,ConnectionData sData,int conID</code> )<br>This method connects this component to the given target component in the way described by the two <code>ConnectionData</code> objects. |
| <a href="#">AbstractComponent</a> | copy( <code>AbstractModel sm</code> )<br><b>Deprecated. June 2004</b>  |
| <a href="#">DrawnComponent</a>    | createDrawnComponent()<br>Returns the renderer for this abstract component.  |
| <a href="#">ConnectionData</a>    | createSourceData( <code>ConnectionData data</code> )<br>Ensures that the given <code>ConnectionData</code> object is suitable for a connection from this component.  |
| <a href="#">ConnectionData</a>    | createTargetData( <code>ConnectionData data</code> )<br>Ensures that the given <code>ConnectionData</code> object is suitable for a connection to this component.  |
| int                               | DBtypeCode()<br>Returns an int describing the component type.  |
| boolean                           | disconnect( <code>Connection con</code> )<br><b>This should only be used by</b> <code>Connection#disconnect</code> . Called by a <code>Connection</code> to remove itself from this component.   |
| void                              | disconnectFrom( <code>AbstractComponent target</code> )<br><b>Deprecated. February 2nd 2005</b>  |
| void                              | fireComponentChanged()<br>Notifies each <code>ComponentListener</code> registered with this component.   |

|                                     |   |
|-------------------------------------|---|
| void                                | <code>fireComponentChanged(ComponentChangedEvent evt)</code><br>Notifies each ComponentListener registered with this component of the given changed event.                          |
| void                                | <code>fireComponentConnected(Connection con)</code><br>This calls the component connected function on all of the listeners currently listening to this abstract component.          |
| void                                | <code>fireComponentDeleted()</code><br>This calls the component deleted function on all of the listeners currently listening to this abstract component                             |
| void                                | <code>fireComponentDisconnected(Connection con)</code><br>This calls the component disconnected function on all of the listeners currently listening to this abstract component.    |
| static Comparator                   | <code>getCatCCComparator()</code><br>Returns a comparator object for comparing the Category and component number of two AbstractComponent objects; for use in sorting or searching. |
| abstract <a href="#">Category</a>   | <code>getCategory()</code><br>Retrieves the most narrow category that this component is a member of.  |
| static Comparator                   | <code>getCCNumberComparator()</code><br>Returns a comparator object for comparing the component number of two AbstractComponent objects; for use in sorting or searching.           |
| <a href="#">AbstractComponent</a>   | <code>getComponent()</code>   |
| <a href="#">AbstractComponent[]</a> | <code>getComponentDependencies(List included)</code><br>Returns an array of the components that this component depends on and should carry with it during copy/paste operations.    |
| int                                 | <code>getConnectionCount()</code><br>Returns the number of connections contained in this component.   |
| String                              | <code>getConnectionName(Connection con)</code><br>This returns the name of this connection as perceived by this component.  |
| <a href="#">Connection[]</a>        | <code>getConnections()</code><br>This is the default "getConnections" function for abstract components.   |
| <a href="#">Connection[]</a>        | <code>getConnectionTypes()</code><br>Returns an empty instance of every Connection derivative that can be connected to this component type.   |
| Action[]                            | <code>getCustomPopupActions()</code><br>Creates Action objects for each of the actions that can be performed on this component.   |
| Vector                              | <code>getCustomPopupItems()</code><br>Creates Custom Menu Items for any popup dialog involving this component.  |
| <a href="#">Connection[][]</a>      | <code>getGroupedConnections()</code><br>Returns this component's Connections grouped by type.   |

|                                   |  |
|-----------------------------------|--|
| <a href="#">AbstractModel</a>     | <p>getModel()</p> <p>Return the AbstractModel to which this AbstractComponent belongs.</p>   |
| String                            | <p>getName()</p> <p>This is overwriting creates a default name for a component in case it is never given one on import.</p>  |
| int                               | <p>getNewCompIdent(int dbid,boolean preserveUnresolved,boolean useDbId)</p> <p>Retrieves the ident of the component with the given DB_ID in this component's model.</p>  |
| int                               | <p>getOrder()</p> <p>Gets this component's relative <i>ordervalue</i>.</p>   |
| static Comparator                 | <p>getOrderComparator()</p> <p>Returns a comparator object for comparing the component order of two AbstractComponent objects; for use in sorting or searching.</p>  |
| <a href="#">ComponentElement</a>  | <p>getOwner()</p>  |
| boolean                           | <p>getRealSize(Real length,Real avgDiam,Real avgAngle)</p> <p>This method is nearly a placeholder for the HydroComponent derivative.</p>   |
| <a href="#">SharedComponent[]</a> | <p>getSharedComponents()</p> <p>Returns an array of the SharedComponents that this component depends on and should carry with it during copy/paste operations.</p>   |
| boolean                           | <p>includeInLoopcheck()</p> <p>Determines if this component should be included in Loopcheck calculations.</p>  |
| boolean                           | <p>isOkayForExport()</p>   |
| boolean                           | <p>isOkayForExport(boolean prompt)</p>   |
| String                            | <p>label()</p> <p>Returns a String suitable for describing the component type on a dialog.</p>   |
| void                              | <p>popupDataDialog(java.awt.Window parent,boolean modal)</p> <p>Creates a new bean editing dialog for this object or resets and refreshes this object's current editing dialog.</p>  |
| void                              | <p>rebuildConnections(Vector sourceComps,Vector destComps)</p> <p>Rebuilds this component's related Connection objects using the given component Vectors as a guide. For direct references, use the source list to decide which connections to preserve and the image list to make the new connections. For ident references use reconnectIdentReferences.</p> |
| void                              | <p>reconnectImage(Vector sourceComps,Vector destComps)</p> <p>Reconnects the internal linkage of this component not handled by reconnectIdentReferences and rebuildConnections during a copy and/or paste operation. This method is called on the <b>source</b> component and is intended to rebuild appropriate structures for the destination component.</p> |
| void                              | <p>removeComponentListener(ComponentListener listener)</p> <p>Removes a component listener from this abstract component.</p>   |

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#), [clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#), [deleteComment](#), [equals](#), [fixme](#), [getCCNumber](#), [getComment](#), [getComments](#), [getComments](#), [getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#), [getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#), [getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#), [popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#), [restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#), [setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setIdent](#), [setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#), [setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Constructors

(continued from last page)

## AbstractComponent

```
public AbstractComponent()
```

Create a AbstractComponent object that does not belong to any model and has a display number of 0

## AbstractComponent

```
public AbstractComponent(AbstractModel model,  
                          int componentNumber)
```

Creates an AbstractComponent object, adds it to model and sets the given component number on it.

### Parameters:

model - the AbstractModel to add this component to.

componentNumber - the component number to use; if 0 a new number will be given to the component.

## Methods

### getCategory

```
public abstract Category getCategory()
```

Retrieves the most narrow category that this component is a member of.

### getConnectionTypes

```
public Connection\[\] getConnectionTypes()
```

Returns an empty instance of every Connection derivative that can be connected to this component type.

### Returns:

a Connection[] containing empty instances of each Connection type.

### addMessage

```
public void addMessage(String text,  
                       int severityCode)
```

Adds the given message as type severityCode

### Parameters:

text - a String containing the message to be displayed.

### addMessage

```
public void addMessage(String text,  
                       ComponentElement obj,  
                       int severityCode)
```

Adds the given message as type severityCode

### Parameters:

text - a String containing the message to be displayed.

### getSharedComponents

```
public SharedComponent\[\] getSharedComponents()
```

Returns an array of the SharedComponents that this component depends on and should carry with it during copy/paste operations.

(continued from last page)

**Returns:**

a SharedComponent[] containing references to all shared components that this component depends on. Duplicate references will not be copied multiple times.

---

## getComponentDependencies

```
public AbstractComponent\[\] getComponentDependencies(List included)
```

Returns an array of the components that this component depends on and should carry with it during copy/paste operations.

**Parameters:**

included - a List containing the components included in the current copy operation.

**Returns:**

an AbstractComponent[] containing references to all the components that this component depends on. Duplicate references will not be copied multiple times.

---

## clone

```
public Object clone()
```

this clone method copies all primitive data types. it will only need to be overridden if inherited classes contain objects which need to be copied. This should not be used for copy/paste as it initializes contained DrawnComponents and thus breaks the subclass copy() methods.

---

## copy

```
public AbstractComponent copy(AbstractModel sm)
```

**Deprecated.** *June 2004*

produces a deep clone of a component so that it can be put in a copy model clipboard.

---

## getNewCompIdent

```
public int getNewCompIdent(int dbid,  
    boolean preserveUnresolved,  
    boolean useDbId)
```

Retrieves the ident of the component with the given DB\_ID in this component's model. If dbid is 0, a 0 is returned.

**Parameters:**

preserveUnresolved - if true, and no component is found dbid will be returned. if false, 0 will be returned.  
useDbId - if true, ident references will be reconnected via find\_x\_ByDB\_ID; if false, ident references will be reconnected via find\_x\_ByIdent

---

## getOwner

```
public ComponentElement getOwner()
```

---

## getComponent

```
public AbstractComponent getComponent()
```

---

## getModel

```
public AbstractModel getModel()
```

Return the AbstractModel to which this AbstractComponent belongs. If not a member of a model, it will return null.

---

## setModel

```
public void setModel(AbstractModel model)
```

Set the AbstractModel. Note this call does not add the AbstractComponent object to the AbstractModel CBvector.

---

## addToModel

```
public void addToModel(AbstractModel model)
```

This method adds this AbstractComponent to model. If the AbstractComponent already belonged to a model, it will be removed from the old model first. A new ident will be assigned to this component when it is added. This call also calls setModel() for this AbstractComponent.

### Parameters:

model - the AbstractModel to add this component to

### See Also:

`AbstractModel.objectAdded()`

`AbstractModel.addComponent(AbstractComponent,boolean)()`

---

## addToModel

```
public void addToModel(AbstractModel model,  
    boolean adjustIdent)
```

This method adds this AbstractComponent to model. If the AbstractComponent already belonged to a model, it will be removed from the old model first. This call also calls setModel() for this AbstractComponent.

### Parameters:

model - the AbstractModel to add this component to

adjustIdent - if true, a new ident will be assigned to this component.

### See Also:

`AbstractModel.objectAdded()`

`AbstractModel.addComponent(AbstractComponent,boolean)()`

---

## removeFromModel

```
public void removeFromModel(AbstractModel model)
```

This method removes this AbstractComponent from model.

### Parameters:

model - the AbstractModel to remove this component from

---

## removeVerify

```
public boolean removeVerify()
```

Verifies that this component can be removed from it's model without unforeseen side effects and returns the truth of this assumption. This method may request user verification from the user.

---

## reconnectImage

```
public void reconnectImage(Vector sourceComps,  
    Vector destComps)
```

Reconnects the internal linkage of this component not handled by reconnectIdentReferences and rebuildConnections during a copy and/or paste operation. This method is called on the **source** component and is intended to rebuild appropriate structures for the destination component. (likely found by the destination's DB\_ID matching the source's ident)/.

---

(continued from last page)

**Parameters:**

`sourceComps` - a Vector containing the components in the source model being reconnected  
`destComps` - a Vector containing the components in the destination model being reconnected

---

## rebuildConnections

```
public void rebuildConnections(Vector sourceComps,  
                                Vector destComps)
```

Rebuilds this component's related Connection objects using the given component Vectors as a guide. For direct references, use the source list to decide which connections to preserve and the image list to make the new connections. For ident references use `reconnectIdentReferences`.

---

## label

```
public String label()
```

Returns a String suitable for describing the component type on a dialog. This default implementation returns the class name.

---

## getName

```
public String getName()
```

This is overwriting creates a default name for a component in case it is never given one on import.

**Returns:**

a String containing the name of a component, user-defined or generated.

---

## DBtypeCode

```
public int DBtypeCode()
```

Returns an int describing the component type. This is a plugin-specific value not actually used in the CAFEAN base code.

---

## setDeleted

```
public void setDeleted(boolean del)
```

If call with a parameter of true, the AbstractComponent object flagged as deleted. This is a plugin-specific value not actually used in the CAFEAN base code.

---

## popupDataDialog

```
public void popupDataDialog(java.awt.Window parent,  
                             boolean modal)
```

Creates a new bean editing dialog for this object or resets and refreshes this object's current editing dialog.

---

## getCustomPopupActions

```
public Action[] getCustomPopupActions()
```

Creates Action objects for each of the actions that can be performed on this component. null values in the returned array will be interpreted as separators.

---

## getCustomPopupItems

```
public Vector getCustomPopupItems()
```

Creates Custom Menu Items for any popup dialog involving this component. The resulting Vector should contain on JMenu, JMenuItem and JSeparator instances for this component.

**Returns:**



(continued from last page)

a Vector containing JMenu's, JMenuItem's, and JSeparators.

---

## setComponentNumber

```
public void setComponentNumber(int num)
```

Setter for Component number (user defines) also know as display number. Also updates the label on the component drawing.

### Parameters:

num - the number to which this value will be set

---

## canConnectTo

```
public boolean canConnectTo(AbstractComponent target)
```

This method checks to see if it is allowable for this AbstractComponent to connect to the given target.

### Parameters:

target - the AbstractComponent object to which a connection has been requested.

### Returns:

true if allowed, false if not allowed.

---

## disconnect

```
public boolean disconnect(Connection con)
```

**This should only be used by** Connection#disconnect. Called by a Connection to remove itself from this component.

### Parameters:

connect - the connection to be disconnected.

### Returns:

true if this component was connected through that connection.

---

## disconnectFrom

```
public void disconnectFrom(AbstractComponent target)
```

**Deprecated.** *February 2nd 2005*

Disconnects all the links between this component and the target. Use this if the target is being deleted otherwise use Connection#disconnect.

---

## complete

```
public void complete()
```

This completes this object's initialization in response to it's creation from a UI event. This method should be overridden by subclasses that require user input or default values for a newly created component.

---

## getRealSize

```
public boolean getRealSize(Real length,  
    Real avgDiam,  
    Real avgAngle)
```

This method is nearly a placeholder for the HydroComponent derivative.

---

(continued from last page)

## isOkayForExport

```
public boolean isOkayForExport()
```

---

## isOkayForExport

```
public boolean isOkayForExport(boolean prompt)
```

---

## getConnectionCount

```
public int getConnectionCount()
```

Returns the number of connections contained in this component.

---

## getConnections

```
public Connection\[\] getConnections()
```

This is the default "getConnections" function for abstract components. It is expected that components that contain connections should override this function.

### Returns:

An array of Connection objects for this component.

---

## addConnection

```
public void addConnection(Connection con)
```

The highest level of addConnection must be called after the connection has been added to the component. That means, it must be called last in any child versions of add connection.

---

## clearConnections

```
public void clearConnections()
```

---

## updateVersion

```
public void updateVersion()
```

Updates this object's current version to be greater than or equal to it's model's version. If the major versions are the same, this object's minor version is incremented;

---

## toString

```
public String toString()
```

---

## getCCNumberComparator

```
public static Comparator getCCNumberComparator()
```

Returns a comparator object for comparing the component number of two AbstractComponent objects; for use in sorting or searching.

---

## getOrderComparator

```
public static Comparator getOrderComparator()
```

---

(continued from last page)

Returns a comparator object for comparing the component order of two AbstractComponent objects; for use in sorting or searching.

---

## getCatCCComparator

```
public static Comparator getCatCCComparator()
```

Returns a comparator object for comparing the Category and component number of two AbstractComponent objects; for use in sorting or searching.

---

## connectTo

```
public final Connection connectTo(AbstractComponent target,  
    ConnectionData tData,  
    ConnectionData sData)
```

This method connects this component to the given target component in the way described by the two ConnectionData objects. This method should not be overridden. The version with the conID parameter should be overridden instead.

### Parameters:

- target - the AbstractComponent target of the connection operation.
- tData - the ConnectionData on the target where the tool was released.
- sData - the ConnectionData on the source where the connection was initiated.

---

## connectTo

```
public Connection connectTo(AbstractComponent target,  
    ConnectionData tData,  
    ConnectionData sData,  
    int conID)
```

This method connects this component to the given target component in the way described by the two ConnectionData objects.

### Parameters:

- target - the AbstractComponent target of the connection operation.
- tData - the ConnectionData on the target where the tool was released.
- sData - the ConnectionData on the source where the connection was initiated.
- conID - the connection number or junction CC number.

---

## createSourceData

```
public ConnectionData createSourceData(ConnectionData data)
```

Ensures that the given ConnectionData object is suitable for a connection from this component. This is normally used to create custom ConnectionData objects from SpecialConnectionData objects when connecting via the Connection tool.

### Parameters:

- data - the ConnectionData generated from the UI.

### Returns:

- the ConnectionData to use for connecting from this component.

---

## createTargetData

```
public ConnectionData createTargetData(ConnectionData data)
```

Ensures that the given ConnectionData object is suitable for a connection to this component. This is normally used to create custom ConnectionData objects from SpecialConnectionData objects when connecting via the Connection tool.

### Parameters:

- data - the ConnectionData generated from the UI.

(continued from last page)

**Returns:**

the ConnectionData to use for connecting to this component.

---

**createDrawnComponent**

```
public DrawnComponent createDrawnComponent ( )
```

Returns the renderer for this abstract component. This will be extended by any abstract component that needs a renderer.

**Returns:**

the DrawnComponent extending renderer for this abstract component

---

**addComponentListener**

```
public void addComponentListener(ComponentListener listener)
```

Adds a new component listener to this abstract component. If there are no listeners then this will create a new vector for the listeners to be in.

---

**removeComponentListener**

```
public void removeComponentListener(ComponentListener listener)
```

Removes a component listener from this abstract component. This should be called when that listener is destroyed

**Parameters:**

listener - the ComponentListener that is being removed from this component.

---

**fireComponentChanged**

```
public void fireComponentChanged()
```

Notifies each ComponentListener registered with this component. Creates a new ComponentChangedEvent to pass to  
{ @link #fireComponentChanged( ComponentChangedEvent evt ) }

---

**fireComponentChanged**

```
public void fireComponentChanged(ComponentChangedEvent evt)
```

Notifies each ComponentListener registered with this component of the given changed event.

---

**fireComponentDeleted**

```
public void fireComponentDeleted()
```

This calls the component deleted function on all of the listeners currently listening to this abstract component

---

**fireComponentConnected**

```
public void fireComponentConnected(Connection con)
```

This calls the component connected function on all of the listeners currently listening to this abstract component.

**Parameters:**

con - the Connection that has just been established.

---

**fireComponentDisconnected**

```
public void fireComponentDisconnected(Connection con)
```

This calls the component disconnected function on all of the listeners currently listening to this abstract component.

(continued from last page)

**Parameters:**

`con` - the Connection that has just been disconnected.

---

**getConnectionName**

```
public String getConnectionName(Connection con)
```

This returns the name of this connection as perceived by this component.

**Parameters:**

`con` - the Connection in question.

**Returns:**

a label for the connection describing it from the point of view of this component.

---

**getGroupedConnections**

```
public Connection[][] getGroupedConnections()
```

Returns this component's Connections grouped by type. The default implementation returns an empty array.

**Returns:**

a `Connection[][]` with each primary dimension being a type.

---

**restoreState**

```
public void restoreState(Hashtable state)
```

---

**includeInLoopcheck**

```
public boolean includeInLoopcheck()
```

Determines if this component should be included in Loopcheck calculations. This is a plugin-specific value not actually used in the CAFEAN core.

---

**getOrder**

```
public int getOrder()
```

Gets this component's relative *ord*value. The returned order is used as an additional sorting criteria for the Comparator returned by `#getOrderComparator`.

---

**setOrder**

```
public void setOrder(int order)
```

Sets this component's relative *ord*value. The given order is used as an additional sorting criteria for the Comparator returned by `#getOrderComparator`.

---

**writeName**

```
public String writeName()
```

Returns the string to be used when writing a component out to ASCII file. If the name is "unnamed", an empty String is returned.

---

## com.cafean.client.analysis

### Class AbstractModel

```

java.lang.Object
|
+-com.cafean.client.analysis.GenericObject
|
+-com.cafean.client.analysis.AbstractModel

```

#### All Implemented Interfaces:

[ModelElement](#), [IdentHolder](#), [StateEditable](#), [Cloneable](#)

public abstract class **AbstractModel**

extends [GenericObject](#)

implements [Cloneable](#), [StateEditable](#), [IdentHolder](#), [ModelElement](#)

An abstract representation of a logical model, it's contained components and elements, and facilities to manage them.

This is the central class for a ModelEditor analysis code plugin. All creating, storing, searching and loading goes through the model.

AbstractModel provides the following functionality to derivative models:

- Unique ident number allocation via `objectAdded`
- Categorized component access by component number, `ident`, `DB_ID` and `Iterator`
- A list and `Category` for `Connection` derivatives.
- A list of `ViewComponent` instances.
- A list of `Elements`, or (`GenericObjects`) accessible by `ident` and `DB_ID`

For more information on the creation of the `Category` hierarchy, see the `Category` documentation.

### Field Summary

|                                    |  |
|------------------------------------|--|
| static final <code>Category</code> | <code>CAT_CONNECTION</code><br>A <code>Category</code> for <code>Connection</code> objects.                              |
| static final <code>Category</code> | <code>CAT_CONSTANT</code><br>The <code>Category</code> inside this model that contains the user defined constants.       |
| static final <code>Category</code> | <code>CAT_DATA_SOURCE</code><br>A <code>Category</code> for <code>Data Sources</code> stored in this model               |
| static final <code>Category</code> | <code>CAT_FUNCTION</code><br>The <code>Category</code> inside this model that contains the user defined functions.       |
| static final <code>Category</code> | <code>CAT_NUMERICS</code><br>The <code>Category</code> inside this model that contains the user defined numerics.        |
| static final <code>Category</code> | <code>CAT_RANGE</code><br>A <code>Category</code> for <code>Data Sources</code> stored in this model                     |
| static final <code>Category</code> | <code>CAT_SOURCE_ROOT</code><br>A convenience <code>Category</code> for selecting all types of <code>Data Sources</code> |

|                       |   |
|-----------------------|---|
| static final Category | CAT_VALUES<br>The Category inside this model that contains all user defined values.           |
| static final Category | CAT_VARIABLE<br>The Category inside this model that contains the user user defined variables. |
| static final Category | CAT_VIEW<br>A Catagory for ViewComponent objects.   |
| static final String   | THEORYMAN<br>Value: <b>Theory Manual</b>  |
| static final String   | USERSMAN<br>Value: <b>User's Manual</b>   |

Fields inherited from class [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

## Constructor Summary

|        |  |
|--------|--|
| public | AbstractModel()<br>Creates a new AbstractModel with a new model ident, a default name and a default owner. |
|--------|--|

## Method Summary

|         |   |
|---------|---|
| void    | addComment(String comment)<br>{ @inheritDoc } Proxied to program options.   |
| void    | addComponent(AbstractComponent component)<br>Adds the given component to this model and gives it a new ident.                     |
| void    | addComponent(AbstractComponent component,boolean adjustIdent)<br>Adds the given component to this model.                          |
| void    | addComponentLookup(AbstractComponent comp,int dbid)<br>adds the given component to the component lookup table for the given dbid. |
| void    | addElement(GenericObject element)<br>Adds an element to the model, adjusting it's DB_ID and ident in the process.                 |
| void    | addElement(GenericObject element,boolean adjustIdent)<br>Adds a GenericObject to the model.                                       |
| void    | addMultipleComments(Vector com)<br>{ @inheritDoc } Proxied to program options.  |
| boolean | addPopupMenuEntries(JPopupMenu menu)<br>Appends this model's custom popup menu entries to the given popup menu.                   |

|                                   |   |
|-----------------------------------|---|
| boolean                           | <code>ccNumberCheck(Category category,String emptyMsg,String listName,String context,boolean warnEmpty)</code><br>Checks that all CC numbers in the given component list are valid and unique.                            |
| abstract boolean                  | <code>checkModel()</code><br>Performs checks for accuracy on the current AbstractModel  |
| void                              | <code>cleanUpDeleted()</code><br>Removes all deleted components from the lists maintained by this model.  |
| void                              | <code>clearComponentLookup()</code><br>clears the pasted component lookup table.  |
| void                              | <code>clearDbIds()</code><br>Clears the DB_ID's of this model's directly contained objects.   |
| int                               | <code>compareVersion(GenericObject obj)</code><br>Compares the version of the given object with that of this model to determine if the object's version is less than, equal to or greater than the version of this model. |
| boolean                           | <code>containsRestartChanges(Iterator iterator)</code><br>Returns true if any of the given objects is either deleted or has a major version number greater than or equal to that of this model.                           |
| <a href="#">AbstractComponent</a> | <code>createComponent(Category category)</code><br>Creates an appropriate AbstractComponent for the given leaf node category.   |
| void                              | <code>deleteAllComments()</code><br>{@inheritDoc} Proxied to program options.   |
| void                              | <code>deleteComment(int num)</code><br>{@inheritDoc} Proxied to program options.  |
| boolean                           | <code>executeModelValidations(boolean printErrors)</code><br>This executes all of the tests that are found in this model.   |
| void                              | <code>executeUserDefinedFunctions()</code><br>Executes all user defined functions in this model.  |
| boolean                           | <code>exportModelMetrics(java.io.File dFile)</code><br>Export a file containing the state of all testable metrics for the given model.  |
| boolean                           | <code>exportModelMetricsSpec(java.io.File dFile)</code><br>Export a file containing the metrics specification for the model.  |
| <a href="#">AbstractComponent</a> | <code>findComponentByCC(int cc,Category category)</code><br>Retrieves the component in this model that has the given CC or component number and is in a subset of the given category.                                     |
| <a href="#">AbstractComponent</a> | <code>findComponentByCC(int cc,String categoryName)</code><br>Retrieves the component in this model that has the given CC or component number and is in a subset of the given category.                                   |



|                                   |   |
|-----------------------------------|---|
| <a href="#">AbstractComponent</a> | <p><code>findComponentByDB_ID(int dbid)</code></p> <p>Retrieves the component in this model that has the given dbid by searching through each category.</p>                                       |
| <a href="#">AbstractComponent</a> | <p><code>findComponentByDB_ID(int dbid,Category category)</code></p> <p>Retrieves the component in this model that has the given dbid and is in a subset of the given category.</p>               |
| <a href="#">AbstractComponent</a> | <p><code>findComponentByIdent(int ident)</code></p> <p>Retrieves the component in this model that has the given ident by searching through each category.</p>                                     |
| <a href="#">AbstractComponent</a> | <p><code>findComponentByIdent(int ident,Category category)</code></p> <p>Retrieves the component in this model that has the given ident and is in a subset of the given category.</p>             |
| ComponentNumberGroup              | <p><code>findComponentGroup(Category category)</code></p> <p>Retrieves the appropriate ComponentNumberGroup instance for the given Category.</p>  |
| <a href="#">GenericObject</a>     | <p><code>findElementByDB_ID(int dbid)</code></p> <p>Retrieves the GenericObject in this model with the given DB_ID.</p>   |
| <a href="#">GenericObject</a>     | <p><code>findElementByIdent(int ident)</code></p> <p>Retrieves the element in this model with the given ident.</p>  |
| <a href="#">SharedComponent</a>   | <p><code>findEquivalentSharedComponent(SharedComponent shared)</code></p> <p>Retrieves an equivalent SharedComponent for the given component.</p>   |
| <a href="#">Real</a>              | <p><code>findReal(String siUnitsName)</code></p> <p>Finds the Real derivative who's SI units are equal to the given String or an empty Dimless if none is found.</p>                              |
| <a href="#">Category[]</a>        | <p><code>getCategories()</code></p> <p>Retrieves the root Categories for this model type.</p>   |
| Object                            | <p><code>getCategoryObject(Category category)</code></p> <p>Retrieves the optional object associated with the given category.</p>   |
| int                               | <p><code>getCCNumberIncrement()</code></p> <p>Retrieves the increment number for retrieved CC numbers.</p>  |
| String                            | <p><code>getComment(int num)</code></p> <p>{ @inheritDoc } Proxied to program options.</p>  |
| int                               | <p><code>getComponentCount()</code></p> <p>Retrieves a count of the components in this model by calling <code>getComponentCount</code> on each Category returned by { @link #getCategories }.</p> |
| int                               | <p><code>getComponentCount(Category category)</code></p> <p>Retrieves a count of the components in this model that are in a subset of the given category.</p>                                     |
| ComponentNumberGroup[]            | <p><code>getComponentGroups()</code></p> <p>Retrieves an array of the ComponentNumberGroup instances used by this model to ensure that each component has a valid component number.</p>           |

|  |  |
|--|--|
| Iterator   | <p><code>getComponentIterator(Category category)</code></p> <p>Creates an iterator suitable for traversing all the components that are in subsets of the given category.</p> |
| <a href="#">AbstractComponent[]</a>                        | <p><code>getComponents(Category category)</code></p> <p>Retrieves all components in this model that are in a subset of the given category.</p>                               |
| String   | <p><code>getCreateDate()</code></p> <p>Gets this model's creation date as a String.</p>  |
| <a href="#">DeckWriter</a>                                 | <p><code>getDeckWriter(String filename,boolean restart)</code></p> <p>Gets the DeckWriter implementation for this model if one is available.</p>                             |
| String   | <p><code>getDescription()</code></p> <p>{ @inheritDoc } Proxied to program options.</p>  |
| <a href="#">Real</a>                                       | <p><code>getDimensionless()</code></p> <p>Returns a new dimensionless value based on the units for this model's plugin.</p>  |
| <code>com.cafean.utils.ReferenceDocs.DocumentLink[]</code> | <p><code>getDocumentLinks(Object obj,String docType)</code></p> <p>Return the array of document links for a given object.</p>  |
| <a href="#">GenericObject</a>                              | <p><code>getElementAt(int i)</code></p> <p>Retrieves the element at the given index.</p>   |
| int  | <p><code>getElementCount()</code></p> <p>Retrieves a count of the elements contained in this model's element list.</p>   |
| Iterator   | <p><code>getElementIterator()</code></p> <p>Retrieves a new Iterator for use in traversing this model's Element list.</p>  |
| String   | <p><code>getExportNote()</code></p> <p>Getter for property exportNote.</p>   |
| int  | <p><code>getExportUnits()</code></p> <p>This gets the units for this model's ASCII export.</p>   |
| int  | <p><code>getInitialCCNumber(Category category)</code></p> <p>Retrieves the first CC number available to components of the given component class.</p>                         |
| int  | <p><code>getLastCCNumber(Category category)</code></p> <p>Retrieves the largest CC number in use that is available to components of the given component Category.</p>        |
| double   | <p><code>getLenScaleFactor()</code></p> <p>Gets the scale factor used to adjust cell length for drawn components rendering components of this model.</p>                     |
| <code>com.cafean.client.orginize.AbstractLoopCheck</code>  | <p><code>getLoopCheck(Vector components)</code></p> <p>Retrieves a new loop checker appropriate for checking hydraulic loops in this model.</p>                              |
| int  | <p><code>getMaxCCNumber(Category category)</code></p> <p>Retrieves the last CC number available to components of the given component Category.</p>                           |

|                               |   |
|-------------------------------|---|
| <a href="#">AbstractModel</a> | getModel()  |
| int                           | getModelDB_ID()<br>Retrieves this model's ID on the database server.  |
| int                           | getModelMajorVersion()<br>Returns the major version number of this model.   |
| int                           | getModelMinorVersion()<br>Retrieves this model's minor version number   |
| Object                        | getModelOptions()<br>Returns the java bean object that contains this model's "model options" if this model uses such a thing.   |
| String                        | getName()<br>{ @inheritDoc } Proxied to program options.  |
| int                           | getNewComponentIdent(int dbid,Category category,boolean preserveUnresolved,boolean useDbId)<br>Retrieves the ident of the Component with the given DB_ID in this model.             |
| static int                    | getNewElementIdent(AbstractModel model,int dbid,boolean preserveUnresolved,boolean useDbId)<br>Retrieves the ident of the element with the given DB_ID or ident in the given model. |
| int                           | getNextCCNumber(Category category)<br>Retrieves a unique and available CC number for the given component Category.  |
| int                           | getNextCCNumber(Category category,int requestedNumber)<br>Retrieves a new and available CC number for the given component Category.   |
| int                           | getNumComments()<br>{ @inheritDoc } Proxied to program options.   |
| String                        | getOwner()<br>Retrieves the username of the creator of this file.   |
| int                           | getParentDB_ID()<br>Retrieves the ID of this model's parent model in the database.  |
| <a href="#">MECodePlugin</a>  | getPlugin()<br>Returns a reference to the plugin that created this model.   |
| abstract String               | getPluginId()<br>Retrieves the name of this model's code family.  |
| ProgramOptions                | getProgramOptions()<br>Retrieves this model's ProgramOptions object, or null if this model type does not use ProgramOptions.  |
| int                           | getProjectDB_ID()<br>Retrieves the ID of this model's project in the database.  |

|                                     |  |
|-------------------------------------|--|
| <a href="#">Real</a>                | <p><code>getRealByIndex(int index)</code></p> <p>Returns the Real at the given index.</p>  |
| <a href="#">AbstractComponent[]</a> | <p><code>getRootComponents()</code></p> <p>Returns the set of AbstractComponents that should appear in the Navigator as peer nodes to ModelOptions.</p>  |
| <code>java.io.File</code>           | <p><code>getSaveFile()</code></p> <p>Retrieves the file this model was last saved to or opened from.</p>   |
| <code>String</code>                 | <p><code>getSaveFileName()</code></p> <p>Retrieves the name of the file this model was last saved to or opened from.</p>   |
| <code>int</code>                    | <p><code>getUnitIndex(Real unit)</code></p> <p>Finds the index of the given unit or -1 if the unit is not found inside this model.</p>   |
| <code>int</code>                    | <p><code>getUnitIndex(String siUnitsName)</code></p> <p>Finds the index of the unit who's SI units are equal to the given String or -1.</p>  |
| <code>int</code>                    | <p><code>getUnits()</code></p> <p>Returns the current units for this model.</p>  |
| <code>String[]</code>               | <p><code>getUnitsDisplay()</code></p> <p>Creates an array of strings for displaying model units for selection.</p>   |
| <code>ValidationOptions</code>      | <p><code>getValidationOptions()</code></p> <p>This gets the ValidationOptions that is stored inside a given model.</p>   |
| <code>ValidationTest[]</code>       | <p><code>getValidationTests()</code></p> <p>This gets the array of ValidationTests from the current model.</p>   |
| <code>double</code>                 | <p><code>getWidthScaleFactor()</code></p> <p>Gets the scale factor used to adjust cell width for drawn components rendering components of this model.</p>  |
| <code>boolean</code>                | <p><code>hasModelMetrics()</code></p> <p>Return true if this model supports output of model metrics.</p>   |
| <code>void</code>                   | <p><code>incrementMajorVersion()</code></p> <p>Increments this model's major version number and sets it's minor version to 0.</p>  |
| <code>void</code>                   | <p><code>incrementMinorVersion()</code></p>  |
| <code>boolean</code>                | <p><code>isDirty()</code></p> <p>if true, this model has been modified since it was saved.</p>   |
| <code>boolean</code>                | <p><code>isEditingRestart()</code></p> <p>Returns true if this model is subsequent edits to this model will be considered restart edits and as such, should be graphically represented.</p>  |
| <code>boolean</code>                | <p><code>isRestartableModel()</code></p> <p>Returns true if this model can be used to export a restart run. A restartable run is defined as one that has a parent dbid and has some components that have a major version number greater or equal to the major version of this model.</p> |

|                                   |   |
|-----------------------------------|---|
| boolean                           | <code>isValidCCNumber(int ccNumber,Category category)</code><br>Returns true if the given cc number is theoretically valid.   |
| boolean                           | <code>joinPipe(Vector selected)</code><br>This function joins two pipes into one.   |
| abstract void                     | <code>layoutComponents(Vector drawnComponents,com.cafean.utils.ProgressMon progress)</code><br>Lays out the given drawn components using a plugin-specific organization method. |
| <a href="#">DrawnComponent</a>    | <code>loadDrawnComponent(com.appt.xdr.PibBlock block)</code><br>Creates a custom DrawnComponent derivative from the given PibBlock.   |
| abstract void                     | <code>loadRestartData(com.cafean.utils.RestartData data)</code><br>Loads the given timeslice of RestartData into this model's components.                                       |
| <a href="#">AbstractComponent</a> | <code>lookupComponent(int dbid)</code><br>finds the component for the given dbid in the component lookup table  |
| void                              | <code>objectAdded(GenericObject object,boolean assignIdent)</code><br>Updates the model's top ident and the given object's ident and DB_ID upon addition to the model.          |
| boolean                           | <code>performLoopCheck(Vector components)</code><br>Performs a loop check on the given vector of components.  |
| void                              | <code>reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)</code><br>Reconnects the ident references of this models directly contained objects.                 |
| boolean                           | <code>removeComponent(AbstractComponent component)</code><br>Removes the given component from the model view.   |
| boolean                           | <code>removeElement(GenericObject element)</code><br>Removes the specified GenericObject from the model.  |
| boolean                           | <code>renodalizePipeCells(AbstractComponent component)</code><br>This function renodalizes a pipe cells.  |
| void                              | <code>renumberComponents(AbstractComponent[] components,int offset)</code><br>Ensures that the given components have appropriately unique CC numbers within their model.        |
| void                              | <code>reportModelCheck(boolean status)</code><br>Displays a report of the model check to the message window and a OptionPane.   |
| abstract void                     | <code>saveModel()</code><br>This method saves this AbstractModel to a SAM file using it's current save file or requesting a file if none is set.                                |
| abstract void                     | <code>saveModel(boolean showProgress)</code><br>This method saves this AbstractModel to a SAM file using it's current save file or requesting a file if none is set.            |
| void                              | <code>setCCNumberIncrement(int increment)</code><br>Sets the increment number for retrieved CC numbers.   |

|      |  |
|------|--|
| void | setCreateDate(String cDate)<br>Sets this model's creation date to the date in the given String.  |
| void | setDescription(String desc)<br>{ @inheritDoc } Proxied to program options.   |
| void | setDirty(boolean dirty)<br>sets this model dirty; a dirty model has been modified since it was saved.  |
| void | setExportNote(String exportNote)<br>Setter for property exportNote.  |
| void | setIdent(AbstractModel model)<br>Sets the ident of this model to that of the given model for use in copy/paste.  |
| void | setIdent(int uid)  |
| void | setLenScaleFactor(double fact)<br>Gets the scale factor used to adjust cell length for drawn components rendering components of this model.                                  |
| void | setModelDB_ID(int dbid)<br>Sets this model's DB_ID to the given dbid.  |
| void | setModelMajorVersion(int version)<br>Sets this model's major version number.   |
| void | setModelMinorVersion(int version)<br>Sets this model's minor version number.   |
| void | setName(String name)<br>{ @inheritDoc } Override GenericObject SetName to make sure view title is updated when this is called and to proxy the call into the ProgramOptions. |
| void | setOwner(String name)<br>Sets the username of the creator of this file.  |
| void | setParentDB_ID(int pdbid)<br>Sets this model's parent model's database ID.   |
| void | setProgramOptions(ProgramOptions options)<br>Sets the program options object that will be used by this model.  |
| void | setProjectDB_ID(int pdbid)<br>Sets this model's database project ID.   |
| void | setSaveFile(java.io.File file)<br>Sets the name of the file this model was last saved to or opened from.   |
| void | setUnits(int units)<br>Sets the current units for this model.  |
| void | setUnitsConstrained(int units)<br>Sets the units constrained by the available unit types.  |

|                       |  |
|-----------------------|--|
| void                  | setWidthScaleFactor(double fact)<br><br>Gets the scale factor used to adjust cell width for drawn components rendering components of this model. |
| String                | showComment(int num)<br><br>{ @inheritDoc } Proxied to program options.  |
| boolean               | splitPipe(Vector selected)<br><br>This function splits a pipe into two pipes.  |
| com.appt.xdr.PibBlock | storeDrawnComponent(java.awt.Component component)<br><br>Creates a custom PibBlock derivative from the given Component.                          |
| String                | toString()   |
| void                  | validateAllComponents()<br><br>Validates each component in every category in this model.   |

#### Methods inherited from class [com.cafean.client.analysis.GenericObject](#)

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#), [clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#), [deleteComment](#), [equals](#), [fixme](#), [getCCnumber](#), [getComment](#), [getComments](#), [getComments](#), [getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#), [getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#), [getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#), [popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#), [restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#), [setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setIdent](#), [setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#), [setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Fields

### CAT\_CONNECTION

public static final com.cafean.client.analysis.Category **CAT\_CONNECTION**

A Category for Connection objects.

### CAT\_DATA\_SOURCE

public static final com.cafean.client.analysis.Category **CAT\_DATA\_SOURCE**

A Category for Data Sources stored in this model

(continued from last page)

---

## CAT\_SOURCE\_ROOT

```
public static final com.cafean.client.analysis.Category CAT_SOURCE_ROOT
```

A convenience Category for selecting all types of Data Sources

---

## CAT\_RANGE

```
public static final com.cafean.client.analysis.Category CAT_RANGE
```

A Category for Data Sources stored in this model

---

## CAT\_VIEW

```
public static final com.cafean.client.analysis.Category CAT_VIEW
```

A Catagory for ViewComponent objects.

---

## CAT\_CONSTANT

```
public static final com.cafean.client.analysis.Category CAT_CONSTANT
```

The Category inside this model that contains the user defined constants.

---

## CAT\_VARIABLE

```
public static final com.cafean.client.analysis.Category CAT_VARIABLE
```

The Category inside this model that contains the user user defined variables.

---

## CAT\_FUNCTION

```
public static final com.cafean.client.analysis.Category CAT_FUNCTION
```

The Category inside this model that contains the user defined functions.

---

## CAT\_VALUES

```
public static final com.cafean.client.analysis.Category CAT_VALUES
```

The Category inside this model that contains all user defined values.

---

## CAT\_NUMERICS

```
public static final com.cafean.client.analysis.Category CAT_NUMERICS
```

The Category inside this model that contains the user defined numerics.

---

## USERSMAN

```
public static final java.lang.String USERSMAN
```

---

## THEORYMAN

```
public static final java.lang.String THEORYMAN
```

---

## Constructors



(continued from last page)

## AbstractModel

```
public AbstractModel()
```

Creates a new AbstractModel with a new model ident, a default name and a default owner.

## Methods

### getPlugin

```
public MECodePlugin getPlugin()
```

Returns a reference to the plugin that created this model. This implementation finds the plugin with a plugin id that matches this model's code family name.

**Returns:**

the MECodePlugin that create this model.

### getCategories

```
public Category[] getCategories()
```

Retrieves the root Categories for this model type. The root categories are the top level parent categories that among them contain all the components in this model. This is not determines on an instance by instance basis. All instances of a particular model should have the same categories. Derivatives should take care to include AbstractModel's categories in their getCategories method.

### getComponents

```
public AbstractComponent[] getComponents(Category category)
```

Retrieves all components in this model that are in a subset of the given category.

**Parameters:**

category - the Category that the desired components are a member

**Returns:**

an AbstractComponent[] containing the desired components.

### findComponentByCC

```
public AbstractComponent findComponentByCC(int cc,  
      Category category)
```

Retrieves the component in this model that has the given CC or component number and is in a subset of the given category.

**Parameters:**

cc - the component number of the desired component.

category - the Category that the desired component is a subset of.

**Returns:**

the AbstractComponent desired or null if not found.

### findComponentByCC

```
public AbstractComponent findComponentByCC(int cc,  
      String categoryName)
```

Retrieves the component in this model that has the given CC or component number and is in a subset of the given category.

**Parameters:**

cc - the component number of the desired component.

(continued from last page)

category - a String containing the name of the category

**Returns:**

the AbstractComponent desired or null if not found.

---

## findComponentByIdent

```
public AbstractComponent findComponentByIdent(int ident)
```

Retrieves the component in this model that has the given ident by searching through each category.

**Parameters:**

ident - the unique id number of the desired component.

**Returns:**

the AbstractComponent desired or null if not found.

---

## findComponentByIdent

```
public AbstractComponent findComponentByIdent(int ident,  
Category category)
```

Retrieves the component in this model that has the given ident and is in a subset of the given category.

**Parameters:**

ident - the unique id number of the desired component.

category - the Category that the desired component is a subset of.

**Returns:**

the AbstractComponent desired or null if not found.

---

## findComponentByDB\_ID

```
public AbstractComponent findComponentByDB_ID(int dbid,  
Category category)
```

Retrieves the component in this model that has the given dbid and is in a subset of the given category.

**Parameters:**

dbid - the DB\_ID of the desired component.

category - the Category that the desired component is a subset of.

**Returns:**

the AbstractComponent desired or null if not found.

---

## findComponentByDB\_ID

```
public AbstractComponent findComponentByDB_ID(int dbid)
```

Retrieves the component in this model that has the given dbid by searching through each category. Note: This is potentially a very expensive operation. Use #findComponentByDB\_ID(int,Category) wherever possible.

**Parameters:**

dbid - the DB\_ID of the desired component.

**Returns:**

the AbstractComponent desired or null if not found.

---

## addComponent

```
public void addComponent(AbstractComponent component)
```

---

(continued from last page)

Adds the given component to this model and gives it a new ident. The component's Category will be used to determine which internal list it is added to.

**Parameters:**

component - the AbstractComponent to add.

---

## addComponent

```
public void addComponent(AbstractComponent component,  
                           boolean adjustIdent)
```

Adds the given component to this model. The component's Category will be used to determine which internal list it is added to. If adjustIdent is true, a new ident will be assigned to the given component; if false, an ident will be assigned if the component's current ident is 0. If the component has a valid ident and adjustIdent is false, the top ident counter will be updated to include the component's current ident. Extending classes should ensure that #objectAdded is called on each component that **not** passed to this method.

**Parameters:**

component - the AbstractComponent to add.

adjustIdent - if true, the given component will be treated as a new component to this model and given a new ident;

**Throws:**

IllegalArgumentException - if the given component is in a Category that is not handled by this model.

---

## createComponent

```
public AbstractComponent createComponent(Category category)
```

Creates an appropriate AbstractComponent for the given leaf node category. Derivatives of createComponent should call this only if the given Category cannot be handled by the derivative model.

**Parameters:**

category - the Category of the desired new component.

**Returns:**

an AbstractComponent of the desired category

**Throws:**

java.lang.IllegalArgumentException - if there is no such category or the given category is a parent category.

---

## removeComponent

```
public boolean removeComponent(AbstractComponent component)
```

Removes the given component from the model view. Subclasses should override this method. Derivative methods should call this before removing a component.

**Parameters:**

component - the component to be removed from the current model

**Returns:**

true if the component removal was handled and a component deleted event was fired.

---

## getComponentIterator

```
public Iterator getComponentIterator(Category category)
```

Creates an iterator suitable for traversing all the components that are in subsets of the given category.

**Parameters:**

category - the Category to create an iterator for.

(continued from last page)

**Returns:**

an Iterator that will traverse only the subset of components given.

**See Also:**

`ComponentList.iterator(Category)()`

## getComponentCount

```
public int getComponentCount(Category category)
```

Retrieves a count of the components in this model that are in a subset of the given category. Note that this operation is  $O(1)$  for root categories and  $O(n)$  for subsets where  $n$  is the total number of components in this model.

**Parameters:**

`category` - the Category to retrieve a count for.

## getCategoryObject

```
public Object getCategoryObject(Category category)
```

Retrieves the optional object associated with the given category. The returned object is assumed to be a JavaBean with appropriate BeanInfo and editor set. This object's properties will be used whenever the Category's properties are desired (such as selection in the Navigator).

**Parameters:**

`category` - the Category for the desired properties delegate object

**Returns:**

the Object to use as the properties of the given Category or null if the given Category has no properties delegate

## validateAllComponents

```
public void validateAllComponents()
```

Validates each component in every category in this model.

**See Also:**

`GenericObject.validate()`

## getComponentCount

```
public int getComponentCount()
```

Retrieves a count of the components in this model by calling `getComponentCount` on each Category returned by `@link #getCategories`.

## getNewComponentIdent

```
public int getNewComponentIdent(int dbid,
    Category category,
    boolean preserveUnresolved,
    boolean useDbId)
```

Retrieves the ident of the Component with the given DB\_ID in this model. If dbid is 0, a 0 is returned.

**Parameters:**

`preserveUnresolved` - if true, and no Component is found dbid will be returned. if false, 0 will be returned.  
`useDbId` - if true, ident references will be reconnected via `findComponentByDB_ID`; if false, ident references will be reconnected via `findComponentByIdent`

(continued from last page)

---

## getPluginId

```
public abstract String getPluginId()
```

Retrieves the name of this model's code family.

**See Also:**

[com.cafean.CodePlugins.MEPluginData.getPluginId\(\)](#)

---

## saveModel

```
public abstract void saveModel()
```

This method saves this AbstractModel to a SAM file using it's current save file or requesting a file if none is set.

**See Also:**

`.getSaveFile()`

---

## saveModel

```
public abstract void saveModel(boolean showProgress)
```

This method saves this AbstractModel to a SAM file using it's current save file or requesting a file if none is set.

**See Also:**

`.getSaveFile()`

---

## loadRestartData

```
public abstract void loadRestartData(com.cafean.utils.RestartData data)
```

Loads the given timeslice of RestartData into this model's components.

**Parameters:**

`data` - the RestartData object containing the timeslice and related data to load.

---

## getDeckWriter

```
public DeckWriter getDeckWriter(String filename,  
    boolean restart)
```

Gets the DeckWriter implementation for this model if one is available.

**Parameters:**

`filename` - a String containing the full path of the file to be written.  
`restart` - if true, this DeckWriter should write a restart deck if possible

**Returns:**

a DeckWriter configured to write an input deck to the given file.

---

## addPopupMenuEntries

```
public boolean addPopupMenuEntries(JPopupMenu menu)
```

Appends this model's custom popup menu entries to the given popup menu. For example: A model's "Properties" item.

**Parameters:**

`menu` - the JPopupMenu to append this model's menu entries to.

**Returns:**

true if any entries were added.

---

## checkModel

```
public abstract boolean checkModel()
```

Performs checks for accuracy on the current AbstractModel

**Returns:**

the status of the model after checks

---

## getLoopCheck

```
public com.cafean.client.organize.AbstractLoopCheck getLoopCheck(Vector components)
```

Retrieves a new loop checker appropriate for checking hydraulic loops in this model.

**Parameters:**

`components` - the Vector of HydroComponent instances to check

**Returns:**

a LookCheck instance appropriate for this model

---

## performLoopCheck

```
public boolean performLoopCheck(Vector components)
```

Performs a loop check on the given vector of components.

**See Also:**

`.getLoopCheck()`

---

## getComponentGroups

```
public ComponentNumberGroup[] getComponentGroups()
```

Retrieves an array of the ComponentNumberGroup instances used by this model to ensure that each component has a valid component number.

**Returns:**

a ComponentNumberGroup[] containing this model's component number groups; or null if ComponentNumberGroups are not supported by this model.

---

## findComponentGroup

```
public ComponentNumberGroup findComponentGroup(Category category)
```

Retrieves the appropriate ComponentNumberGroup instance for the given Category.

**Parameters:**

`category` - the Category to retrieve a ComponentNumberGroup for.

**Returns:**

the ComponentNumberGroup used to ensure valid component numbers for the given Category of components.

---

## getInitialCCNumber

```
public int getInitialCCNumber(Category category)
```

Retrieves the first CC number available to components of the given component class. This is simply the beginning of the pool of possible CC numbers for the given cc group. NOTE: The returned CC number may or may not be in use.

**Returns:**

the lowest possible CC number for the given Category

---

(continued from last page)

**See Also:**`.getNextCCNumber(Category, int)()`

---

**getMaxCCNumber**

```
public int getMaxCCNumber(Category category)
```

Retrieves the last CC number available to components of the given component Category. This is simply the end of the pool of possible CC numbers for the given cc group. NOTE: The returned CC number may or may not be in use.

**Returns:**

the highest possible CC number for the given Category

**See Also:**`.getNextCCNumber(Category, int)()`

---

**setCCNumberIncrement**

```
public void setCCNumberIncrement(int increment)
```

Sets the increment number for retrieved CC numbers.

**Parameters:**

increment - the new increment used for allocating new cc numbers.

**See Also:**`.getNextCCNumber(Category, int)()`

---

**getCCNumberIncrement**

```
public int getCCNumberIncrement()
```

Retrieves the increment number for retrieved CC numbers.

**Returns:**

the increment used for allocating new cc numbers

**See Also:**`.getNextCCNumber(Category, int)()`

---

**getNextCCNumber**

```
public int getNextCCNumber(Category category)
```

Retrieves a unique and available CC number for the given component Category.

**Parameters:**

category - the Category of the component requiring a new CC number

**See Also:**`.getNextCCNumber(Category, int)()`

---

**getNextCCNumber**

```
public int getNextCCNumber(Category category,  
int requestedNumber)
```

Retrieves a new and available CC number for the given component Category. If the requested cc number of available it will be returned, otherwise the next available CC number will be returned. Available is defined as: greater than or equal to the initial cc number, lower than or equal to the maximum cc number and not currently used by another component in the given cc group.

(continued from last page)

**Parameters:**

`category` - the Category of the component requiring a new CC number  
`requestedNumber` - The desired CC number. If it is available, it will be returned.

**Returns:**

the requested cc number if available or a newly allocate cc number.

---

## getLastCCNumber

```
public int getLastCCNumber(Category category)
```

Retrieves the largest CC number in use that is available to components of the given component Category.

**Parameters:**

`category` - the Category of components to find the last cc for

**Returns:**

the largest cc number in use by the cc group for the given category

---

## isValidCCNumber

```
public boolean isValidCCNumber(int ccNumber,  
    Category category)
```

Returns true if the given cc number is theoretically valid.

---

## ccNumberCheck

```
public boolean ccNumberCheck(Category category,  
    String emptyMsg,  
    String listName,  
    String context,  
    boolean warnEmpty)
```

Checks that all CC numbers in the given component list are valid and unique. Errors are printed for invalid and duplicate cc numbers.

**Parameters:**

`category` - the Category of components to check CC numbers for.  
`emptyMsg` - a String containing the empty list message such as There are no  
`listName` - the name of the given list of components  
`context` - a String containing a description of the given context such as in this model.  
`warnEmpty` - if true a warning will be added to the MessageWindow if components is empty.

---

## layoutComponents

```
public abstract void layoutComponents(Vector drawnComponents,  
    com.cafean.utils.ProgressMon progress)
```

Lays out the given drawn components using a plugin-specific organization method.

**Parameters:**

`drawnComponents` - a Vector of DrawnComponent instances to be organized.  
`progress` - a ProgressMon to notify of progress.

---

## objectAdded

```
public void objectAdded(GenericObject object,  
    boolean assignIdent)
```

Updates the model's top ident and the given object's ident and DB\_ID upon addition to the model. If assignIdent is true, or object's ident is 0, a new ident will be assigned and object's DB\_ID will be set to it's previous ident.



(continued from last page)

**Parameters:**

`object` - the GenericObject being added to the model.  
`assignIdent` - if true, a new ident will be assigned, if false the top ident will be updated.

---

## reportModelCheck

```
public void reportModelCheck(boolean status)
```

Displays a report of the model check to the message window and a OptionPane.

**Parameters:**

`status` - the result of `checkModel()`

---

## removeElement

```
public boolean removeElement(GenericObject element)
```

Removes the specified GenericObject from the model.

**Parameters:**

`element` - the GenericObject to remove from the model

**Returns:**

true if removal successful, FALSE otherwise.

---

## addElement

```
public void addElement(GenericObject element)
```

Adds an element to the model, adjusting it's DB\_ID and ident in the process.

**Parameters:**

`element` - the GenericObject to add to the model.

---

## addElement

```
public void addElement(GenericObject element,  
boolean adjustIdent)
```

Adds a GenericObject to the model.

**Parameters:**

`element` - the GenericObject to add to the model.

`adjustIdent` - if true, the GenericObject's DB\_ID will be set to it's ident, and it's ident will be assigned to a new model-unique ident.

---

## getElementIterator

```
public Iterator getElementIterator()
```

Retrieves a new Iterator for use in traversing this model's Element list.

**Returns:**

an Iterator for traversing the element list

---

## getElementCount

```
public int getElementCount()
```

Retrieves a count of the elements contained in this model's element list.

(continued from last page)

**Returns:**

the number of elements in the element list.

---

**getElementAt**

```
public GenericObject getElementAt(int i)
```

Retrieves the element at the given index.

**Returns:**

the GenericObject at the given index in the element list

---

**findElementByIdent**

```
public GenericObject findElementByIdent(int ident)
```

Retrieves the element in this model with the given ident.

**Parameters:**

ident - the ident of the desired element.

**Returns:**

the GenericObject in the element list with the given ident

---

**findElementByDB\_ID**

```
public GenericObject findElementByDB_ID(int dbid)
```

Retrieves the GenericObject in this model with the given DB\_ID.

**Parameters:**

dbid - the DB\_ID of the desired element

---

**getNewElementIdent**

```
public static int getNewElementIdent(AbstractModel model,  
    int dbid,  
    boolean preserveUnresolved,  
    boolean useDbId)
```

Retrieves the ident of the element with the given DB\_ID or ident in the given model. If dbid is 0, a 0 is returned.

**Parameters:**

preserveUnresolved - if true, and no element is found dbid will be returned. if false, 0 will be returned.

useDbId - if true, ident references will be reconnected via findElementByDB\_ID; if false, ident references will be reconnected via findElementByIdent

---

**setModelDB\_ID**

```
public void setModelDB_ID(int dbid)
```

Sets this model's DB\_ID to the given dbid.

**Parameters:**

dbid - the new DB\_ID value to use

**See Also:**`.getModelDB_ID()`

---

**setIdent**

```
public void setIdent(AbstractModel model)
```

(continued from last page)

Sets the ident of this model to that of the given model for use in copy/paste. The model's ident is used to determine if a paste operation should treat the target model as the same model. When pasting to the same model some operations (such as copying shared components) are not performed.

---

## setIdent

```
public void setIdent(int uid)
```

---

## getModelDB\_ID

```
public int getModelDB_ID()
```

Retrieves this model's ID on the database server. Unlike that of AbstractComponents, this DB\_ID is actually a database ID. This is only used for submitting and restarting runs when a database server is being used.

**Returns:**

this model's database ID.

---

## setProjectDB\_ID

```
public void setProjectDB_ID(int pdbid)
```

Sets this model's database project ID.

**Parameters:**

pdbid - the new database project ID.

**See Also:**

[.getProjectDB\\_ID\(\)](#)

---

## getProjectDB\_ID

```
public int getProjectDB_ID()
```

Retrieves the ID of this model's project in the database. This is only used for submitting and restarting runs when a database server is being used.

**Returns:**

the ID of this model's project in the database.

---

## reconnectIdentReferences

```
public void reconnectIdentReferences(boolean preserveUnresolved,  
    boolean useDbId)
```

Reconnects the ident references of this models directly contained objects. These objects include { @link ComponentList component lists}, program options, model options, etc. Models that include root components are required to handle the reconnection of those components.

**Parameters:**

preserveUnresolved - if true, ident references that aren't resolvable will be left dangling, if false, they will be set to 0.  
useDbId - if true, ident references will be reconnected via find\_x\_ByDB\_ID; if false, ident references will be reconnected via find\_x\_ByIdent

**See Also:**

[com.cafean.client.analysis.GenericObject.reconnectIdentReferences\(\)](#)  
[.getNewComponentIdent\(\)](#)

(continued from last page)

---

## clearComponentLookup

```
public void clearComponentLookup()
```

clears the pasted component lookup table.

---

## lookupComponent

```
public AbstractComponent lookupComponent(int dbid)
```

finds the component for the given dbid in the component lookup table

---

## addComponentLookup

```
public void addComponentLookup(AbstractComponent comp,  
int dbid)
```

adds the given component to the component lookup table for the given dbid. Creates a new table if needed.

---

## clearDbIds

```
public void clearDbIds()
```

Clears the DB\_ID's of this model's directly contained objects. These objects include { @link ComponentList component lists}, program options, model options, etc. Models that include root components are required to handle the DB\_ID clearing of those components.

---

## setParentDB\_ID

```
public void setParentDB_ID(int pdbid)
```

Sets this model's parent model's database ID.

### Parameters:

pdbid - the new parent model ID.

### See Also:

`.getParentDB_ID()`

---

## getParentDB\_ID

```
public int getParentDB_ID()
```

Retrieves the ID of this model's parent model in the database. The parent model is the model that this model is a restart of. This is only used for submitting and restarting runs when a database server is being used.

### Returns:

the ID of this model's parent model in the database.

---

## getSaveFileName

```
public String getSaveFileName()
```

Retrieves the name of the file this model was last saved to or opened from.

### Returns:

a String containing the path to the file this model was last saved to or loaded from or null if neither has happened.

---

## getSaveFile

```
public java.io.File getSaveFile()
```

Retrieves the file this model was last saved to or opened from.

---

(continued from last page)

**Returns:**

the file this model was last saved to or opened from or null if neither has happened.

---

**setSaveFile**

```
public void setSaveFile(java.io.File file)
```

Sets the name of the file this model was last saved to or opened from.

**Parameters:**

name - a String containing the path to the file this model was last saved to or loaded from.

---

**compareVersion**

```
public int compareVersion(GenericObject obj)
```

Compares the version of the given object with that of this model to determine if the object's version is less than, equal to or greater than the version of this model. This method is primarily used to determine if a model has changes that justify a restart, and if so, which components should be part of that restart. Restartable components are those that have a version greater than or equal to that of their model.

**Parameters:**

obj - the GenericObject to compare major and minor versions with.

**Returns:**

-1 if obj's major is less than model's major or obj's minor is less than model's minor; 0 if model and obj's versions are the same; 1 if obj's major and minor are greater than model's

**See Also:**

[.containsRestartChanges\(\)](#)

---

**getModelMajorVersion**

```
public int getModelMajorVersion()
```

Returns the major version number of this model.

**Returns:**

the current major version number

**See Also:**

[.compareVersion\(\)](#)

---

**setModelMajorVersion**

```
public void setModelMajorVersion(int version)
```

Sets this model's major version number.

**Parameters:**

version - the new version number

**See Also:**

[.compareVersion\(\)](#)

---

**getModelMinorVersion**

```
public int getModelMinorVersion()
```

Retrieves this model's minor version number

**Returns:**

(continued from last page)

the current minor version number.

**See Also:**

`.compareVersion()`

---

## setModelMinorVersion

```
public void setModelMinorVersion(int version)
```

Sets this model's minor version number.

**Parameters:**

`version` - the new version number

**See Also:**

`.compareVersion()`

---

## incrementMajorVersion

```
public void incrementMajorVersion()
```

Increments this model's major version number and sets it's minor version to 0.

**See Also:**

`.compareVersion()`

---

## incrementMinorVersion

```
public void incrementMinorVersion()
```

**See Also:**

`.compareVersion()`

---

## cleanUpDeleted

```
public void cleanUpDeleted()
```

Removes all deleted components from the lists maintained by this model.

---

## isEditingRestart

```
public boolean isEditingRestart()
```

Returns true if this model is subsequent edits to this model will be considered restart edits and as such, should be graphically represented. Note: The base implementation returns false;

---

## isRestartableModel

```
public boolean isRestartableModel()
```

Returns true if this model can be used to export a restart run. A restartable run is defined as one that has a parent dbid and has some components that have a major version number greater or equal to the major version of this model. NOTE: Derived classes should note that a model cannot be restartable if it's parent dbid is  $\leq 0$ ;

---

## containsRestartChanges

```
public boolean containsRestartChanges(Iterator iterator)
```

Returns true if any of the given objects is either deleted or has a major version number greater than or equal to that of this model.

(continued from last page)

**Parameters:**

`iterator` - an Iterator through a container of `GenericObject` references to check deleted or compare major version numbers.

---

## setCreateDate

```
public void setCreateDate(String cDate)
```

Sets this model's creation date to the date in the given String. The given date is parsed with `DateFormat.getDateTimeInstance(DateFormat.MEDIUM, DateFormat.MEDIUM)` ;

**Parameters:**

`cDate` - a String containing the new creation date.

**See Also:**

`DateFormat.getDateTimeInstance()`

---

## getCreateDate

```
public String getCreateDate()
```

Gets this model's creation date as a String. return a String containing this model's creation date.

**See Also:**

`.setCreateDate()`

---

## getOwner

```
public String getOwner()
```

Retrieves the username of the creator of this file.

**Returns:**

a String containing the username of the creator of this file or the current user if no owner is set.

---

## setOwner

```
public void setOwner(String name)
```

Sets the username of the creator of this file.

**Parameters:**

`name` - a String containing the username of the creator of this file.

---

## splitPipe

```
public boolean splitPipe(Vector selected)
```

This function splits a pipe into two pipes. This method is part of the ModelEditor Renodalization system and should be overridden by models that wish to support the Split Pipe feature. This is the empty error version of this function to indicate which models don't support this feature

**Parameters:**

`selected` - the Vector of selected drawn components

---

## joinPipe

```
public boolean joinPipe(Vector selected)
```

This function joins two pipes into one. This method is part of the ModelEditor Renodalization system and should be overridden by models that wish to support the Join Pipe feature. This is the empty error version of this function to indicate which models don't support this feature.

(continued from last page)

**Parameters:**

selected - the Vector of selected drawn components

---

## renodalizePipeCells

```
public boolean renodalizePipeCells(AbstractComponent component)
```

This function renodalizes a pipe cells. This method is part of the ModelEditor Renodalization system and should be overridden by models that wish to support the Renodalize Cells feature. This is the empty error version of this function to indicate which models don't support this feature..

**Parameters:**

component - an AbstractComponent reference to the pipe to renodalize

---

## getName

```
public String getName()
```

Retrieves this object's name Proxied to program options.

---

## setName

```
public void setName(String name)
```

Sets this object's name to the given name after trimming it. Override GenericObject SetName to make sure view title is updated when this is called and to proxy the call into the ProgramOptions.

**Parameters:**

name - a String containing the new name.

---

## setDescription

```
public void setDescription(String desc)
```

Sets this object's description to the given desc. Proxied to program options.

**See Also:**

```
ProgramOptions.setDescription()
```

---

## getDescription

```
public String getDescription()
```

Retrieves the description of this object. Proxied to program options.

**See Also:**

```
ProgramOptions.getDescription()
```

---

## addComment

```
public void addComment(String comment)
```

Appends the given string as to the list of comments in this object Proxied to program options.

**See Also:**

```
ProgramOptions.addComment()
```

---

## addMultipleComments

```
public void addMultipleComments(Vector com)
```

Add the given Vector of Comments to this object. Proxied to program options.



(continued from last page)

**See Also:**`ProgramOptions.addMultipleComments()`

---

**getNumComments**`public int getNumComments()`

Retrieves the count of Comment objects stored in this object. Proxied to program options.

**See Also:**`ProgramOptions.getNumComments()`

---

**deleteComment**`public void deleteComment(int num)`

Removes the given Comment from this object's comment list. Proxied to program options.

**See Also:**`ProgramOptions.deleteComment()`

---

**deleteAllComments**`public void deleteAllComments()`

Removes all Comments from this object. Proxied to program options.

**See Also:**`ProgramOptions.deleteAllComments()`

---

**showComment**`public String showComment(int num)`

Retrieves the text in the comment at the given index. Proxied to program options.

**See Also:**`ProgramOptions.showComment()`

---

**getComment**`public String getComment(int num)`

Retrieves the comment at the given index. Proxied to program options.

**See Also:**`ProgramOptions.getComment()`

---

**getModelOptions**`public Object getModelOptions()`

Returns the java bean object that contains this model's "model options" if this model uses such a thing.

---

**getRootComponents**`public AbstractComponent\[\] getRootComponents()`

Returns the set of AbstractComponents that should appear in the Navigator as peer nodes to ModelOptions.

(continued from last page)

---

## getProgramOptions

```
public ProgramOptions getProgramOptions()
```

Retrieves this model's ProgramOptions object, or null if this model type does not use ProgramOptions.

**Returns:**

the ProgramOptions object for this model or null if none is used.

---

## setProgramOptions

```
public void setProgramOptions(ProgramOptions options)
```

Sets the program options object that will be used by this model.

**Parameters:**

options - a ProgramOptions object to be used for this model's options.

---

## renumberComponents

```
public void renumberComponents(AbstractComponent[] components,  
    int offset)
```

Ensures that the given components have appropriately unique CC numbers within their model. Sets new unique component numbers on all components with duplicate or invalid numbers.

**Parameters:**

components - the AbstractComponent[] containing the components to be renumbered

offset - the amount to offset each cc number by; if 0, renumber using the first available numbers for each component

---

## getLenScaleFactor

```
public double getLenScaleFactor()
```

Gets the scale factor used to adjust cell length for drawn components rendering components of this model.

---

## getWidthScaleFactor

```
public double getWidthScaleFactor()
```

Gets the scale factor used to adjust cell width for drawn components rendering components of this model.

---

## setLenScaleFactor

```
public void setLenScaleFactor(double fact)
```

Gets the scale factor used to adjust cell length for drawn components rendering components of this model.

---

## setWidthScaleFactor

```
public void setWidthScaleFactor(double fact)
```

Gets the scale factor used to adjust cell width for drawn components rendering components of this model.

---

## getDocumentLinks

```
public com.cafean.utils.ReferenceDocs.DocumentLink[] getDocumentLinks(Object obj,  
    String docType)
```

Return the array of document links for a given object.

---

(continued from last page)

---

## loadDrawnComponent

```
public DrawnComponent loadDrawnComponent (com.apt.xdr.PibBlock block)
```

Creates a custom DrawnComponent derivative from the given PibBlock.

**Returns:**

null if no custom components exist for the given block

---

## storeDrawnComponent

```
public com.apt.xdr.PibBlock storeDrawnComponent (java.awt.Component component)
```

Creates a custom PibBlock derivative from the given Component.

**Returns:**

null if no custom block exist for the given component

---

## executeUserDefinedFunctions

```
public void executeUserDefinedFunctions()
```

Executes all user defined functions in this model.

---

## findEquivalentSharedComponent

```
public SharedComponent findEquivalentSharedComponent (SharedComponent shared)
```

Retrieves an equivalent SharedComponent for the given component. Shared components are carried over when copy/pasting between models. This allows for such things as shared geometry references.

**See Also:**

`SharedComponent.isEquivalent()`

---

## getExportNote

```
public String getExportNote()
```

Getter for property exportNote.

**Returns:**

Value of property exportNote.

---

## setDirty

```
public void setDirty(boolean dirty)
```

sets this model dirty; a dirty model has been modified since it was saved.

---

## isDirty

```
public boolean isDirty()
```

if true, this model has been modified since it was saved.

---

## setExportNote

```
public void setExportNote(String exportNote)
```

Setter for property exportNote.

**Parameters:**

exportNote - New value of property exportNote.

---

---

## toString

```
public String toString()
```

---

---

## getUnitsDisplay

```
public String[] getUnitsDisplay()
```

Creates an array of strings for displaying model units for selection.

---

---

## findReal

```
public Real findReal(String siUnitsName)
```

Finds the Real derivative who's SI units are equal to the given String or an empty Dimless if none is found.

---

---

## getRealByIndex

```
public Real getRealByIndex(int index)
```

Returns the Real at the given index. This method allows an editor to select from the array of units by index.

---

---

## getUnitIndex

```
public int getUnitIndex(Real unit)
```

Finds the index of the given unit or -1 if the unit is not found inside this model.

---

---

## getUnitIndex

```
public int getUnitIndex(String siUnitsName)
```

Finds the index of the unit who's SI units are equal to the given String or -1.

---

---

## getDimensionless

```
public Real getDimensionless()
```

Returns a new dimensionless value based on the units for this model's plugin.

---

---

## getUnits

```
public int getUnits()
```

Returns the current units for this model.

---

---

## setUnits

```
public void setUnits(int units)
```

Sets the current units for this model.

---

---

## setUnitsConstrained

```
public void setUnitsConstrained(int units)
```

Sets the units constrained by the available unit types.

---

(continued from last page)

---

## getExportUnits

```
public int getExportUnits()
```

This gets the units for this model's ASCII export. The default is the same units that are used everywhere else. Some codes however are written to accept only files using certain units. This method is called before any ASCII is written out, including the AsciiViewer.

---

## getValidationTests

```
public ValidationTest[] getValidationTests()
```

This gets the array of ValidationTests from the current model. This defaults to returning an empty array.

---

## getValidationOptions

```
public ValidationOptions getValidationOptions()
```

This gets the ValidationOptions that is stored inside a given model. This defaults to return null.

---

## executeModelValidations

```
public boolean executeModelValidations(boolean printErrors)
```

This executes all of the tests that are found in this model.

**Parameters:**

printErrors - the boolean flag that determines if errors should be written to the MainFrame.

**Returns:**

true if all the tests passed.

---

## hasModelMetrics

```
public boolean hasModelMetrics()
```

Return true if this model supports output of model metrics.

---

## exportModelMetricsSpec

```
public boolean exportModelMetricsSpec(java.io.File dFile)
```

Export a file containing the metrics specification for the model.

**Parameters:**

dFile - the output file.

---

## exportModelMetrics

```
public boolean exportModelMetrics(java.io.File dFile)
```

Export a file containing the state of all testable metrics for the given model.

**Parameters:**

dFile - the output file.

---

## getModel

```
public AbstractModel getModel()
```

---

## com.cafean.client.analysis

### Class Category

java.lang.Object

└─com.cafean.client.analysis.Category

#### All Implemented Interfaces:

Cloneable, Comparable

```
public class Category
extends Object
implements Comparable, Cloneable
```

A node in a hierarchy of component types within a model.

Categories are used to represent each specific component type. Each of these categories can also have parent and child Categories representing subsets and supersets of component types.

Any given model should have a Category for each of its component types and a { @link AbstractModel#getCategories root Category } for each component or subset of components that are to be manipulated by the ModelEditor user interface.

**WARNING:**Do not check Category equivalence with ==. Use #equals instead as derivative plugins may need to replace a defined Category.

An abbreviated example of creating a hierarchy of Category objects:

```
public static final Category CAT_PIPE = new Category("Pipes", null, ICON_PIPE,
ImageMgr.class.getResource("pipe32.gif"));
public static final Category CAT_PUMP = new Category("Pumps", null, ICON_PUMP,
ImageMgr.class.getResource("pump32.gif"));
private static final Category[] HYDROS = { CAT_PIPE, CAT_PUMP };
public static final Category CAT_HYDRAULIC = new Category("Hydraulic Components", null,
ICON_HYDRAULIC, null,
HYDRAULIC_COMPONENTS );
private static final Category[] COMPONENTS = { CAT_HYDRAULIC, CAT_CONTROL_SYS };
```

With the above example, getCategories would then be overridden like this:

```
public Category[] getCategories()
{
    Category[] supers = super.getCategories();
    Category[] categories = new Category[COMPONENTS.length+supers.length];
    System.arraycopy(COMPONENTS, 0, categories, 0, COMPONENTS.length );
    System.arraycopy(supers, 0, categories, COMPONENTS.length, supers.length );
    return categories;
}
```

## Constructor Summary

|        |  |
|--------|--|
| public | <p>Category(String name,String tooltip,ImageIcon treeIcon,java.net.URL imageURL)</p> <p>Creates a new Category with the given name, tooltip, tree icon and image URL that is considered a visual Category.</p> |
|--------|--|

|        |  |
|--------|--|
| public | <pre>Category(String name,String tooltip,ImageIcon treeIcon,java.net.URL imageURL,boolean visual)</pre> <p>Creates a new Category with the given name, tooltip, tree icon, image URL, and visual status.</p> |
| public | <pre>Category(String name,String tooltip,ImageIcon treeIcon,java.net.URL imageURL,Category[] children)</pre> <p>Creates a new Category with the given name and child nodes.</p>                              |

## Method Summary

|                            |   |
|----------------------------|---|
| void                       | <pre>addChild(Category category)</pre> <p>Adds the given category as a child to this Category.</p>  |
| Object                     | <pre>clone()</pre> <p>Produces a shallow clone of this Category, including it's fields and child array but not the children themselves.</p> |
| int                        | <pre>compareTo(Object object)</pre>   |
| <a href="#">Category</a>   | <pre>deepClone()</pre> <p>Produces a deep clone of this Category, including it's fields and clones of it's child nodes.</p>                 |
| boolean                    | <pre>equals(Object o)</pre> <p>Checks equivalence based on reference equivalence or name equivalence.</p>                                   |
| <a href="#">Category[]</a> | <pre>getChildren()</pre> <p>Retrieves this Category's direct children.</p>  |
| ImageIcon                  | <pre>getCollapsedIcon()</pre> <p>Return the collapsed tree node icon for use in the Navigator component.</p>                                |
| ImageIcon                  | <pre>getExpandedIcon()</pre> <p>Return the expanded tree node icon for use in the Navigator component.</p>                                  |
| java.net.URL               | <pre>getImageURL()</pre> <p>Returns the URL of this Category's Tool Box image as retrieved from <code>Class.getResource</code>.</p>         |
| <a href="#">Category</a>   | <pre>getParent()</pre> <p>Retrieves this Category's direct parent.</p>  |
| <a href="#">Category</a>   | <pre>getRootParent()</pre> <p>Retrieves this Category's root parent.</p>  |
| String                     | <pre>getToolTipText()</pre> <p>Gets the tooltip text to use for this Category when name is not appropriate.</p>                             |
| boolean                    | <pre>isSubset(Category category)</pre> <p>Returns true if this category is a subset of the given category.</p>                              |
| boolean                    | <pre>isSuperset(Category category)</pre> <p>Returns true if this category is a superset of the given category.</p>                          |

|         |   |
|---------|---|
| boolean | <code>isVisual()</code><br>Returns true if this category is of Visual Components. |
| String  | <code>toString()</code>   |

#### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### Category

```
public Category(String name,
                 String tooltip,
                 ImageIcon treeIcon,
                 java.net.URL imageURL)
```

Creates a new Category with the given name, tooltip, tree icon and image URL that is considered a visual Category.

#### Parameters:

`name` - a String containing the desired name of this category.  
`tooltip` - a String containing the tooltip text to use for this Category  
`treeIcon` - the ImageIcon to use for this Category in the Navigator.  
`imageURL` - the URL of this Category's Tool Box image as retrieved from `Class.getResource`.

### Category

```
public Category(String name,
                 String tooltip,
                 ImageIcon treeIcon,
                 java.net.URL imageURL,
                 boolean visual)
```

Creates a new Category with the given name, tooltip, tree icon, image URL, and visual status.

#### Parameters:

`name` - a String containing the desired name of this category.  
`tooltip` - a String containing the tooltip text to use for this Category  
`treeIcon` - the ImageIcon to use for this Category in the Navigator.  
`imageURL` - the URL of this Category's Tool Box image as retrieved from `Class.getResource`.  
`visual` - if true, this Category is to be considered visual.

### Category

```
public Category(String name,
                 String tooltip,
                 ImageIcon treeIcon,
                 java.net.URL imageURL,
                 Category[] children)
```

Creates a new Category with the given name and child nodes. This Category will not be visual as it will have children.

#### Parameters:

`name` - a String containing the desired name of this category.  
`tooltip` - a String containing the tooltip text to use for this Category  
`treeIcon` - the ImageIcon to use for this Category in the Navigator.  
`imageURL` - the URL of this Category's Tool Box image as retrieved from `Class.getResource`.



(continued from last page)

children - a Category[] of child nodes.

## Methods

### addChild

```
public void addChild(Category category)
```

Adds the given category as a child to this Category.

### getChildren

```
public Category[] getChildren()
```

Retrieves this Category's direct children.

**Returns:**

a Category[] containing the direct child Categories

### getParent

```
public Category getParent()
```

Retrieves this Category's direct parent.

**Returns:**

the Category that is this Category's direct parent.

### getRootParent

```
public Category getRootParent()
```

Retrieves this Category's root parent. The root parent is the Category that is this Category's parent and has no parent of its own.

**Returns:**

the Category that is this Category's direct parent.

### isSuperset

```
public boolean isSuperset(Category category)
```

Returns true if this category is a superset of the given category.

**Parameters:**

category - the Category that this may or may not be a superset of

**Returns:**

true if this is a superset of the given Category

### isSubset

```
public boolean isSubset(Category category)
```

Returns true if this category is a subset of the given category.

**Parameters:**

category - the Category that this may or may not be a subset of

**Returns:**

true if this is a subset of the given Category

---

## compareTo

```
public int compareTo(Object object)
```

---

## toString

```
public String toString()
```

---

## getToolTipText

```
public String getToolTipText()
```

Gets the tooltip text to use for this Category when name is not appropriate.

**Returns:**

a String containing the tooltip for this Category

---

## isVisual

```
public boolean isVisual()
```

Returns true if this category is of Visual Components. Visual components have no children and can appear in views and the Tool Box. They should also have `DrawnComponents` defined.

---

## getExpandedIcon

```
public ImageIcon getExpandedIcon()
```

Return the expanded tree node icon for use in the Navigator component.

**Returns:**

the tree icon for this Category.

---

## getCollapsedIcon

```
public ImageIcon getCollapsedIcon()
```

Return the collapsed tree node icon for use in the Navigator component.

**Returns:**

the tree icon for this Category.

---

## getImageURL

```
public java.net.URL getImageURL()
```

Returns the URL of this Category's Tool Box image as retrieved from `Class.getResource`.

---

## equals

```
public boolean equals(Object o)
```

Checks equivalence based on reference equivalence or name equivalence.

**See Also:**

`Object.equals()`

---

(continued from last page)

## clone

```
public Object clone()
```

Produces a shallow clone of this Category, including it's fields and child array but not the children themselves.

**See Also:**

Cloneable

---

## deepClone

```
public Category deepClone()
```

Produces a deep clone of this Category, including it's fields and clones of it's child nodes.

**See Also:**

.clone()

Cloneable

## com.cafean.client.analysis Interface ComponentElement

All Superinterfaces:

[ModelElement](#)

All Known Implementing Classes:

[AbstractComponent](#)

---

public interface **ComponentElement**  
extends [ModelElement](#)

An interface describing an object that is contained by an [AbstractComponent](#) either directly or by being a part of an object that is.

---

### Method Summary

|                                   |  |
|-----------------------------------|--|
| <a href="#">AbstractComponent</a> | <code>getComponent ( )</code><br>Retrieves the <a href="#">AbstractComponent</a> that contains this <a href="#">ComponentElement</a> . This method may call a parent's <code>getComponent()</code> and so on rather than a direct reference. |
| <a href="#">ComponentElement</a>  | <code>getOwner ( )</code><br>Retrieves the owner of this <a href="#">ComponentElement</a> .  |

---

Methods inherited from interface [com.cafean.client.analysis.ModelElement](#)

[getModel](#)

---

### Methods

#### getComponent

public [AbstractComponent](#) **getComponent ( )**

Retrieves the [AbstractComponent](#) that contains this [ComponentElement](#). This method may call a parent's `getComponent()` and so on rather than a direct reference.

**Returns:**

the [AbstractModel](#) that contains this [ModelElement](#)

---

#### getOwner

public [ComponentElement](#) **getOwner ( )**

Retrieves the owner of this [ComponentElement](#).

## com.cafean.client.analysis

### Class ComponentList

```

java.lang.Object
|
+-com.cafean.client.analysis.ElementList
   |
   +-com.cafean.client.analysis.ComponentList

```

**All Implemented Interfaces:**  
Cloneable, Cloneable

```

public class ComponentList
extends ElementList
implements Cloneable, Cloneable

```

A storage class for AbstractComponent instances.

ComponentList requires that all it's contained components have unique ident numbers.

Components are optionally stored and retrieved in sorted order via binary searches for efficiency.

### Constructor Summary

|        |   |
|--------|---|
| public | ComponentList(Category category,boolean sorted)<br><br>Creates a new instance of ComponentList with the given category name and the given initial sorted state. |
|--------|---|

### Method Summary

|                                     |  |
|-------------------------------------|--|
| <a href="#">AbstractComponent</a>   | findByCC(int cc,Category category)<br><br>Retrieves the AbstractComponent with the given component number.                       |
| boolean                             | isContained(Category category)<br><br>Returns true if the given Category is contained in this list.                              |
| Iterator                            | iterator(Category category)<br><br>Retutrnns an Iterator object for iterating through this list.                                 |
| int                                 | size(Category category)<br><br>Returns the number of components in this list that are part of the given category.                |
| <a href="#">AbstractComponent[]</a> | toArray(Category category)<br><br>Returns an array containing all of the components in this list that are in the given Category. |
| String                              | toString()<br><br>Returns a String representation of this ComponentList.   |

Methods inherited from class [com.cafean.client.analysis.ElementList](#)

[add](#), [clear](#), [clearDbIds](#), [clone](#), [contains](#), [findByDB\\_ID](#), [findByIdent](#), [findByIdent](#), [get](#), [getCategory](#), [indexOf](#), [isSorted](#), [iterator](#), [reconnectIdentReferences](#), [remove](#), [removeDeleted](#), [setCategory](#), [size](#), [sort](#), [toArray](#), [toArray](#), [toString](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### ComponentList

```
public ComponentList(Category category,
                     boolean sorted)
```

Creates a new instance of ComponentList with the given category name and the given initial sorted state.

**Parameters:**

`category` - the Category that is the superset of all components that will be stored in this ComponentList.  
`sorted` - if true all additions will be inserted in thier sorted positions.

**See Also:**

`.sort()`

## Methods

### findByCC

```
public AbstractComponent findByCC(int cc,
                                   Category category)
```

Retrieves the AbstractComponent with the given component number. if the component number is 0 null is returned.

**Parameters:**

`cc` - the cc number of the desired component.  
`category` - the Category of the desired component

**Returns:**

an AbstractComponent corresponding to the search parameters

### size

```
public int size(Category category)
```

Returns the number of components in this list that are part of the given category. If the category is this list's category, `#size()` is returned.

**Parameters:**

`category` - the Category to return the number of components in.

### toArray

```
public AbstractComponent[] toArray(Category category)
```

Returns an array containing all of the components in this list that are in the given Category. This will return an empty array if there are no components inside the list with the given category.

(continued from last page)

**Parameters:**

`category` - the Category to components for

**Returns:**

an `AbstractComponent[]` containing in the given Category.

---

**toString**

```
public String toString()
```

Returns a String representation of this ComponentList.

---

**iterator**

```
public Iterator iterator(Category category)
```

Returns an Iterator object for iterating through this list.

**Returns:**

an Iterator suitable for traversing the components in this ComponentList that are in the given Category.

---

**isContained**

```
public boolean isContained(Category category)
```

Returns true if the given Category is contained in this list.

**Parameters:**

`category` - the Category to check

## com.cafean.client.analysis Interface ComponentListener

### All Known Implementing Classes:

DrawnComponent, AsciiViewer

public interface **ComponentListener**

A component listener listens to an abstract component for commands to update their rendering of the component.

### Method Summary

|      |  |
|------|--|
| void | componentChanged(ComponentChangedEvent evt)<br>A component changed event tells the listener that the internal data of the component has changed.       |
| void | componentConnected(Connection con)<br>A component connected event tells the listener when a component completes a connection to a different component. |
| void | componentDeleted()<br>A component deleted event tells the listener that the component has been deleted by the user.                                    |
| void | componentDisconnected(Connection con)<br>A component disconnected event tells the listener when a component disconnects from a different compnoent.    |

### Methods

#### componentChanged

public void **componentChanged**([ComponentChangedEvent](#) evt)

A component changed event tells the listener that the internal data of the component has changed. This causes DrawnComponents to re-initialize their data.

#### componentDeleted

public void **componentDeleted**()

A component deleted event tells the listener that the component has been deleted by the user. The listener should stop rendering that object and remove itself from the model as well.

#### componentConnected

public void **componentConnected**([Connection](#) con)

A component connected event tells the listener when a component completes a connection to a different component.

#### Parameters:

con - The Connection that has just been created.



## **componentDisconnected**

```
public void componentDisconnected(Connection con)
```

A component disconnected event tells the listener when a component disconnects from a different compnoent.

## com.cafean.client.analysis Class Connection

```

java.lang.Object
|
+-com.cafean.client.analysis.GenericObject
|
|+-com.cafean.client.analysis.AbstractComponent
|
|+-com.cafean.client.analysis.Connection

```

**Direct Known Subclasses:**

[ConnectionBean](#)

public abstract class **Connection**  
extends [AbstractComponent](#)

A class representing a connection between two components in an AbstractModel. The two sides are referred to as left and right and have their idents stored in the Connection.

ConnectionData objects are used to describe each side and can be used, for instance, to determine which ConnectingPt in a DrawnComponent corresponds to a given Connection, or passed to AbstractComponent#connectTo to describe the desired Connection.

The convenience methods #getThisSideData, #getOtherSide and #getOtherSideData have been provided to simplify Connection related code.

**See Also:**

DrawnComponent#createConnectionPt

**Fields inherited from class** [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

### Constructor Summary

|        |   |
|--------|---|
| public | Connection()<br>Creates a new instance of Connection with no model and no left or right component.  |
| public | Connection(AbstractModel model, AbstractComponent leftComp)<br>Creates a new instance of Connection with the given model and the given left component.                                      |
| public | Connection(AbstractModel model, AbstractComponent leftComp, int cc)<br>Creates a new instance of Connection with the given model and the given left component.                              |
| public | Connection(AbstractModel model, int leftComp, int rightComp, int cc)<br>Creates a new instance of Connection with the given model and the given left component.                             |
| public | Connection(AbstractModel model, AbstractComponent leftComp, AbstractComponent rightComp, int cc)<br>Creates a new instance of Connection with the given model and the given left component. |

## Method Summary

|   |  |
|---|--|
| Object                                  | clone()<br>Creates a DrawnConnection to render this Connection.  |
| <a href="#">DrawnComponent</a>          | createDrawnComponent()<br>Creates a DrawnConnection to render this Connection.   |
| void                                    | disconnect()<br>Disconnects this Connection between two AbstractComponent objects.   |
| <a href="#">Category</a>                | getCategory()<br>{ @inheritDoc } NOTE: This is the Category of the Connection class and has no relevance to the components it connects.                              |
| java.awt.Color                          | getConnectionColor()<br>Returns the Color used to paint this Connection in a View.   |
| java.awt.Stroke                         | getConnectionStroke()<br>Returns the Stroke used to paint this connection in a View.   |
| Vector                                  | getCustomPopupItems()<br>Creates Custom Menu Items for any popup dialog involving this component   |
| String                                  | getDocDescription(AbstractComponent side,boolean includeComponent)<br>Returns an HTML formatted documentation description from of the given side of this connection. |
| <a href="#">AbstractComponent</a>       | getLeftComponent()<br>Retrieves the left component of this connection via ident.   |
| int                                     | getLeftComponentID()<br>Retrieves the ident of the left component of this connection.  |
| abstract <a href="#">ConnectionData</a> | getLeftConnectData()<br>Retrieves a ConnectionData object describing the connection to the component on the left of this connection.                                 |
| <a href="#">AbstractComponent</a>       | getOtherSide(AbstractComponent component)<br>Retrieves the component that is on the other side of this Connection.   |
| <a href="#">ConnectionData</a>          | getOtherSideData(AbstractComponent component)<br>Retrieves the ConnectionData for the other side of this Connection.   |
| <a href="#">AbstractComponent</a>       | getRightComponent()<br>Retrieves the right component of this connection via ident.   |
| int                                     | getRightComponentID()<br>Retrieves the ident of the right component of this connection.  |
| abstract <a href="#">ConnectionData</a> | getRightConnectData()<br>Retrieves a ConnectionData object describing the connection to the component on the right of this connection.                               |

|                                |  |
|--------------------------------|--|
| <a href="#">ConnectionData</a> | <p><code>getThisSideData(AbstractComponent component)</code></p> <p>Retrieves the ConnectionData for this side of this Connection.</p>   |
| boolean                        | <p><code>isEqualTo(Connection con)</code></p> <p>This determines if the Connection passed in is the equivalent of this Connection.</p>   |
| boolean                        | <p><code>isIndependentComponent()</code></p> <p>Returns true if this Connection can exist without its left and right components. Non-Independent connections are deleted if and when their surrounding components are no longer available.</p> |
| boolean                        | <p><code>isVisual()</code></p> <p>Returns true if this connection should be visually represented.</p>  |
| String                         | <p><code>label()</code></p>  |
| void                           | <p><code>reconnectIdentReferences(boolean preserveUnresolved, boolean useDbId)</code></p>  |
| void                           | <p><code>setLeftComponent(AbstractComponent component)</code></p> <p>Sets the left component of this connection.</p>   |
| void                           | <p><code>setLeftComponentID(int ident)</code></p> <p>Sets the left component of this Connection by ident.</p>  |
| void                           | <p><code>setRightComponent(AbstractComponent component)</code></p> <p>Sets the right component of this connection.</p>   |
| void                           | <p><code>setRightComponentID(int ident)</code></p> <p>Sets the right component of this Connection by ident.</p>  |
| String                         | <p><code>toString()</code></p>   |
| void                           | <p><code>userDisconnect()</code></p> <p>Disconnects the connection from the Navigator.</p>   |

#### Methods inherited from class [com.cafean.client.analysis.AbstractComponent](#)

[addComponentListener](#), [addConnection](#), [addMessage](#), [addMessage](#), [addToModel](#), [addToModel](#), [canConnectTo](#), [clearConnections](#), [clone](#), [complete](#), [connectTo](#), [connectTo](#), [copy](#), [createDrawnComponent](#), [createSourceData](#), [createTargetData](#), [DBTypeCode](#), [disconnect](#), [disconnectFrom](#), [fireComponentChanged](#), [fireComponentChanged](#), [fireComponentConnected](#), [fireComponentDeleted](#), [fireComponentDisconnected](#), [getCatCCCComparator](#), [getCategory](#), [getCCNumberComparator](#), [getComponent](#), [getComponentDependencies](#), [getConnectionCount](#), [getConnectionName](#), [getConnections](#), [getConnectionTypes](#), [getCustomPopupActions](#), [getCustomPopupItems](#), [getGroupedConnections](#), [getModel](#), [getName](#), [getNewCompIdent](#), [getOrder](#), [getOrderComparator](#), [getOwner](#), [getRealSize](#), [getSharedComponents](#), [includeInLoopcheck](#), [isOkayForExport](#), [isOkayForExport](#), [label](#), [popupDataDialog](#), [rebuildConnections](#), [reconnectImage](#), [removeComponentListener](#), [removeFromModel](#), [removeVerify](#), [restoreState](#), [setComponentNumber](#), [setDeleted](#), [setModel](#), [setOrder](#), [toString](#), [updateVersion](#), [writeName](#)

#### Methods inherited from class [com.cafean.client.analysis.GenericObject](#)

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#), [clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#), [deleteComment](#), [equals](#), [fixme](#), [getCCnumber](#), [getComment](#), [getComments](#), [getComments](#), [getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#), [getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#), [getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#), [popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#), [restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#), [setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setIdent](#), [setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#), [setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### Connection

```
public Connection()
```

Creates a new instance of Connection with no model and no left or right component.

### Connection

```
public Connection(AbstractModel model,
                  AbstractComponent leftComp)
```

Creates a new instance of Connection with the given model and the given left component.

#### Parameters:

`model` - the `AbstractModel` that this `Connection` is to be part of.  
`leftComp` - the `AbstractComponent` that is the left component of this connection.

### Connection

```
public Connection(AbstractModel model,
                  AbstractComponent leftComp,
                  int cc)
```

Creates a new instance of Connection with the given model and the given left component.

#### Parameters:

`model` - the `AbstractModel` that this `Connection` is to be part of.  
`leftComp` - the `AbstractComponent` that is the left component of this Connection.  
`cc` - the component number of this Connection

### Connection

```
public Connection(AbstractModel model,
                  int leftComp,
                  int rightComp,
                  int cc)
```

Creates a new instance of Connection with the given model and the given left component.

(continued from last page)

**Parameters:**

model - the AbstractModel that this Connection is to be part of.  
leftComp - the ident of the AbstractComponent that is the left component of this Connection.  
rightComp - the ident of the AbstractComponent that is the right component of this Connection.  
cc - the component number of this Connection

---

## Connection

```
public Connection(AbstractModel model,  
                  AbstractComponent leftComp,  
                  AbstractComponent rightComp,  
                  int cc)
```

Creates a new instance of Connection with the given model and the given left component.

**Parameters:**

model - the AbstractModel that this Connection is to be part of.  
leftComp - the AbstractComponent that is the left component of this Connection.  
rightComp - the AbstractComponent that is the right component of this Connection.  
cc - the component number of this Connection

## Methods

### getLeftConnectData

```
public abstract ConnectionData getLeftConnectData()
```

Retrieves a ConnectionData object describing the connection to the component on the left of this connection.

---

### getRightConnectData

```
public abstract ConnectionData getRightConnectData()
```

Retrieves a ConnectionData object describing the connection to the component on the right of this connection.

---

### isIndependentComponent

```
public boolean isIndependentComponent()
```

Returns true if this Connection can exist without its left and right components. Non-Independent connections are deleted if and when their surrounding components are no longer available.

---

### getOtherSide

```
public AbstractComponent getOtherSide(AbstractComponent component)
```

Retrieves the component that is on the other side of this Connection. If the given component is the left, this returns the right and vice versa.

**Returns:**

component the AbstractComponent that is the other side of this Connection or null if there is no other side.

**Throws:**

IllegalArgumentException - if the given AbstractComponent is neither side of this Connection.

---

### getOtherSideData

```
public ConnectionData getOtherSideData(AbstractComponent component)
```

Retrieves the ConnectionData for the other side of this Connection. If the given component is the left, this returns the right and vice versa.

(continued from last page)

**Parameters:**

`component` - the `AbstractComponent` that is the other side of this `Connection`.

**Throws:**

`IllegalArgumentException` - if the given `AbstractComponent` is neither side of this `Connection`.

---

## getThisSideData

```
public ConnectionData getThisSideData(AbstractComponent component)
```

Retrieves the `ConnectionData` for this side of this `Connection`.

**Parameters:**

`component` - the `AbstractComponent` that is the desired side of this `Connection`.

**Throws:**

`IllegalArgumentException` - if the given `AbstractComponent` is neither side of this `Connection`.

---

## getLeftComponent

```
public AbstractComponent getLeftComponent()
```

Retrieves the left component of this connection via ident. NOTE: Derivatives of `Component` should override this where possible to use an appropriate category for `findComponentByIdent`.

---

## getLeftComponentID

```
public int getLeftComponentID()
```

Retrieves the ident of the left component of this connection.

---

## setLeftComponentID

```
public void setLeftComponentID(int ident)
```

Sets the left component of this `Connection` by ident.

---

## setLeftComponent

```
public void setLeftComponent(AbstractComponent component)
```

Sets the left component of this connection.

**Parameters:**

`component` - the `AbstractComponent` to use as this connection's left component

---

## getRightComponent

```
public AbstractComponent getRightComponent()
```

Retrieves the right component of this connection via ident. NOTE: Derivatives of `Component` should override this where possible to use an appropriate category for `findComponentByIdent`.

---

## getRightComponentID

```
public int getRightComponentID()
```

Retrieves the ident of the right component of this connection.

(continued from last page)

---

## setRightComponentID

```
public void setRightComponentID(int ident)
```

Sets the right component of this Connection by ident.

---

## setRightComponent

```
public void setRightComponent(AbstractComponent component)
```

Sets the right component of this connection.

### Parameters:

component - the AbstractComponent to use as this connection's right component

---

## getCategory

```
public Category getCategory()
```

Retrieves the most narrow category that this component is a member of. NOTE: This is the Category of the Connection class and has no relevance to the components it connects.

---

## toString

```
public String toString()
```

---

## getDocDescription

```
public String getDocDescription(AbstractComponent side,  
    boolean includeComponent)
```

Returns an HTML formatted documentation description from of the given side of this connection.

### Parameters:

side - the AbstractComponent on the side to describe from

### Returns:

a String containing the documentation description

---

## clone

```
public Object clone()
```

---

## disconnect

```
public void disconnect()
```

Disconnects this Connection between two AbstractComponent objects. Removes this connection from both components and from the model. Fires all appropriate events to notify the Navigator, views and other listeners. Handlers for user interactive disconnection should use #userDisconnect instead.

---

## userDisconnect

```
public void userDisconnect()
```

Disconnects the connection from the Navigator. This is the method called when a connection is disconnected from the Navigator or from one of this Connection's DrawnConnect renderers.

---



(continued from last page)

---

## reconnectIdentReferences

```
public void reconnectIdentReferences(boolean preserveUnresolved,  
    boolean useDbId)
```

---

## label

```
public String label()
```

---

## isVisual

```
public boolean isVisual()
```

Returns true if this connection should be visually represented. Visually represented means that this Connection will have DrawnConnection objects created for it if both sides of the Connection are present in a view.

---

## isEqualTo

```
public boolean isEqualTo(Connection con)
```

This determines if the Connection passed in is the equivalent of this Connection. Used instead of #equals to allow multiple equivalent connections between two components.

**Returns:**

true if both sides of each connection are the same.

---

## createDrawnComponent

```
public DrawnComponent createDrawnComponent()
```

Creates a DrawnConnection to render this Connection.

**Returns:**

a DrawnConnection for this Connection.

---

## getConnectionColor

```
public java.awt.Color getConnectionColor()
```

Returns the Color used to paint this Connection in a View.

---

## getConnectionStroke

```
public java.awt.Stroke getConnectionStroke()
```

Returns the Stroke used to paint this connection in a View.

---

## getCustomPopupItems

```
public Vector getCustomPopupItems()
```

Creates Custom Menu Items for any popup dialog involving this component

**Returns:**

a Vector of Action objects for this Connection.

## com.cafean.client.analysis

### Class ConnectionBean

```

java.lang.Object
├── com.cafean.client.analysis.GenericObject
│   ├── com.cafean.client.analysis.AbstractComponent
│   │   ├── com.cafean.client.analysis.Connection
│   │   │   └── com.cafean.client.analysis.ConnectionBean

```

public abstract class **ConnectionBean**  
 extends [Connection](#)

A class representing a connection between two bean components in an AbstractModel.

#### See Also:

[Connection](#)

**Fields inherited from class** [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

## Constructor Summary

|        |                                  |
|--------|----------------------------------|
| public | <a href="#">ConnectionBean()</a> |
|--------|----------------------------------|

## Method Summary

|         |   |
|---------|---|
| boolean | <a href="#">dumpBlockParams(java.io.PrintWriter dumpFile)</a><br>A stub method implemented here to allow ConnectionBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format.                 |
| Vector  | <a href="#">getCustomPopupItems()</a><br>Creates Custom Menu Items for any popup dialog involving this component.   |
| void    | <a href="#">popupDataDialog(java.awt.Window parent,boolean modal)</a>   |
| boolean | <a href="#">readBlockParams(com.appt.xdr.PibFile pibFile,int[] blockparm)</a><br>A stub method implemented here to allow ConnectionBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. |
| boolean | <a href="#">writeBlockParams(com.appt.xdr.PibFile pibFile)</a><br>A stub method implemented here to allow ConnectionBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format.                |

**Methods inherited from class** [com.cafean.client.analysis.Connection](#)

[clone](#), [createDrawnComponent](#), [disconnect](#), [getCategory](#), [getConnectionColor](#), [getConnectionStroke](#), [getCustomPopupItems](#), [getDocDescription](#), [getLeftComponent](#), [getLeftComponentID](#), [getLeftConnectData](#), [getOtherSide](#), [getOtherSideData](#), [getRightComponent](#), [getRightComponentID](#), [getRightConnectData](#), [getThisSideData](#), [isEqualTo](#), [isIndependentComponent](#), [isVisual](#), [label](#), [reconnectIdentReferences](#), [setLeftComponent](#), [setLeftComponentID](#), [setRightComponent](#), [setRightComponentID](#), [toString](#), [userDisconnect](#)

#### Methods inherited from class [com.cafean.client.analysis.AbstractComponent](#)

[addComponentListener](#), [addConnection](#), [addMessage](#), [addMessage](#), [addToModel](#), [addToModel](#), [canConnectTo](#), [clearConnections](#), [clone](#), [complete](#), [connectTo](#), [connectTo](#), [copy](#), [createDrawnComponent](#), [createSourceData](#), [createTargetData](#), [DBTypeCode](#), [disconnect](#), [disconnectFrom](#), [fireComponentChanged](#), [fireComponentChanged](#), [fireComponentConnected](#), [fireComponentDeleted](#), [fireComponentDisconnected](#), [getCatCCComparator](#), [getCategory](#), [getCCNumberComparator](#), [getComponent](#), [getComponentDependencies](#), [getConnectionCount](#), [getConnectionName](#), [getConnections](#), [getConnectionTypes](#), [getCustomPopupActions](#), [getCustomPopupItems](#), [getGroupedConnections](#), [getModel](#), [getName](#), [getNewCompIdent](#), [getOrder](#), [getOrderComparator](#), [getOwner](#), [getRealSize](#), [getSharedComponents](#), [includeInLoopcheck](#), [isOkayForExport](#), [isOkayForExport](#), [label](#), [popupDataDialog](#), [rebuildConnections](#), [reconnectImage](#), [removeComponentListener](#), [removeFromModel](#), [removeVerify](#), [restoreState](#), [setComponentNumber](#), [setDeleted](#), [setModel](#), [setOrder](#), [toString](#), [updateVersion](#), [writeName](#)

#### Methods inherited from class [com.cafean.client.analysis.GenericObject](#)

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#), [clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#), [deleteComment](#), [equals](#), [fixme](#), [getCCnumber](#), [getComment](#), [getComments](#), [getComments](#), [getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#), [getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#), [getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#), [popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#), [restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#), [setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setIdent](#), [setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#), [setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

#### Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### ConnectionBean

```
public ConnectionBean()
```

## Methods

(continued from last page)

## dumpBlockParams

```
public boolean dumpBlockParams(java.io.PrintWriter dumpFile)
```

A stub method implemented here to allow ConnectionBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## writeBlockParams

```
public boolean writeBlockParams(com.appt.xdr.PibFile pibFile)
```

A stub method implemented here to allow ConnectionBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## readBlockParams

```
public boolean readBlockParams(com.appt.xdr.PibFile pibFile,  
                                int[] blockparm)
```

A stub method implemented here to allow ConnectionBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## getCustomPopupItems

```
public Vector getCustomPopupItems()
```

Creates Custom Menu Items for any popup dialog involving this component. The resulting Vector should contain on JMenu, JMenuItem and JSeparator instances for this component.

**Returns:**

a Vector containing JMenu's, JMenuItem's, and JSeparators.

---

## popupDataDialog

```
public void popupDataDialog(java.awt.Window parent,  
                             boolean modal)
```

## com.cafean.client.analysis Class ConnectionData

java.lang.Object

└-com.cafean.client.analysis.ConnectionData

### All Implemented Interfaces:

Cloneable

### Direct Known Subclasses:

[SpecialConnectionData](#)

public abstract class **ConnectionData**

extends Object

implements Cloneable

A simple representation of one side of a Connection.

This object is used by DrawnComponent and DrawnConnection objects to find an appropriate ConnectingPt object based on one side of a Connection. Instances of derivatives of this class represent each side.

The {@link #equals} method is used by [DrawnComponent](#) and [DrawnConnection](#) objects to determine the appropriate {@link ConnectingPt} and must be implemented completely by each derivative class.

**NOTE:**All ConnectionData derivatives must be proper JavaBeans to be included as embedded connection points in [com.cafean.client.ui.DrawnViewComponents](#).

### See Also:

[com.cafean.client.ui.DrawnComponent.createConnectionPt\(\)](#), [com.cafean.client.ui.ConnectingPt.getConnectionData\(\)](#)

## Constructor Summary

|        |  |
|--------|--|
| public | ConnectionData()<br>Creates a new instance of ConnectionData   |
| public | ConnectionData(int index)<br>Creates a new instance of Connection data with a specific connection point index specified. |

## Method Summary

|         |  |
|---------|--|
| Object  | clone()<br>Retrieves this ConnectionData's ConnectingPt index.                                 |
| boolean | equals(Object obj)<br>Determines the equivalence of this ConnectionData with the given object. |
| int     | getConnectionIndex()<br>Retrieves this ConnectionData's ConnectingPt index.                    |
| void    | setConnectionIndex(int connectionIndex)<br>Sets this ConnectionData's ConnectingPt index.      |

|        |            |
|--------|------------|
| String | toString() |
|--------|------------|

---

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

---

## Constructors

### ConnectionData

```
public ConnectionData()
```

Creates a new instance of `ConnectionData`

---

### ConnectionData

```
public ConnectionData(int index)
```

Creates a new instance of `Connection data` with a specific connection point index specified.

**Parameters:**

`index` - the connection point index of this connection.

---

## Methods

### equals

```
public boolean equals(Object obj)
```

Determines the equivalence of this `ConnectionData` with the given object. Derivative classes must override this method and must include a comparison of `connectionIndex` as the default implementation returns reference equality.

**See Also:**

`Object.equals()`

---

### clone

```
public Object clone()
```

---

### getConnectionIndex

```
public int getConnectionIndex()
```

Retrieves this `ConnectionData`'s `ConnectingPt` index. **WARNING:** This value has a a `Connection` type specific meaning.

---

### setConnectionIndex

```
public void setConnectionIndex(int connectionIndex)
```

Sets this `ConnectionData`'s `ConnectingPt` index. **WARNING:** This value has a a `Connection` type specific meaning.

---

### toString

```
public String toString()
```

---

## com.cafean.client.analysis

### Class ConnectionList

java.lang.Object

└─com.cafean.client.analysis.ConnectionList

#### All Implemented Interfaces:

Cloneable

public class **ConnectionList**  
 extends Object  
 implements Cloneable

A list of connections to be used inside an AbstractComponent.

### Constructor Summary

|        |   |
|--------|---|
| public | ConnectionList(AbstractComponent component)<br>Creates a new instance of ConnectionList   |
| public | ConnectionList(AbstractComponent component, ConnectionList list)<br>Creates a new instance of connectionList, copying an existing list. |

### Method Summary

|                              |  |
|------------------------------|--|
| void                         | addConnection(Connection connection)<br>Adds the given component to this ConnectionList.   |
| void                         | clear()<br>Removes all entries in this list.   |
| Object                       | clone()  |
| boolean                      | contains(Connection con)<br>Returns true if the given component is referred to by this subsystem.  |
| int                          | getConnectionCount()<br>Returns the count of the component references in this Subsystem.   |
| <a href="#">Connection[]</a> | getConnections()<br>Retrieves an array of the components referred to by this subsystem.  |
| Iterator                     | getIterator()<br>Retrieves an Iterator suitable for traversing this entire list.   |
| void                         | reconnectIdentReferences(AbstractModel model, boolean preserveUnresolved, boolean useDbId)<br>Resets this component's internal ident references to refer to appropriate components in the current AbstractModel. |

|      |   |
|------|---|
| void | <b>removeConnection</b> ( <a href="#">Connection</a> con)<br>Removes the given component from this subsystem. |
|------|---|

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### ConnectionList

**public** **ConnectionList**([AbstractComponent](#) component)  
Creates a new instance of ConnectionList

### ConnectionList

**public** **ConnectionList**([AbstractComponent](#) component,  
[ConnectionList](#) list)  
Creates a new instance of connectionList, copying an existing list.

## Methods

### addConnection

**public void** **addConnection**([Connection](#) connection)  
Adds the given component to this ConnectionList.

**Parameters:**

`connection` - the Connection to add to this ConnectionList.

### clear

**public void** **clear**()  
Removes all entries in this list.

### clone

**public Object** **clone**()

### getConnections

**public** [Connection\[\]](#) **getConnections**()  
Retrieves an array of the components referred to by this subsystem.

**Returns:**

an [AbstractComponent\[\]](#) containing the components referred to



(continued from last page)

---

## reconnectIdentReferences

```
public void reconnectIdentReferences(AbstractModel model,  
    boolean preserveUnresolved,  
    boolean useDbId)
```

Resets this component's internal ident references to refer to appropriate components in the current AbstractModel. Intended for use after adding a component to a model and thus it's DB\_ID will have been set to it's ident in the previous model. For ident references use find\_\_ByDbId to find the new component, then store it's ident. Note: If this component's DB\_ID is 0 this method does nothing.

### Parameters:

model - the AbstractModel to reconnectIdents inside of.

preserveUnresolved - if true, ident references that aren't resolvable will be left dangling, if false, they will be set to 0.

useDbId - if true, ident references will be reconnected via find\_x\_ByDB\_ID; if false, ident references will be reconnected via find\_x\_ByIdent

---

## contains

```
public boolean contains(Connection con)
```

Returns true if the given component is referred to by this subsystem.

---

## removeConnection

```
public void removeConnection(Connection con)
```

Removes the given component from this subsystem.

---

## getConnectionCount

```
public int getConnectionCount()
```

Returns the count of the component references in this Subsystem.

---

## getIterator

```
public Iterator getIterator()
```

Retrieves an Iterator suitable for traversing this entire list.

### Returns:

an Iterator suitable for iterating over the entirety of this list

## com.cafean.client.analysis Class ElementBean

```

java.lang.Object
  |
  +--com.cafean.client.analysis.GenericObject
        |
        +--com.cafean.client.analysis.ElementBean
  
```

public abstract class **ElementBean**  
extends [GenericObject](#)

The base class for ModelEditor Elements that are full fledged beans.

### See Also:

[AbstractModel#addElement](#)

Fields inherited from class [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

## Constructor Summary

|        |  |
|--------|--|
| public | <code>ElementBean()</code><br>Creates a new ElementBean with a DATA_INCOMPLETE data state and named unnamed                    |
| public | <code>ElementBean(int componentNumber)</code><br>Creates a new ElementBean with a DATA_INCOMPLETE data state and named unnamed |

## Method Summary

|         |   |
|---------|---|
| boolean | <code>dumpBlockParams(java.io.PrintWriter dumpFile)</code><br>A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format.                 |
| boolean | <code>readBlock(com.appt.xdr.PibFile pibFile)</code><br>A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format.                       |
| boolean | <code>readBlockParams(com.appt.xdr.PibFile pibFile)</code><br>A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format.                 |
| boolean | <code>readBlockParams(com.appt.xdr.PibFile pibFile,int[] blockparm)</code><br>A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. |
| void    | <code>restoreState(String prefix,Hashtable state)</code><br>Restore the state of the bean from an earlier edit.   |

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#),  
[clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#),  
[deleteComment](#), [equals](#), [fixme](#), [getCCNumber](#), [getComment](#), [getComments](#), [getComments](#),  
[getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#),  
[getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#),  
[getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#),  
[popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#),  
[rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#),  
[restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#),  
[setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setId](#),  
[setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#),  
[setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#),  
[writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#),  
[writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#),  
[writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

```
public ElementBean()
```

```
public ElementBean()
```

Creates a new `ElementBean` with a `DATA INCOMPLETE` data state and named `unnamed`

public **Element**

```
public ElementBean(int componentNumber)
```

Creates a new `ElementBean` with a `DATA INCOMPLETE` data state and named `unnamed`

```
public boolean dumpB
```

```
public boolean dumpBlockParams(java.io.PrintWriter dumpFile)
```

A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

```
public bool
```

```
public boolean readBlock(com.apt.xdr.PibFile pibFile)
```

A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## writeBlockParams

```
public boolean writeBlockParams(com.appt.xdr.PibFile pibFile)
```

A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## readBlockParams

```
public boolean readBlockParams(com.appt.xdr.PibFile pibFile)
```

A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## readBlockParams

```
public boolean readBlockParams(com.appt.xdr.PibFile pibFile,  
    int[] blockparm)
```

A stub method implemented here to allow ElementBean objects to be PibBlocks, stored and loaded directly to a PIB generated file format. This method simply returns true.

---

## storeState

```
public void storeState(String prefix,  
    Hashtable state)
```

Store the state of the bean to permit undo.

### Parameters:

- state - a Hashtable containing modified parameters.
  - prefix - a String containing the prefix for hash entries.
- 

## restoreState

```
public void restoreState(String prefix,  
    Hashtable state)
```

Restore the state of the bean from an earlier edit.

### Parameters:

- state - a Hashtable containing modified parameters.
  - prefix - a String containing the prefix for hash entries.
-

## com.cafean.client.analysis Class ElementList

java.lang.Object

└--com.cafean.client.analysis.ElementList

### All Implemented Interfaces:

Cloneable

### Direct Known Subclasses:

[ComponentList](#)

public class **ElementList**

extends Object

implements Cloneable

A storage class for GenericObjects.

ElementList requires that all it's contained elements have unique ident numbers.

Elements are optionally stored and retrieved in sorted order via binary searches for efficiency.

## Constructor Summary

|        |  |
|--------|--|
| public | ElementList()<br>Creates a new instance of ElementList with a category of Components that starts in a sorted state.                                    |
| public | ElementList(boolean sorted)<br>Creates a new instance of ElementList with a category of Components.  |
| public | ElementList(String category, boolean sorted)<br>Creates a new instance of ElementList with the given category name and the given initial sorted state. |

## Method Summary

|        |   |
|--------|---|
| void   | add(GenericObject object)<br>Adds the given object to this list.                              |
| void   | clear()<br>Removes all elements from this list.   |
| void   | clearDbIds()<br>Resets the DB_ID's of all contained elements.                                 |
| Object | clone()<br>{ @inheritDoc } Note: Only the contained references to GenericObject's are copied. |

|                               |   |
|-------------------------------|---|
| boolean                       | contains( <code>GenericObject element</code> )<br>Returns true if this list contains an object with the same ident as the given object.   |
| <a href="#">GenericObject</a> | findByDB_ID( <code>int dbid</code> )<br>Retrieves the <code>GenericObject</code> with the given <code>DB_ID</code> .  |
| <a href="#">GenericObject</a> | findById( <code>int identNumber</code> )<br>Retrieves the <code>GenericObject</code> with the given ident.  |
| <a href="#">GenericObject</a> | findById( <code>int identNumber,boolean includeDeleted</code> )<br>Retrieves the <code>GenericObject</code> with the given ident.   |
| <a href="#">GenericObject</a> | get( <code>int index</code> )<br>Retrieves the element at the given index.  |
| String                        | getCategory()<br>Retrieves the name of the category of the elements contained by <code>thisElementList</code>   |
| int                           | indexOf( <code>GenericObject element</code> )<br>Returns the index of the given element in this list or -1.   |
| boolean                       | isSorted()<br>Returns true if this list has been sorted and will continue to add objects in a sorted order.   |
| Iterator                      | iterator()<br>Retutrnrs an <code>Iterator</code> object for iterating through this list.  |
| void                          | reconnectIdentReferences( <code>boolean preserveUnresolved,boolean useDbId</code> )<br>Reconnects ident references for this list's contained elements.  |
| boolean                       | remove( <code>GenericObject element</code> )<br>Removes the given object from this list.  |
| void                          | removeDeleted()<br>Removes all elements contained in the list that are marked deleted as indicated by <code>GenericObject.isDeleted()</code>  |
| void                          | setCategory( <code>String category</code> )<br>Sets the name of the category of the elements contained by <code>thisElementList</code>  |
| int                           | size()<br>Retrieves the size of this <code>ElementList</code> .   |
| void                          | sort()<br>Sorts this list and indicates that any objects added in the future should be added into thier sorted order.   |
| <code>Object[]</code>         | toArray()<br>Returns an array containing all of the elements in this list in the correct order.   |
| <code>Object[]</code>         | toArray( <code>Object[] a</code> )<br>Returns an array containing all of the elements in this list in the correct order; the runtime type of the returned array is that of the specified array. |

|        |  |
|--------|--|
| String | toString()<br>Returns a String representation of this ElementList. |
|--------|--|

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### ElementList

```
public ElementList()
```

Creates a new instance of ElementList with a category of Components that starts in a sorted state.

**See Also:**

```
.sort()
```

### ElementList

```
public ElementList(boolean sorted)
```

Creates a new instance of ElementList with a category of Components.

**Parameters:**

`sorted` - if true all additions will be inserted in thier sorted positions.

**See Also:**

```
.sort()
```

### ElementList

```
public ElementList(String category,
                   boolean sorted)
```

Creates a new instance of ElementList with the given category name and the given initial sorted state.

**Parameters:**

`category` - a String containing the category name to use for this element list.

`sorted` - if true all additions will be inserted in thier sorted positions.

**See Also:**

```
.sort()
```

## Methods

### findById

```
public GenericObject findById(int identNumber)
```

Retrieves the GenericObject with the given ident. If identNumber is 0, null is returned.

**Parameters:**

`identNumber` - the ident of the required object.

**Returns:**

(continued from last page)

a GenericObject object corresponding to the search parameters

---

## findById

```
public GenericObject findById(int identNumber,  
    boolean includeDeleted)
```

Retrieves the GenericObject with the given ident. If identNumber is 0, null is returned.

### Parameters:

identNumber - the ident of the required object.  
includeDeleted - if true; objects set deleted will still be returned.

### Returns:

a GenericObject object corresponding to the search parameters

---

## findByDB\_ID

```
public GenericObject findByDB_ID(int dbid)
```

Retrieves the GenericObject with the given DB\_ID. if the DB\_ID is 0, null is returned.

### Parameters:

dbid - the DB\_ID of the desired object.

### Returns:

a GenericObject object corresponding to the search parameters

---

## reconnectIdentReferences

```
public void reconnectIdentReferences(boolean preserveUnresolved,  
    boolean useDbId)
```

Reconnects ident references for this list's contained elements.

### Parameters:

preserveUnresolved - if true, ident references that aren't resolvable will be left dangling, if false, they will be set to 0.  
useDbId - if true, ident references will be reconnected via find\_x\_ByDB\_ID; if false, ident references will be reconnected via find\_x\_ByIdent

### See Also:

[com.cafean.client.analysis.GenericObject.clearDbIds\(\)](#)

---

## clearDbIds

```
public void clearDbIds()
```

Resets the DB\_ID's of all contained elements.

### See Also:

[com.cafean.client.analysis.GenericObject.clearDbIds\(\)](#)

---

## removeDeleted

```
public void removeDeleted()
```

Removes all elements contained in the list that are marked deleted as indicated by `GenericObject.isDeleted()`

---

## get

```
public GenericObject get(int index)
```



(continued from last page)

Retrieves the element at the given index.

**Parameters:**

`index` - the index of the desired element

**Returns:**

the `AbstractComponent` of the desired element

---

**size**

```
public int size()
```

Retrieves the size of this `ElementList`.

---

**add**

```
public void add(GenericObject object)
```

Adds the given object to this list. If `isSorted` returns true the object will be added into sorted order and adding duplicates will cause an `IllegalArgumentException` to be thrown.

**Parameters:**

`object` - the `GenericObject` to add to this list.

**Throws:**

`IllegalArgumentException` - if the given object is already in this list.

---

**remove**

```
public boolean remove(GenericObject element)
```

Removes the given object from this list.

**Parameters:**

`element` - the `GenericObject` to remove

**Returns:**

true if the given element was actually removed.

---

**indexOf**

```
public int indexOf(GenericObject element)
```

Returns the index of the given element in this list or -1.

**Parameters:**

`element` - the `GenericObject` to find the index of

---

**contains**

```
public boolean contains(GenericObject element)
```

Returns true if this list contains an object with the same ident as the given object.

---

**clear**

```
public void clear()
```

Removes all elements from this list.

---

(continued from last page)

---

## isSorted

```
public boolean isSorted()
```

Returns true if this list has been sorted and will continue to add objects in a sorted order.

---

## sort

```
public void sort()
```

Sorts this list and indicates that any objects added in the future should be added into thier sorted order.

---

## toArray

```
public Object[] toArray()
```

Returns an array containing all of the elements in this list in the correct order.

**Returns:**

an array containing all of the elements in this list in the correct order.

---

## toArray

```
public Object[] toArray(Object[] a)
```

Returns an array containing all of the elements in this list in the correct order; the runtime type of the returned array is that of the specified array. If the list fits in the specified array, it is returned therein. Otherwise, a new array is allocated with the runtime type of the specified array and the size of this list. If the list fits in the specified array with room to spare (i.e., the array has more elements than the list), the element in the array immediately following the end of the collection is set to null. This is useful in determining the length of the list *only* if the caller knows that the list does not contain any nullelements.

**Parameters:**

a – the array into which the elements of the list are to be stored, if it is big enough; otherwise, a new array of the same runtime type is allocated for this purpose.

**Returns:**

an array containing the elements of the list.

**Throws:**

`ArrayStoreException` – if the runtime type of a is not a supertype of the runtime type of every element in this list.

---

## iterator

```
public Iterator iterator()
```

Retutrnns an Iterator object for iterating through this list.

---

## getCategory

```
public String getCategory()
```

Retrieves the name of the category of the elements contained by thisElementList

**Returns:**

a String containing the category name of this ElementList

---

## setCategory

```
public void setCategory(String category)
```

Sets the name of the category of the elements contained by thisElementList

---

(continued from last page)

**Parameters:**

`category` - a String containing the name of this ElementList

---

**toString**

```
public String toString()
```

Returns a String representation of this ElementList.

---

**clone**

```
public Object clone()
```

Note: Only the contained references to GenericObject's are copied. The GenericObject's themselves are not cloned.

## com.cafean.client.analysis Class GenericObject

java.lang.Object

└-com.cafean.client.analysis.GenericObject

### All Implemented Interfaces:

[IdentHolder](#), [StateEditable](#), [Cloneable](#)

### Direct Known Subclasses:

[ElementBean](#), [AbstractModel](#), [AbstractComponent](#)

public abstract class **GenericObject**

extends Object

implements [Cloneable](#), [StateEditable](#), [IdentHolder](#)

A generic model object with a name, description data state etc. This class is the base for all objects are contained in an ElementList.

## Field Summary

|                  |   |
|------------------|---|
| static final int | DATA_COMPLETE<br>Indication that this object is complete and error free.<br>Value: <b>2147483647</b>          |
| static final int | DATA_ERROR<br>Indication that this object has a fatal error that requires attention.<br>Value: <b>2</b>       |
| static final int | DATA_INCOMPLETE<br>Indication that this object is missing required data.<br>Value: <b>0</b>                   |
| static final int | DATA_WARNING<br>Indication that this object has a non-fatal error that requires attention.<br>Value: <b>1</b> |

## Constructor Summary

|        |   |
|--------|---|
| public | GenericObject()<br>Creates a new GenericObject with a DATA_INCOMPLETE data state and named unnamed                    |
| public | GenericObject(int componentNumber)<br>Creates a new GenericObject with a DATA_INCOMPLETE data state and named unnamed |

## Method Summary

|             |  |
|-------------|--|
| void        | addComment(String comment)<br>Appends the given string as to the list of comments in this object   |
| void        | addMultipleComments(Vector comments)<br>Add the given Vector of Comments to this object.   |
| boolean     | checkRealArrayList(ArrayList table,String desc)<br>Performs a check of the given ArrayList<Real> by iterating through each row and checking for Unknown values.    |
| boolean     | checkRealArrayTable(ArrayList table,String desc)<br>Performs a check of the given ArrayList<Real[]> by iterating through each row and checking for Unknown values. |
| void        | clearDbIds()<br>Resets all the DB_ID's associated with this base object to 0   |
| Object      | clone()<br>Creates a deep copy of this object.   |
| void        | closeAllViews()<br>Closes all open views of this object.   |
| int         | compareTo(Object o)<br>Compares this GenericObject with the given object based on ident for a natural ordering.  |
| void        | copyFrom(GenericObject o)<br>Copy the attributes from a source object to this instance.  |
| void        | createDataPages(GenericObject original)<br>Creates data pages for this object's Component View based on the given original object.                                 |
| static void | debug(String message)  |
| void        | deleteAllComments()<br>Removes all Comments from this object.  |
| void        | deleteComment(int num)<br>Removes the given Comment from this object's comment list.   |
| boolean     | equals(Object obj)<br>Returns true if the given object is this object, determined by reference equivalence.  |
| static void | fixme(String message)<br>Prints the given message to stderr in FIXME format.   |
| String      | getCCnumber()<br>Retrieves this objects's CC number.   |
| String      | getComment(int num)<br>Retrieves the comment at the given index.   |
| String[]    | getComments()<br>Retrieves all the comments inside this object as Strings  |

|               |  |
|---------------|--|
| String        | getComments(int index)<br>Retrieves the comment at the given index.  |
| static String | getComponentCCNumber(GenericObject component)<br>Returns the CC number of the given component or "0" if null.  |
| int           | getComponentNumber()<br>Retrieves this object's component number.  |
| int           | getDataState()<br>Retrieves this object's data state.  |
| int           | getDB_ID()<br>Retrieves this object's DB_ID  |
| String        | getDescription()<br>Retrieves the description of this object.  |
| int           | getIdent()<br>Retrieves the ident of this object, NOT the DB_ID, and not the CC/Code Number.   |
| int           | getMajorCreationVersion()<br>Retrieves the major revision number that this object was created with.  |
| int           | getMajorVersion()<br>Retrieves this objects current major revision number.   |
| int           | getMinorCreationVersion()<br>Retrieves ths minor revision number that this object was created with.  |
| int           | getMinorVersion()<br>Retrieves this object's current minor revision number.  |
| String        | getName()<br>Retrieves this object's name  |
| static int    | getNewCompIdent(AbstractModel model,int dbid,boolean preserveUnresolved,boolean useDbId)<br>Retrieves the ident of the component with the given DB_ID in this component's model. |
| int           | getNumComments()<br>Retrieves the count of Comment objects stored in this object.  |
| boolean       | isDeleted()<br>Returns true if this object has been set deleted.   |
| void          | popupDataDialog()<br>Creates a new ComponentView for this object or resets and refreshes this object's current ComponentView.  |
| void          | popupDataDialog(java.awt.Window parent,boolean modal)<br>Creates a new ComponentView for this object or resets and refreshes this object's current ComponentView.                |

|         |   |
|---------|---|
| boolean | <code>rangeCheck(double i,double min,double max,String msg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .                      |
| boolean | <code>rangeCheck(double i,double min,double max,String msg,boolean printMsg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .     |
| boolean | <code>rangeCheck(float i,float min,float max,String msg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .                         |
| boolean | <code>rangeCheck(float i,float min,float max,String msg,boolean printMsg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .        |
| boolean | <code>rangeCheck(int i,int min,int max,String msg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .                               |
| boolean | <code>rangeCheck(Int i,int min,int max,String msg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .                               |
| boolean | <code>rangeCheck(int i,int min,int max,String msg,boolean printMsg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .              |
| boolean | <code>rangeCheck(Int i,int min,int max,String msg,boolean printMsg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .              |
| boolean | <code>rangeCheck(Real i,double min,double max,String msg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .                        |
| boolean | <code>rangeCheck(Real r,double min,double max,String msg,boolean printMsg)</code><br>Checks whether the given value <code>i</code> is within the range specified by the <code>min</code> and <code>max</code> .       |
| void    | <code>reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)</code><br>Resets this component's internal ident references to refer to appropriate components in the current <code>AbstractModel</code> . |
| void    | <code>restoreState(Hashtable state)</code><br>Restore the state of the bean from an earlier edit by using <code>copyFrom</code> on the previously stored clone.   |
| void    | <code>restoreState(String prefix,Hashtable state)</code><br>Restore the state of the bean from an earlier edit.   |
| void    | <code>setComments(int index,String value)</code><br>Sets the comment at the given index.  |
| void    | <code>setComments(String[] comments)</code><br>Sets all of the comments inside this object as Strings   |
| void    | <code>setComponentNumber(int i)</code><br>Setter for the component number.  |
| void    | <code>setCreationVersion(int major,int minor)</code><br>Sets this object's creation version to the given numbers.   |

|             |   |
|-------------|---|
| void        | setDataState(int ds)<br>Sets this object's data state.  |
| void        | setDB_ID(int dbid)<br>Sets this object's DB_ID  |
| void        | setDeleted(boolean del)<br>Sets this object deletedand thus inoperative.  |
| void        | setDescription(String desc)<br>Sets this object's description to the given desc.  |
| void        | setIdent(int uid)<br>Sets the unique ident of this object.  |
| void        | setMajorCreationVersion(int version)<br>Sets this object's creation version to the given version.   |
| void        | setMajorVersion(int major_tag)<br>Sets this objects current major revision number.  |
| void        | setMinorCreationVersion(int version)<br>Sets this object's creation version to the given version.   |
| void        | setMinorVersion(int minor_tag)<br>Sets this objects current minor revision number.  |
| void        | setName(String name)<br>Sets this object's name to the given name after trimming it.  |
| String      | showComment(int num)<br>Retrieves the text in the comment at the given index.   |
| void        | storeState(Hashtable state)<br>Stores the state of the object to permit undo by cloning itself and storing the clone.   |
| void        | storeState(String prefix,Hashtable state)<br>Store the state of the bean to permit undo.  |
| static void | trace(String message)<br>Prints the given message to stderr in TRACE format.  |
| void        | updateVersion()<br>Updates this object's current version to be greater than or equal to it's model's version.   |
| void        | validate()<br>Determines this object's general data state by examining it's internal data.  |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,Dimless[] values)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array. |



|             |   |
|-------------|---|
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,Dimless[] values,int columns)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                   |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,int[] values)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                                   |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,int[] values,int columns)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                       |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,Object[] values)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                                |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,Object[] values,int columns)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                    |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,Real[] values)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                                  |
| static void | writeArrayLoadValue(java.io.PrintWriter out,String header,Real[] values,int columns)<br>writes out a value in TRACE 'LOAD Format' for each element in the given array.                      |
| static void | writeMuxLoadArray(java.io.PrintWriter out,String header,Object[] xValues,Object[] yValues)<br>writes the given demultiplexed x,y pairs to the given writer in multiplexed TRAC LOAD format. |
| static void | writeMuxLoadArray(java.io.PrintWriter out,String header,Real[] xValues,Real[] yValues)<br>writes the given demultiplexed x,y pairs to the given writer in multiplexed TRAC LOAD format.     |
| static void | writeSP(java.io.PrintWriter out,int value,int width)<br>Write an integer to a PrintWriter as a fixed width string, padding the left with spaces.  |
| static void | writeSP(java.io.PrintWriter out,String text,int width)<br>Write a fixed width string to a PrintWriter, padding the left with spaces.  |

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### DATA\_INCOMPLETE

public static final int **DATA\_INCOMPLETE**

Indication that this object is missing required data.

---

## DATA\_WARNING

```
public static final int DATA_WARNING
```

Indication that this object has a non-fatal error that requires attention.

---

## DATA\_ERROR

```
public static final int DATA_ERROR
```

Indication that this object has a fatal error that requires attention.

---

## DATA\_COMPLETE

```
public static final int DATA_COMPLETE
```

Indication that this object is complete and error free.

---

## Constructors

### GenericObject

```
public GenericObject()
```

Creates a new GenericObject with a DATA\_INCOMPLETE data state and named unnamed

---

### GenericObject

```
public GenericObject(int componentNumber)
```

Creates a new GenericObject with a DATA\_INCOMPLETE data state and named unnamed

---

## Methods

### restoreState

```
public void restoreState(String prefix,  
    Hashtable state)
```

Restore the state of the bean from an earlier edit.

**Parameters:**

state - a Hashtable containing modified parameters.  
prefix - a String containing the prefix for hash entries.

---

### restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit by using copyFrom on the previously stored clone.

**Parameters:**

state - A hash table containing modified parameters.

---

### storeState

```
public void storeState(String prefix,  
    Hashtable state)
```

Store the state of the bean to permit undo.

---

(continued from last page)

**Parameters:**

- state - a Hashtable containing modified parameters.
  - prefix - a String containing the prefix for hash entries.
- 

## storeState

```
public void storeState(Hashtable state)
```

Stores the state of the object to permit undo by cloning itself and storing the clone. NOTE: If the component storing its state needs a deep copy that its clone() method does not provide, it must override storeState to find that functionality elsewhere.

**Parameters:**

- state - A hash table containing modified parameters.
- 

## clone

```
public Object clone()
```

Creates a deep copy of this object.

---

## copyFrom

```
public void copyFrom(GenericObject o)
```

Copy the attributes from a source object to this instance. This is used for copying data from an edited working copy into the original object. Most notably, it is used to enable undo/redo for the legacy beanless ModelEditor architecture as well as for importing new altered data from restart decks for some plugins. This should call copyFrom on children that support it. **Note: Never call copyFrom from clone or copy!**

**Parameters:**

- o - the GenericObject source object.
- 

## setName

```
public void setName(String name)
```

Sets this object's name to the given name after trimming it.

**Parameters:**

- name - a String containing the desired name.
- 

## getName

```
public String getName()
```

Retrieves this object's name

**Returns:**

- a String containing this GenericObject's name.
- 

## addComment

```
public void addComment(String comment)
```

Appends the given string as to the list of comments in this object

---

## addMultipleComments

```
public void addMultipleComments(Vector comments)
```

Add the given Vector of Comments to this object.

---

(continued from last page)

**Parameters:**`comments` - the Vector of Comment objects to add.

---

**setDescription**

```
public void setDescription(String desc)
```

Sets this object's description to the given desc.

**Parameters:**`desc` - a String containing this object's new description.

---

**getDescription**

```
public String getDescription()
```

Retrieves the description of this object.

**Returns:**

a String containing this object's description.

---

**getNumComments**

```
public int getNumComments()
```

Retrieves the count of Comment objects stored in this object.

---

**deleteComment**

```
public void deleteComment(int num)
```

Removes the given Comment from this object's comment list.

---

**deleteAllComments**

```
public void deleteAllComments()
```

Removes all Comments from this object.

---

**showComment**

```
public String showComment(int num)
```

Retrieves the text in the comment at the given index.

---

**getComment**

```
public String getComment(int num)
```

Retrieves the comment at the given index.

---

**getComments**

```
public String[] getComments()
```

Retrieves all the comments inside this object as Strings

---

**getComments**

```
public String getComments(int index)
```

Retrieves the comment at the given index.

---

## setComments

```
public void setComments(int index,  
                        String value)
```

Sets the comment at the given index.

---

## setComments

```
public void setComments(String[] comments)
```

Sets all of the comments inside this object as Strings

---

## getIdent

```
public int getIdent()
```

Retrieves the ident of this object, NOT the DB\_ID, and not the CC/Code Number. For internal use only. Do not show this to users. Set by the model when added.

**Returns:**

the unique id of this object.

---

## setId

```
public void setId(int uid)
```

Sets the unique ident of this object. This should only be used by the AbstractModel or related structures when adding this object to a model.

---

## isDeleted

```
public boolean isDeleted()
```

Returns true if this object has been set deleted.

**See Also:**

`.setDeleted()`

---

## setDeleted

```
public void setDeleted(boolean del)
```

Sets this object deleted and thus inoperative. If this object has any children, it will propagate this deleted status to it's children via this method.

---

## reconnectIdentReferences

```
public void reconnectIdentReferences(boolean preserveUnresolved,  
                                     boolean useDbId)
```

Resets this component's internal ident references to refer to appropriate components in the current AbstractModel. Intended for use after adding a component to a model and thus it's DB\_ID will have been set to it's ident in the previous model. For ident references use find\_\_ByDbId to find the new component, then store it's ident. Note: If this component's DB\_ID is 0 this method does nothing.

**Parameters:**

`preserveUnresolved` - if true, ident references that aren't resolvable will be left dangling, if false, they will be set to 0.

`useDbId` - if true, ident references will be reconnected via find\_x\_ByDB\_ID; if false, ident references will be reconnected via find\_x\_ByIdent

---

(continued from last page)

---

## clearDbIds

```
public void clearDbIds()
```

Resets all the DB\_ID's associated with this base object to 0

---

## getDB\_ID

```
public int getDB_ID()
```

Retrieves this object's DB\_ID

**Returns:**

the DB\_ID

---

## setDB\_ID

```
public void setDB_ID(int dbid)
```

Sets this object's DB\_ID

**Parameters:**

dbid - the new DB\_ID

---

## compareTo

```
public int compareTo(Object o)
```

Compares this GenericObject with the given object based on ident for a natural ordering.

---

## validate

```
public void validate()
```

Determines this object's general data state by examining it's internal data. The data state is used to color code objects so the user can see the current state. The `validate` method will set the state attribute to one of these values. Note that this state is not inclusive of the checks in [isOkayForExport](#) and is used only to validate data required for the visual representation. Derivatives should call `super.validate()` before determining thier own data state. This implementation sets this object's data state to DATA\_COMPLETE without any analysis.

---

## getDataState

```
public int getDataState()
```

Retrieves this object's data state.

**Returns:**

a DATA\_\* enumerated value such as DATA\_COMPLETE or DATA\_INCOMPLETE.

**See Also:**

`.validate()`

---

## setDataState

```
public void setDataState(int ds)
```

Sets this object's data state.

**See Also:**

`.getDataState()`  
`.validate()`

---

(continued from last page)

---

## getCCNumber

```
public String getCCNumber()
```

Retrieves this objects's CC number. The CC number is the object's display number in String form.

---

## getComponentNumber

```
public int getComponentNumber()
```

Retrieves this object's component number.

---

## getComponentCCNumber

```
public static String getComponentCCNumber(GenericObject component)
```

Returns the CC number of the given component or "0" if null.

---

## setComponentNumber

```
public void setComponentNumber(int i)
```

Setter for the component number. Also known as the display number

### Parameters:

i - the number to set this component number to

---

## closeAllViews

```
public void closeAllViews()
```

Closes all open views of this object.

---

## popupDataDialog

```
public void popupDataDialog()
```

Creates a new ComponentView for this object or resets and refreshes this object's current ComponentView.

---

## popupDataDialog

```
public void popupDataDialog(java.awt.Window parent,  
                             boolean modal)
```

Creates a new ComponentView for this object or resets and refreshes this object's current ComponentView.

---

## createDataPages

```
public void createDataPages(GenericObject original)
```

Creates data pages for this object's Component View based on the given original object.

---

## equals

```
public boolean equals(Object obj)
```

Returns true if the given object is this object, determined by reference equivalence. Due to the current undo system, this method cannot be overridden to do a more useful comparison.

### Parameters:

obj - {@inheritDoc}

### See Also:

`Object.equals()`

---

---

## updateVersion

```
public void updateVersion()
```

Updates this object's current version to be greater than or equal to it's model's version. If the major versions are the same, this object's minor version is incremented. NOTE: This implementation does nothing. Derivative classes must implement.

---

## setCreationVersion

```
public void setCreationVersion(int major,  
                                int minor)
```

Sets this object's creation version to the given numbers. NOTE: Should only be used on load.

### Parameters:

major - the new major creation version number  
minor - the new minor creation version number

---

## setMajorCreationVersion

```
public void setMajorCreationVersion(int version)
```

Sets this object's creation version to the given version. NOTE: Should only be used on load.

### Parameters:

version - the new creation version number

---

## setMinorCreationVersion

```
public void setMinorCreationVersion(int version)
```

Sets this object's creation version to the given version. NOTE: Should only be used on load.

### Parameters:

version - the new creation version number

---

## setMajorVersion

```
public void setMajorVersion(int major_tag)
```

Sets this objects current major revision number.

---

## setMinorVersion

```
public void setMinorVersion(int minor_tag)
```

Sets this objects current minor revision number.

---

## getMajorVersion

```
public int getMajorVersion()
```

Retrieves this objects current major revision number.

---

## getMinorVersion

```
public int getMinorVersion()
```

Retrieves this object's current minor revision number.

---



(continued from last page)

---

## getMajorCreationVersion

```
public int getMajorCreationVersion()
```

Retrieves the major revision number that this object was created with.

---

## getMinorCreationVersion

```
public int getMinorCreationVersion()
```

Retrieves this minor revision number that this object was created with.

---

## checkRealArrayTable

```
public boolean checkRealArrayTable(ArrayList table,  
    String desc)
```

Performs a check of the given ArrayList<Real[]> by iterating through each row and checking for Unknown values.

**Parameters:**

table - the ArrayList to check

desc - a String containing a description usable in the error message window

---

## checkRealArrayList

```
public boolean checkRealArrayList(ArrayList table,  
    String desc)
```

Performs a check of the given ArrayList<Real> by iterating through each row and checking for Unknown values.

**Parameters:**

table - the ArrayList to check

desc - a String containing a description usable in the error message window

**Returns:**

false if the ArrayList contains unknown values, false otherwise.

---

## fixme

```
public static void fixme(String message)
```

Prints the given message to stderr in FIXME format. The fixme format is specified as: `FIXME[ 'class name'. 'method name'<'line number'>]: 'message'`

---

## debug

```
public static void debug(String message)
```

---

## trace

```
public static void trace(String message)
```

Prints the given message to stderr in TRACE format. The TRACE format is specified as: `TRACE[ 'class name'. 'method name'<'line number'>]: 'message'`

---

## getNewCompIdent

```
public static int getNewCompIdent(AbstractModel model,  
    int dbid,  
    boolean preserveUnresolved,  
    boolean useDbId)
```

---

(continued from last page)

Retrieves the ident of the component with the given DB\_ID in this component's model. If dbid is 0, a 0 is returned.

**Parameters:**

`preserveUnresolved` - if true, and no component is found `dbid` will be returned. if false, 0 will be returned.  
`useDbId` - if true, ident references will be reconnected via `find_x_ByDB_ID`; if false, ident references will be reconnected via `find_x_ByIdent`

---

## rangeCheck

```
public boolean rangeCheck(int i,  
    int min,  
    int max,  
    String msg)
```

Checks whether the given value `i` is within the range specified by the `min` and `max`.

**Parameters:**

`i` - the value to be checked  
`min` - the minimum allowed value  
`max` - the maximum allowed value  
`msg` - a String containing the description of the object being checked

**Returns:**

true if the value is within the range specified

---

## rangeCheck

```
public boolean rangeCheck(int i,  
    int min,  
    int max,  
    String msg,  
    boolean printMsg)
```

Checks whether the given value `i` is within the range specified by the `min` and `max`.

**Parameters:**

`i` - the value to be checked  
`min` - the minimum allowed value  
`max` - the maximum allowed value  
`msg` - a String containing the description of the object being checked  
`printMsg` - if true, an error message will be printed if the value is out of range

**Returns:**

true if the value is within the range specified

---

## rangeCheck

```
public boolean rangeCheck(double i,  
    double min,  
    double max,  
    String msg)
```

Checks whether the given value `i` is within the range specified by the `min` and `max`.

**Parameters:**

`i` - the value to be checked  
`min` - the minimum allowed value  
`max` - the maximum allowed value  
`msg` - a String containing the description of the object being checked  
`printMsg` - if true, an error message will be printed if the value is out of range

**Returns:**

true if the value is within the range specified

## rangeCheck

```
public boolean rangeCheck(double i,  
    double min,  
    double max,  
    String msg,  
    boolean printMsg)
```

Checks whether the given value *i* is within the range specified by the *min* and *max*.

### Parameters:

*i* - the value to be checked  
*min* - the minimum allowed value  
*max* - the maximum allowed value  
*msg* - a String containing the description of the object being checked  
*printMsg* - if true, an error message will be printed if the value is out of range

### Returns:

true if the value is within the range specified

---

## rangeCheck

```
public boolean rangeCheck(float i,  
    float min,  
    float max,  
    String msg)
```

Checks whether the given value *i* is within the range specified by the *min* and *max*.

### Parameters:

*i* - the value to be checked  
*min* - the minimum allowed value  
*max* - the maximum allowed value  
*msg* - a String containing the description of the object being checked

### Returns:

true if the value is within the range specified

---

## rangeCheck

```
public boolean rangeCheck(float i,  
    float min,  
    float max,  
    String msg,  
    boolean printMsg)
```

Checks whether the given value *i* is within the range specified by the *min* and *max*.

### Parameters:

*i* - the value to be checked  
*min* - the minimum allowed value  
*max* - the maximum allowed value  
*msg* - a String containing the description of the object being checked  
*printMsg* - if true, an error message will be printed if the value is out of range

### Returns:

true if the value is within the range specified

---

(continued from last page)

## rangeCheck

```
public boolean rangeCheck(Int i,  
    int min,  
    int max,  
    String msg)
```

Checks whether the given value i is within the range specified by the min and max.

### Parameters:

i - the Int value to be checked  
min - the minimum allowed value  
max - the maximum allowed value  
msg - a String containing the description of the object being checked

### Returns:

true if the value is within the range specified

---

## rangeCheck

```
public boolean rangeCheck(Int i,  
    int min,  
    int max,  
    String msg,  
    boolean printMsg)
```

Checks whether the given value i is within the range specified by the min and max.

### Parameters:

i - the Int value to be checked  
min - the minimum allowed value  
max - the maximum allowed value  
msg - a String containing the description of the object being checked  
printMsg - if true, an error message will be printed if the value is out of range

### Returns:

true if the value is within the range specified

---

## rangeCheck

```
public boolean rangeCheck(Real i,  
    double min,  
    double max,  
    String msg)
```

Checks whether the given value i is within the range specified by the min and max.

### Parameters:

i - the Real value to be checked  
min - the minimum allowed value  
max - the maximum allowed value  
msg - a String containing the description of the object being checked

### Returns:

true if the value is within the range specified

---

(continued from last page)

## rangeCheck

```
public boolean rangeCheck(Real r,  
    double min,  
    double max,  
    String msg,  
    boolean printMsg)
```

Checks whether the given value *i* is within the range specified by the min and max.

### Parameters:

*i* - the value to be checked  
*min* - the minimum allowed value  
*max* - the maximum allowed value  
*msg* - a String containing the description of the object being checked  
*printMsg* - if true, an error message will be printed if the value is out of range

### Returns:

true if the value is within the range specified

---

## writeSP

```
public static void writeSP(java.io.PrintWriter out,  
    String text,  
    int width)
```

Write a fixed width string to a PrintWriter, padding the left with spaces. If the given text is longer than width, it will be truncated.

### Parameters:

*out* - the java.io.PrintWriter to write the padded String to.  
*text* - the String to pad and write out to out.  
*width* - the desired width to pad text to.

---

## writeSP

```
public static void writeSP(java.io.PrintWriter out,  
    int value,  
    int width)
```

Write an integer to a PrintWriter as a fixed width string, padding the left with spaces. If the given text is longer than width, it will be truncated.

### Parameters:

*out* - the java.io.PrintWriter to write the padded String to.  
*value* - the int to write to the given writer  
*width* - the desired width to pad value to.

---

## writeMuxLoadArray

```
public static void writeMuxLoadArray(java.io.PrintWriter out,  
    String header,  
    Object[] xValues,  
    Object[] yValues)
```

writes the given demultiplexed x,y pairs to the given writer in multiplexed TRAC LOAD format.

### Parameters:

*out* - the PrintWriter to write the values to.  
*header* - the comment/header to begin each line with; cell length uses "\* dx \*";  
*xValues* - an Object[] containing the X values to be written.  
*yValues* - an Object[] containing the Y values to be written.

## writeMuxLoadArray

```
public static void writeMuxLoadArray(java.io.PrintWriter out,  
    String header,  
    Real[] xValues,  
    Real[] yValues)
```

writes the given demultiplexed x,y pairs to the given writer in multiplexed TRAC LOAD format.

### Parameters:

- out - the PrintWriter to write the values to.
  - header - the comment/header to begin each line with; cell length uses "\* dx \*";
  - xValues - an Object[] containing the X values to be written.
  - yValues - an Object[] containing the Y values to be written.
- 

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    int[] values)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

- out - the PrintWriter to write the values to.
  - header - the comment/header to begin each line with; cell length uses "\* dx \*";
  - values - an int[] containing the values to be written.
- 

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    int[] values,  
    int columns)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

- out - the PrintWriter to write the values to.
  - header - the comment/header to begin each line with; cell length uses "\* dx \*";
  - values - an int[] containing the values to be written.
  - columns - the number of values to output per line
- 

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    Real[] values)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

- out - the PrintWriter to write the values to.
  - header - the comment/header to begin each line with; cell length uses "\* dx \*";
  - values - an Object[] containing the values to be written.
-

(continued from last page)

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    Real[] values,  
    int columns)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

out - the PrintWriter to write the values to.  
header - the comment/header to begin each line with; cell length uses "\* dx \*";  
values - an Object[] containing the values to be written.  
columns - the number of values to output per line

---

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    Dimless[] values)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

out - the PrintWriter to write the values to.  
header - the comment/header to begin each line with; cell length uses "\* dx \*";  
values - an Dimless[] containing the values to be written.

---

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    Dimless[] values,  
    int columns)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

out - the PrintWriter to write the values to.  
header - the comment/header to begin each line with; cell length uses "\* dx \*";  
values - an Dimless[] containing the values to be written.  
columns - the number of values to output per line

---

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    Object[] values)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

### Parameters:

out - the PrintWriter to write the values to.  
header - the comment/header to begin each line with; cell length uses "\* dx \*";  
values - an Object[] containing the values to be written.

---

## writeArrayLoadValue

```
public static void writeArrayLoadValue(java.io.PrintWriter out,  
    String header,  
    Object[] values,  
    int columns)
```

writes out a value in TRACE 'LOAD Format' for each element in the given array.

(continued from last page)

**Parameters:**

- out - the PrintWriter to write the values to.
- header - the comment/header to begin each line with; cell length uses "\* dx \*";
- values - an Object[] containing the values to be written.
- columns - the number of values to output per line



## com.cafean.client.analysis Interface IdentHolder

All Known Implementing Classes:  
GenericObject

public interface **IdentHolder**

A placeholder interface indicating that this object contains ident references that must be reconnected during  
AbstractModel#reconnectIdentReferences

### Method Summary

|      |   |
|------|---|
| void | <code>reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)</code><br>Resets this component's internal ident references to refer to appropriate components in the current AbstractModel. |
|------|---|

### Methods

#### reconnectIdentReferences

```
public void reconnectIdentReferences(boolean preserveUnresolved,  
    boolean useDbId)
```

Resets this component's internal ident references to refer to appropriate components in the current AbstractModel. Intended for use after adding a component to a model and thus its DB\_ID will have been set to its ident in the previous model. For ident references use find\_\_ByDbId to find the new component, then store its ident. Note: If this component's DB\_ID is 0 this method does nothing.

##### Parameters:

`preserveUnresolved` - if true, ident references that aren't resolvable will be left dangling, if false, they will be set to 0.  
`useDbId` - if true, ident references will be reconnected via find\_x\_ByDB\_ID; if false, ident references will be reconnected via find\_x\_ByIdent

## com.cafean.client.analysis Interface ModelDependent

### All Known Implementing Classes:

ComponentSelectionEditor, RealArrayEditor, RealBeanEditor, RealTextField, RealEditor

---

### public interface **ModelDependent**

An interface describing an object that is dependent on an AbstractModel either directly or by being a part of an object that is.

---

## Method Summary

|                               |   |
|-------------------------------|---|
| <a href="#">AbstractModel</a> | <code>getModel()</code><br>Retrieves the model that this object depends on. This method may call a parent's <code>getModel()</code> and so on rather than a direct reference. |
| <code>void</code>             | <code>setModel(AbstractModel model)</code><br>Sets the model that this object depends on.   |

---

## Methods

### getModel

public [AbstractModel](#) **getModel()**

Retrieves the model that this object depends on. This method may call a parent's `getModel()` and so on rather than a direct reference.

#### Returns:

the AbstractModel that this object depends on

---

### setModel

public void **setModel**([AbstractModel](#) model)

Sets the model that this object depends on.

#### Parameters:

model - the AbstractModel that this object will now depend on

---

## com.cafean.client.analysis Interface ModelElement

### All Subinterfaces:

[ComponentElement](#)

### All Known Implementing Classes:

AbstractModel

---

public interface **ModelElement**

An interface describing an object that is contained by an `AbstractModel` either directly or by being a part of an object that is.

---

## Method Summary

[AbstractModel](#)

`getModel()`

Retrieves the model that contains this `ModelElement`. This method may call a parent's `getModel()` and so on rather than a direct reference.

---

## Methods

### **getModel**

public [AbstractModel](#) **getModel()**

Retrieves the model that contains this `ModelElement`. This method may call a parent's `getModel()` and so on rather than a direct reference.

#### **Returns:**

the `AbstractModel` that contains this `ModelElement`

## com.cafean.client.analysis Interface SharedComponent

public interface **SharedComponent**

An interface describing an AbstractComponent that is shared among other components. Examples include shared geometry or materials objects.

### Method Summary

boolean

`isEquivalent(SharedComponent component)`

Returns true if this and the given shared component are equivalent.

### Methods

#### **isEquivalent**

public boolean **isEquivalent**([SharedComponent](#) component)

Returns true if this and the given shared component are equivalent. This differs from `#equals` in that the given component is not necessarily required to have identical data to be equivalent.

##### **Parameters:**

component - the SharedComponent to test for equivalence

## com.cafean.client.analysis Class SpecialConnectionData

```
java.lang.Object
|
+-com.cafean.client.analysis.ConnectionData
|   |
|   +-com.cafean.client.analysis.SpecialConnectionData
```

public class **SpecialConnectionData**  
extends [ConnectionData](#)

SpecialConnectionData is an extension of ConnectionData that tells the system that the user must supply information for the connection to proceed.

### See Also:

[AbstractComponent#connectTo](#), [AbstractComponent#createSourceData](#), [AbstractComponent#createTargetData](#)

## Constructor Summary

|        |   |
|--------|---|
| public | <code>SpecialConnectionData()</code><br>Creates a new instance of SpecialConnectData                                |
| public | <code>SpecialConnectionData(int index)</code><br>Creates a new instance of SpecialConnectData with the given index. |

## Method Summary

|        |                         |
|--------|-------------------------|
| Object | <code>clone()</code>    |
| String | <code>toString()</code> |

### Methods inherited from class [com.cafean.client.analysis.ConnectionData](#)

[clone](#), [equals](#), [getConnectionIndex](#), [setConnectionIndex](#), [toString](#)

### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### SpecialConnectionData

public **SpecialConnectionData()**  
Creates a new instance of SpecialConnectData

## SpecialConnectionData

```
public SpecialConnectionData(int index)
```

Creates a new instance of SpecialConnectData with the given index.

### Parameters:

index - {@inheritDoc}

## Methods

### clone

```
public Object clone()
```

---

### toString

```
public String toString()
```

## com.cafean.client.analysis Class ViewComponent

```

java.lang.Object
  |
  +- com.cafean.client.analysis.GenericObject
      |
      +- com.cafean.client.analysis.AbstractComponent
          |
          +- com.cafean.client.analysis.ViewComponent
  
```

### All Implemented Interfaces:

[PropertyController](#), [IdentHolder](#), [StateEditable](#), [Cloneable](#), [Checkable](#), [ComponentElement](#), [Cloneable](#)

### public class **ViewComponent**

extends [AbstractComponent](#)

implements [Cloneable](#), [ComponentElement](#), [Checkable](#), [Cloneable](#), [StateEditable](#), [IdentHolder](#), [PropertyController](#)

The ViewComponent is an { @link AbstractComponent } that represents a drawing inside the ModelEditor. This contains all of the information needed to create a new { @link DrawnView } from data stored in a local file. It also contains the picture that is associated with a given view, and gets displayed when it is rendered inside a { @link DrawnView }.

## Field Summary

|                  |                  |
|------------------|------------------|
| static final int | PIXELS_PER_METER |
|                  | Value: 20        |

### Fields inherited from class [com.cafean.client.analysis.GenericObject](#)

[DATA\\_COMPLETE](#), [DATA\\_ERROR](#), [DATA\\_INCOMPLETE](#), [DATA\\_WARNING](#)

## Constructor Summary

|        |   |
|--------|---|
| public | ViewComponent()<br>Creates a new instance of ViewComponent  |
| public | ViewComponent(AbstractModel model,int num)<br>Creates a new instance of ViewComponent with a new model and display number |

## Method Summary

|      |  |
|------|--|
| void | addAnnotation(Annotation comp)<br>Adds an Annotation to this ViewComponent, or to the DrawnView if the view has been opened.   |
| void | addComponent(AbstractComponent comp,boolean select)<br>Adds a new component to this view, either by adding it's pibblock if the view is not created, or by adding it directly to the view. |

|                                   |   |
|-----------------------------------|---|
| void                              | <code>addComponents(Iterator itr)</code><br>Uses the given iterator to add all of those components to the View, if it exists, or adds them to the PibBlock list if they don't.                |
| void                              | <code>addComponents(Iterator itr,boolean select)</code><br>Uses the given iterator to add all of those components to the View, if it exists, or adds them to the PibBlock list if they don't. |
| void                              | <code>addPibBlock(com.appt.xdr.PibBlock block)</code><br>Adds a new Pibblock to this View Component.  |
| void                              | <code>buildView()</code><br>This initializes a new DrawnView based on a View Component.   |
| boolean                           | <code>canConnectTo(AbstractComponent target)</code><br>This method checks to see if it is allowable for this AbstractComponent to connect to the given target.                                |
| void                              | <code>clearViewSelection()</code><br>Clears the current selection on the View if it exists  |
| Object                            | <code>clone()</code><br>this clone method copies all primitive data types.  |
| void                              | <code>complete()</code><br>When a ViewComponent is created, it immediately opens.   |
| <a href="#">AbstractComponent</a> | <code>copy(AbstractModel sm)</code><br>the following copy method produces a deep clone of a composite base so that it can be put in a copy model clipboard.                                   |
| void                              | <code>copyFrom(GenericObject o)</code><br>Copy the attributes from a clone to this instance.  |
| <a href="#">DrawnComponent</a>    | <code>createDrawnComponent()</code><br><code>{ @inheritDoc }</code>   |
| void                              | <code>fireComponentChanged()</code><br>This calls the component changed function on all of the listeners currently listening to this abstract component                                       |
| void                              | <code>fireComponentDeleted()</code><br>This calls the component deleted function on all of the listeners currently listening to this abstract component                                       |
| byte[]                            | <code>getAnnotationArray()</code><br>Gets an byte array containing the XML encoded stream of annotations from this DrawnView.   |
| static String                     | <code>getAttributeGroup(String property)</code><br>Retrieves the name of the Attribute Group for the given property name.   |
| static String[]                   | <code>getAttributeGroups()</code><br>Retrieves the Attribute Group Names for this Preferences.  |



|                           |  |
|---------------------------|--|
| int                       | getAttributeIndex(String propertyName)   |
| static String[]           | getAttributesForGroup(String groupName)<br>Retrieves the Attribute Names for the given group.                                      |
| java.awt.Color            | getBackground()<br>Gets the background color from the DrawnView, or from the local variable if the DrawnView has not been created. |
| java.awt.Dimension        | getCanvasSize()<br>Gets the size of the view's canvas or the locally stored canvas size, if the DrawnView has never been opened.   |
| <a href="#">Category</a>  | getCategory()  |
| String                    | getCCnumber()  |
| int[]                     | getColorArray()<br>This returns the current background in the form of an array of three integers.                                  |
| Vector                    | getCustomPopupItems()<br>Creates Custom Menu Items for any popup dialog involving this component                                   |
| <a href="#">DrawnView</a> | getDrawnView()<br>A getter for the DrawnView owned by this ViewComponent.  |
| byte[]                    | getEmbedConnArray()<br>Gets an byte array containing the XML encoded stream of annotations from this DrawnView.                    |
| EmbeddedConnectionMap     | getEmbeddedCons()  |
| java.awt.Color            | getGridLineColor()<br>Getter for property lineColor.   |
| int                       | getHorizGridSpacing()<br>Getter for property horizGridSpacing.   |
| int                       | getHorizSnapSpacing()<br>Getter for property horizSnapSpacing.   |
| ImageIcon                 | getImage()<br>Gets the ImageIcon for rendering this ViewComponent.   |
| byte[]                    | getImageData()<br>Gets the icon image for the DrawnViewComponent that renders this ViewComponent.                                  |
| ViewImgDataElem           | getImageElement()<br>A stub method used in ViewComponentBeanInfo to edit the image data byte[] contained in this ViewComponent.    |
| java.awt.Point            | getLocation()<br>Gets the location of the DrawnView or the locally stored location, if the DrawnView has never been opened.        |

|                    |  |
|--------------------|--|
| int                | getPixelsPerMeter()<br>Getter for property pixelsPerMeter.   |
| boolean            | getShowPoints()<br>Returns whether or not the connection points are drawn on AbstractComponents inside this view.                                    |
| java.awt.Dimension | getSize()<br>Gets the size of the DrawnView or the locally stored size, if the DrawnView has never been opened.                                      |
| float              | getTransparency()<br>Getter for property transparency.   |
| int                | getVertGridSpacing()<br>Getter for property vertGridSpacing.   |
| int                | getVertSnapSpacing()<br>Getter for property vertSnapSpacing.   |
| java.awt.Point     | getViewPosition()<br>Getter for property viewPosition.   |
| double             | getWidthScaleFactor()<br>Gets this view's width scale factor.  |
| double             | getZoomScale()<br>Gets the current zoom factor from the view's zoom panel or the locally stored zoom factor, if the DrawnView has never been opened. |
| boolean            | hasBlocks()<br>Determines whether this view has anything inside it's blocks vector.  |
| boolean            | isGridAbove()<br>Returns true if this View's grid will be painted on top of its contained elements.  |
| boolean            | isLocked()<br>Returns true if this view is locked.   |
| boolean            | isPropertyActive(String propertyName)  |
| boolean            | isPropertyEnabled(String propertyName)   |
| boolean            | isPropertyRequired(String propertyName)  |
| boolean            | isPropertyResizable(String propertyName)   |
| boolean            | isPropertyRestartEditable(String propertyName)   |
| boolean            | isRestartResizable(String propertyName)  |
| boolean            | isShowGrid()<br>Getter for property showGrid.  |

|                              |  |
|------------------------------|--|
| boolean                      | isSnapToGrid()<br>Getter for property snapToGrid.  |
| boolean                      | isViewVisible()<br>Returns true if the view is currently visible, or false if the view is not visible, or doesn't exist yet.   |
| String                       | label()  |
| void                         | layoutView()<br>Runs the organize function on the view component if it exists.   |
| void                         | load(ViewCompRec rec)<br>Loads this view's data from the given ViewCompRec previously stored with #store   |
| static<br>java.awt.Component | loadComponent(com.appt.xdr.PibBlock block, AbstractModel model)  |
| void                         | popupDataDialog(java.awt.Window parent, boolean modal)   |
| void                         | readAnnotationArray(byte[] byteArr, AbstractModel model)<br>Retrieves all of the Annotations from an XML encoded byte array of annotations, and adds them to this ViewComponent. |
| void                         | readEmbedConnArray(byte[] byteArr)<br>Retrieves all of the Annotations from an XML encoded byte array of annotations, and adds them to this ViewComponent.                       |
| void                         | removeFromModel(AbstractModel model)   |
| boolean                      | removeVerify()   |
| void                         | setBackground(java.awt.Color background)<br>Sets the background color on the DrawnView if it has been created, and stores it inside the local variable.                          |
| void                         | setCanvasSize(java.awt.Dimension canvasSize)<br>Sets the locally stored canvas size.   |
| void                         | setColorArray(int[] array)<br>Sets the current background from an array of three integers.   |
| void                         | setEmbeddedCons(EmbeddedConnectionMap cons)  |
| void                         | setGridAbove(boolean above)<br>If set to true, this View's grid will be painted on top of its contained elements.  |
| void                         | setGridLineColor(java.awt.Color lineColor)<br>Setter for property lineColor.   |
| void                         | setHorizGridSpacing(int horizGridSpacing)<br>Setter for property horizGridSpacing.   |

|      |   |
|------|---|
| void | <code>setHorizSnapSpacing(int horizSnapSpacing)</code><br>Setter for property horizSnapSpacing.   |
| void | <code>setImageData(byte[] imageData)</code><br>Sets the icon image for the DrawnViewComponent that renders this ViewComponent.                                      |
| void | <code>setImageElement(ViewImgDataElem element)</code><br>A stub method used in ViewComponentBeanInfo to edit the image data byte[] contained in this ViewComponent. |
| void | <code>setLocation(java.awt.Point location)</code><br>Sets the the locally stored location of the DrawnView.   |
| void | <code>setLocked(boolean locked)</code><br>sets this view's locked state   |
| void | <code>setLockedConstrained(boolean locked)</code><br>sets this view's locked state  |
| void | <code>setPixelsPerMeter(int pixelsPerMeter)</code><br>Setter for property pixelsPerMeter.   |
| void | <code>setShowGrid(boolean showGrid)</code><br>Setter for property showGrid.   |
| void | <code>setShowPoints(boolean showPoints)</code><br>Turns on displaying the connection points on AbstractComponents inside this view.                                 |
| void | <code>setSize(java.awt.Dimension size)</code><br>Sets the size of the DrawnView or the locally stored size, if the DrawnView has never been opened.                 |
| void | <code>setSnapToGrid(boolean snapToGrid)</code><br>Setter for property snapToGrid.   |
| void | <code>setStoredBackground(java.awt.Color background)</code><br>This just sets the background color of the stored local variable.                                    |
| void | <code>setTransparency(float transparency)</code><br>Setter for property transparency.   |
| void | <code>setVertGridSpacing(int vertGridSpacing)</code><br>Setter for property vertGridSpacing.  |
| void | <code>setVertSnapSpacing(int vertSnapSpacing)</code><br>Setter for property vertSnapSpacing.  |
| void | <code>setViewPosition(java.awt.Point viewPosition)</code><br>Setter for property viewPosition.  |
| void | <code>setVisible(boolean value)</code><br>Shows or hides this ViewComponent's DrawnView and it's accompanying dialog or panel.                                      |

|                       |   |
|-----------------------|---|
| void                  | setWidthScaleFactor(double factor)<br>Sets this view's width scale factor.  |
| void                  | setZoomScale(double zoomScale)<br>Sets the locally stored zoom scale.   |
| boolean               | store(com.appt.xdr.PibFile file)<br>Writes this view and it's contents to the given PibFile.                                    |
| com.appt.xdr.PibBlock | storeComponent(java.awt.Component c)<br>Stores the given Component into a PibBlock for use in saving into a PIB formatted file. |
| String                | toString()<br>ViewComponents use their name as their to-string, as opposed to including "View CCnumber" before their name.      |
| void                  | undockView()<br>Builds this view and displays it as an undocked separate window.  |

**Methods inherited from class [com.cafean.client.analysis.AbstractComponent](#)**

[addComponentListener](#), [addConnection](#), [addMessage](#), [addMessage](#), [addToModel](#), [addToModel](#), [canConnectTo](#), [clearConnections](#), [clone](#), [complete](#), [connectTo](#), [connectTo](#), [copy](#), [createDrawnComponent](#), [createSourceData](#), [createTargetData](#), [DBtypeCode](#), [disconnect](#), [disconnectFrom](#), [fireComponentChanged](#), [fireComponentChanged](#), [fireComponentConnected](#), [fireComponentDeleted](#), [fireComponentDisconnected](#), [getCatCCComparator](#), [getCategory](#), [getCCNumberComparator](#), [getComponent](#), [getComponentDependencies](#), [getConnectionCount](#), [getConnectionName](#), [getConnections](#), [getConnectionTypes](#), [getCustomPopupActions](#), [getCustomPopupItems](#), [getGroupedConnections](#), [getModel](#), [getName](#), [getNewCompIdent](#), [getOrder](#), [getOrderComparator](#), [getOwner](#), [getRealSize](#), [getSharedComponents](#), [includeInLoopcheck](#), [isOkayForExport](#), [isOkayForExport](#), [label](#), [popupDataDialog](#), [rebuildConnections](#), [reconnectImage](#), [removeComponentListener](#), [removeFromModel](#), [removeVerify](#), [restoreState](#), [setComponentNumber](#), [setDeleted](#), [setModel](#), [setOrder](#), [toString](#), [updateVersion](#), [writeName](#)

**Methods inherited from class [com.cafean.client.analysis.GenericObject](#)**

[addComment](#), [addMultipleComments](#), [checkRealArrayList](#), [checkRealArrayTable](#), [clearDbIds](#), [clone](#), [closeAllViews](#), [compareTo](#), [copyFrom](#), [createDataPages](#), [debug](#), [deleteAllComments](#), [deleteComment](#), [equals](#), [fixme](#), [getCCnumber](#), [getComment](#), [getComments](#), [getComments](#), [getComponentCCNumber](#), [getComponentNumber](#), [getDataState](#), [getDB\\_ID](#), [getDescription](#), [getIdent](#), [getMajorCreationVersion](#), [getMajorVersion](#), [getMinorCreationVersion](#), [getMinorVersion](#), [getName](#), [getNewCompIdent](#), [getNumComments](#), [isDeleted](#), [popupDataDialog](#), [popupDataDialog](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [rangeCheck](#), [reconnectIdentReferences](#), [restoreState](#), [restoreState](#), [setComments](#), [setComments](#), [setComponentNumber](#), [setCreationVersion](#), [setDataState](#), [setDB\\_ID](#), [setDeleted](#), [setDescription](#), [setIdent](#), [setMajorCreationVersion](#), [setMajorVersion](#), [setMinorCreationVersion](#), [setMinorVersion](#), [setName](#), [showComment](#), [storeState](#), [storeState](#), [trace](#), [updateVersion](#), [validate](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeArrayLoadValue](#), [writeMuxLoadArray](#), [writeMuxLoadArray](#), [writeSP](#), [writeSP](#)

**Methods inherited from class [java.lang.Object](#)**

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

(continued from last page)

## Fields

### PIXELS\_PER\_METER

```
public static final int PIXELS_PER_METER
```

## Constructors

### ViewComponent

```
public ViewComponent()
```

Creates a new instance of ViewComponent

### ViewComponent

```
public ViewComponent(AbstractModel model,  
                     int num)
```

Creates a new instance of ViewComponent with a new model and display number

## Methods

### removeVerify

```
public boolean removeVerify()
```

### removeFromModel

```
public void removeFromModel(AbstractModel model)
```

### load

```
public void load(ViewCompRec rec)
```

Loads this view's data from the given ViewCompRec previously stored with #store

### clone

```
public Object clone()
```

this clone method copies all primitive data types. it will only need to be overridden if inherited classes contain objects which need to be copied. This should not be used for copy/paste as it initializes contained DrawnComponents and thus breaks the subclass copy() methods.

### copy

```
public AbstractComponent copy(AbstractModel sm)
```

the following copy method produces a deep clone of a composite base so that it can be put in a copy model clipboard. Connections to objects in the Vector which is the argument are preserved. Connections to all other objects are removed. NOTE: COPY/PASTE should use this in place of clone()

(continued from last page)

---

## label

```
public String label()
```

---

---

## copyFrom

```
public void copyFrom(GenericObject o)
```

Copy the attributes from a clone to this instance.

**Parameters:**

- o - The cloned object.
- 

---

## toString

```
public String toString()
```

ViewComponents use their name as their to-string, as opposed to including "View CCnumber" before their name.

---

---

## canConnectTo

```
public boolean canConnectTo(AbstractComponent target)
```

This method checks to see if it is allowable for this AbstractComponent to connect to the given target.

**Parameters:**

- target - the AbstractComponent object to which a connection has been requested.

**Returns:**

true if allowed, false if not allowed.

---

---

## getCategory

```
public Category getCategory()
```

---

---

## setVisible

```
public void setVisible(boolean value)
```

Shows or hides this ViewComponent's DrawnView and it's accompanying dialog or panel.

---

---

## layoutView

```
public void layoutView()
```

Runs the organize function on the view component if it exists.

---

---

## buildView

```
public void buildView()
```

This initializes a new DrawnView based on a View Component. Since this does not set the view visible, functions can be called on the view before it is displayed. This is used to speed up generating a new view of components.

---

---

## addPibBlock

```
public void addPibBlock(com.apr.xdr.PibBlock block)
```

Adds a new Pibblock to this View Component. While loading a sam file drawing records are added to the ViewComponent without translating them at all.

---

## addComponent

```
public void addComponent(AbstractComponent comp,  
                           boolean select)
```

Adds a new component to this view, either by adding it's pibblock if the view is not created, or by adding it directly to the view.

---

## addComponents

```
public void addComponents(Iterator itr)
```

Uses the given iterator to add all of those components to the View, if it exists, or adds them to the PibBlock list if they don't.

**Parameters:**

itr - An Iterator on a list of AbstractComponents

---

## addComponents

```
public void addComponents(Iterator itr,  
                           boolean select)
```

Uses the given iterator to add all of those components to the View, if it exists, or adds them to the PibBlock list if they don't.

**Parameters:**

itr - an Iterator on a list of AbstractComponents

select - if true and the view is visible the added components will be selected.

---

## hasBlocks

```
public boolean hasBlocks()
```

Determines whether this view has anything inside it's blocks vector. Basically whether this view has been opened or is empty.

**Returns:**

true if this view is new, or has already been opened.

---

## store

```
public boolean store(com.apt.xdr.PibFile file)
```

Writes this view and it's contents to the given PibFile. May call AbstractComponent#storeDrawnComponent to create pib blocks for components that are not part of the core ModelEditor distribution.

**Parameters:**

file - the PibFile to write this ViewComponent to

**Returns:**

true on success; false on failure with messages printed to the message window.

---

## storeComponent

```
public com.apt.xdr.PibBlock storeComponent(java.awt.Component c)
```

Stores the given Component into a PibBlock for use in saving into a PIB formatted file.

**Parameters:**

a - PibBlock containing the stored component

---



## getDrawnView

```
public DrawnView getDrawnView()
```

A getter for the DrawnView owned by this ViewComponent. This is NULL if this view has never been opened.

**Returns:**

The DrawnView owned by this ViewComponent

---

## getLocation

```
public java.awt.Point getLocation()
```

Gets the location of the DrawnView or the locally stored location, if the DrawnView has never been opened.

**Returns:**

the java.awt.Point location of the DrawnView on the screen.

---

## setLocation

```
public void setLocation(java.awt.Point location)
```

Sets the the locally stored location of the DrawnView. This is used for loading the data from the file, or setting the initial position.

**Parameters:**

location - the java.awt.Point location of the DrawnView on the screen.

---

## getSize

```
public java.awt.Dimension getSize()
```

Gets the size of the DrawnView or the locally stored size, if the DrawnView has never been opened.

**Returns:**

the java.awt.Dimension size of the DrawnView on the screen.

---

## setSize

```
public void setSize(java.awt.Dimension size)
```

Sets the size of the DrawnView or the locally stored size, if the DrawnView has never been opened.

**Parameters:**

size - the java.awt.Dimension size of the DrawnView on the screen.

---

## getCanvasSize

```
public java.awt.Dimension getCanvasSize()
```

Gets the size of the view's canvas or the locally stored canvas size, if the DrawnView has never been opened.

**Returns:**

the java.awt.Dimension size of the DrawnView's canvas.

**See Also:**

`DrawnView.getCanvasSize()`

---

(continued from last page)

## setCanvasSize

```
public void setCanvasSize(java.awt.Dimension canvasSize)
```

Sets the locally stored canvas size.

**Parameters:**

canvasSize - the java.awt.Dimension size of the DrawnView's canvas.

---

## getZoomScale

```
public double getZoomScale()
```

Gets the current zoom factor from the view's zoom panel or the locally stored zoom factor, if the DrawnView has never been opened.

**Returns:**

the double zoom factor from the DrawnView.

**See Also:**

DrawnView.getZoomScale()

---

## setZoomScale

```
public void setZoomScale(double zoomScale)
```

Sets the locally stored zoom scale.

**Parameters:**

zoomScale - the double for the zoom scale factor of the DrawnView.

---

## setShowPoints

```
public void setShowPoints(boolean showPoints)
```

Turns on displaying the connection points on AbstractComponents inside this view.

---

## getShowPoints

```
public boolean getShowPoints()
```

Returns whether or not the connection points are drawn on AbstractComponents inside this view.

**Returns:**

true if the connection points are drawn.

---

## undockView

```
public void undockView()
```

Builds this view and displays it as an undocked separate window. This method does nothing in Multiple Window Arrangement.

---

## getCustomPopupItems

```
public Vector getCustomPopupItems()
```

Creates Custom Menu Items for any popup dialog involving this component

---

## getViewPosition

```
public java.awt.Point getViewPosition()
```

---

(continued from last page)

Getter for property viewPosition.

**Returns:**

Value of property viewPosition.

---

## setViewPosition

```
public void setViewPosition(java.awt.Point viewPosition)
```

Setter for property viewPosition.

**Parameters:**

viewPosition - New value of property viewPosition.

---

## isViewVisible

```
public boolean isViewVisible()
```

Returns true if the view is currently visible, or false if the view is not visible, or doesn't exist yet.

---

## getImage

```
public ImageIcon getImage()
```

Gets the ImageIcon for rendering this ViewComponent.

**Returns:**

the ImageIcon for the DrawnViewComponent

---

## getImageData

```
public byte[] getImageData()
```

Gets the icon image for the DrawnViewComponent that renders this ViewComponent.

**Returns:**

the byte[] holding the raw image data directly from the file.

---

## setImageData

```
public void setImageData(byte[] imageData)
```

Sets the icon image for the { @link DrawnViewComponent } that renders this ViewComponent. This also creates the image from the raw datas.

**Parameters:**

imageData - the byte[] holding the raw image data directly from the file.

---

## createDrawnComponent

```
public DrawnComponent createDrawnComponent()
```

Returns the renderer for this abstract component. This will be extended by any abstract component that needs a renderer.

---

## clearViewSelection

```
public void clearViewSelection()
```

Clears the current selection on the View if it exists

(continued from last page)

---

## addAnnotation

```
public void addAnnotation(Annotation comp)
```

Adds an Annotation to this ViewComponent, or to the DrawnView if the view has been opened.

**Parameters:**

comp - the Annotation.

---

## getAnnotationArray

```
public byte[] getAnnotationArray()
```

Gets an byte array containing the XML encoded stream of annotations from this DrawnView. This byte array is used to store the annotations into the PibBlock for this ViewComponent

**Returns:**

the byte[] containing all of the XML encoded annotations.

---

## readAnnotationArray

```
public void readAnnotationArray(byte[] byteArr,  
    AbstractModel model)
```

Retrieves all of the Annotations from an XML encoded byte array of annotations, and adds them to this ViewComponent.

**Parameters:**

byteArr - the byte[] containing all of the XML encoded annotations.  
model - the AbstractModel.

**See Also:**

[.addAnnotation\(\)](#)

---

## getEmbedConnArray

```
public byte[] getEmbedConnArray()
```

Gets an byte array containing the XML encoded stream of annotations from this DrawnView. This byte array is used to store the annotations into the PibBlock for this ViewComponent

**Returns:**

the byte[] containing all of the XML encoded annotations.

---

## readEmbedConnArray

```
public void readEmbedConnArray(byte[] byteArr)
```

Retrieves all of the Annotations from an XML encoded byte array of annotations, and adds them to this ViewComponent.

**Parameters:**

byteArr - the byte[] containing all of the XML encoded annotations.  
model - the AbstractModel.

**See Also:**

[.addAnnotation\(\)](#)

---

## complete

```
public void complete()
```

When a ViewComponent is created, it immediately opens.

---

---

## fireComponentChanged

```
public void fireComponentChanged()
```

This calls the component changed function on all of the listeners currently listening to this abstract component

---

## fireComponentDeleted

```
public void fireComponentDeleted()
```

This calls the component deleted function on all of the listeners currently listening to this abstract component

---

## getBackground

```
public java.awt.Color getBackground()
```

Gets the background color from the DrawnView, or from the local variable if the DrawnView has not been created.

**Returns:**

Value of property background.

---

## setBackground

```
public void setBackground(java.awt.Color background)
```

Sets the background color on the DrawnView if it has been created, and stores it inside the local variable.

**Parameters:**

background - the Color of the background.

---

## setStoredBackground

```
public void setStoredBackground(java.awt.Color background)
```

This just sets the background color of the stored local variable. This is used inside the GUI editor, to allow the background to be changed and still be cancellable.

**Parameters:**

background - the Color of the background.

---

## getColorArray

```
public int[] getColorArray()
```

This returns the current background in the form of an array of three integers.

**Returns:**

the int[] containing the RGB values for the background color.

**See Also:**

.getBackground()

---

## setColorArray

```
public void setColorArray(int[] array)
```

Sets the current background from an array of three integers.

**Parameters:**

array - the int[] containing the RGB values for the background color.

**See Also:**

---

(continued from last page)

```
.getBackground()
```

---

## loadComponent

```
public static java.awt.Component loadComponent(com.appt.xdr.PibBlock block,  
AbstractModel model)
```

---

## getPixelsPerMeter

```
public int getPixelsPerMeter()
```

Getter for property pixelsPerMeter.

### Returns:

Value of property pixelsPerMeter.

---

## setPixelsPerMeter

```
public void setPixelsPerMeter(int pixelsPerMeter)
```

Setter for property pixelsPerMeter.

### Parameters:

pixelsPerMeter - New value of property pixelsPerMeter.

---

## getEmbeddedCons

```
public EmbeddedConnectionMap getEmbeddedCons()
```

---

## setEmbeddedCons

```
public void setEmbeddedCons(EmbeddedConnectionMap cons)
```

---

## setImageElement

```
public void setImageElement(ViewImgDataElem element)
```

A stub method used in ViewComponentBeanInfo to edit the image data byte[] contained in this ViewComponent.

---

## getImageElement

```
public ViewImgDataElem getImageElement()
```

A stub method used in ViewComponentBeanInfo to edit the image data byte[] contained in this ViewComponent.

---

## popupDataDialog

```
public void popupDataDialog(java.awt.Window parent,  
boolean modal)
```

---

## isShowGrid

```
public boolean isShowGrid()
```

Getter for property showGrid.

(continued from last page)

**Returns:**

Value of property showGrid.

---

**setShowGrid**

```
public void setShowGrid(boolean showGrid)
```

Setter for property showGrid.

**Parameters:**

showGrid - New value of property showGrid.

---

**isSnapToGrid**

```
public boolean isSnapToGrid()
```

Getter for property snapToGrid.

**Returns:**

Value of property snapToGrid.

---

**setSnapToGrid**

```
public void setSnapToGrid(boolean snapToGrid)
```

Setter for property snapToGrid.

**Parameters:**

snapToGrid - New value of property snapToGrid.

---

**getGridLineColor**

```
public java.awt.Color getGridLineColor()
```

Getter for property lineColor.

**Returns:**

Value of property lineColor.

---

**setGridLineColor**

```
public void setGridLineColor(java.awt.Color lineColor)
```

Setter for property lineColor.

**Parameters:**

lineColor - New value of property lineColor.

---

**getTransparency**

```
public float getTransparency()
```

Getter for property transparency.

**Returns:**

Value of property transparency.

---

**setTransparency**

```
public void setTransparency(float transparency)
```

Setter for property transparency.

(continued from last page)

**Parameters:**

transparency - New value of property transparency.

---

**getHorizGridSpacing**

```
public int getHorizGridSpacing()
```

Getter for property horizGridSpacing.

**Returns:**

Value of property horizGridSpacing.

---

**setHorizGridSpacing**

```
public void setHorizGridSpacing(int horizGridSpacing)
```

Setter for property horizGridSpacing.

**Parameters:**

horizGridSpacing - New value of property horizGridSpacing.

---

**getVertGridSpacing**

```
public int getVertGridSpacing()
```

Getter for property vertGridSpacing.

**Returns:**

Value of property vertGridSpacing.

---

**setVertGridSpacing**

```
public void setVertGridSpacing(int vertGridSpacing)
```

Setter for property vertGridSpacing.

**Parameters:**

vertGridSpacing - New value of property vertGridSpacing.

---

**getVertSnapSpacing**

```
public int getVertSnapSpacing()
```

Getter for property vertSnapSpacing.

**Returns:**

Value of property vertSnapSpacing.

---

**setVertSnapSpacing**

```
public void setVertSnapSpacing(int vertSnapSpacing)
```

Setter for property vertSnapSpacing.

**Parameters:**

vertSnapSpacing - New value of property vertSnapSpacing.

---

**getHorizSnapSpacing**

```
public int getHorizSnapSpacing()
```



(continued from last page)

Getter for property horizSnapSpacing.

**Returns:**

Value of property horizSnapSpacing.

---

## setHorizSnapSpacing

```
public void setHorizSnapSpacing(int horizSnapSpacing)
```

Setter for property horizSnapSpacing.

**Parameters:**

horizSnapSpacing - New value of property horizSnapSpacing.

---

## isLocked

```
public boolean isLocked()
```

Returns true if this view is locked.

---

## setLocked

```
public void setLocked(boolean locked)
```

sets this view's locked state

---

## setLockedConstrained

```
public void setLockedConstrained(boolean locked)
```

sets this view's locked state

---

## isGridAbove

```
public boolean isGridAbove()
```

Returns true if this View's grid will be painted on top of its contained elements.

---

## setGridAbove

```
public void setGridAbove(boolean above)
```

If set to true, this View's grid will be painted on top of its contained elements.

---

## isPropertyEnabled

```
public boolean isPropertyEnabled(String propertyName)
```

---

## isPropertyRequired

```
public boolean isPropertyRequired(String propertyName)
```

---

## isPropertyRestartEditable

```
public boolean isPropertyRestartEditable(String propertyName)
```

---

(continued from last page)

---

## getAttributeIndex

```
public int getAttributeIndex(String propertyName)
```

---

## isPropertyResizable

```
public boolean isPropertyResizable(String propertyName)
```

---

## isRestartResizable

```
public boolean isRestartResizable(String propertyName)
```

---

## isPropertyActive

```
public boolean isPropertyActive(String propertyName)
```

---

## getAttributeGroups

```
public static String[] getAttributeGroups()
```

Retrieves the Attribute Group Names for this Preferences.

**Returns:**

a String[] in which each entry is an attribute group name

---

## getAttributeGroup

```
public static String getAttributeGroup(String property)
```

Retrieves the name of the Attribute Group for the given property name.

**Parameters:**

property - a String containing the name of the property

**Returns:**

a String containing the group name for the given property or null

---

## getAttributesForGroup

```
public static String[] getAttributesForGroup(String groupName)
```

Retrieves the Attribute Names for the given group.

**Returns:**

a String[] containing the attribute names for the given group name or a 0 length String[] if none are found.

---

## getCCnumber

```
public String getCCnumber()
```

---

(continued from last page)

## **getWidthScaleFactor**

```
public double getWidthScaleFactor()
```

Gets this view's width scale factor. This factor is intended for use in scaling the diameter or width of components that may have one dimension much larger than the other; such as a 10 meter long, 0.1 meter wide pipe.

---

## **setWidthScaleFactor**

```
public void setWidthScaleFactor(double factor)
```

Sets this view's width scale factor. This factor is intended for use in scaling the diameter or width of components that may have one dimension much larger than the other; such as a 10 meter long, 0.1 meter wide pipe.

---

## Package

# **com.cafean.client.event**

This package contains a set of event objects used to notify listeners of various AbstractComponent related events.

## com.cafean.client.event Class ArrayChangedEvent

```

java.lang.Object
  |
  +- java.util.EventObject
        |
        +- com.cafean.client.event.ComponentChangedEvent
              |
              +- com.cafean.client.event.ArrayChangedEvent
  
```

### Direct Known Subclasses:

[ElementsRemovedEvent](#), [ElementsAddedEvent](#)

```

public class ArrayChangedEvent
extends ComponentChangedEvent
  
```

An event object describing a change to an array contained in a component.

## Constructor Summary

|        |   |
|--------|---|
| public | ArrayChangedEvent (AbstractComponent component, Object[] oldArr, Object[] newArr)<br><br>Creates a new array changed event for the given component with the given old and new array references. |
|--------|---|

## Method Summary

|          |   |
|----------|---|
| Object[] | getNewArray()<br><br>Retrieves the new array reference. |
| Object[] | getOldArray()<br><br>Retrieves the old array reference. |

### Methods inherited from class [com.cafean.client.event.ComponentChangedEvent](#)

[getComponent](#)

### Methods inherited from class java.util.EventObject

getSource, toString

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

(continued from last page)

## ArrayChangedEvent

```
public ArrayChangedEvent(AbstractComponent component,  
                        Object[] oldArr,  
                        Object[] newArr)
```

Creates a new array changed event for the given component with the given old and new array references.

### Parameters:

component - the AbstractComponent that has changed  
oldArr - an Object[] reference to the original array  
newArr - an Object[] reference to the newly changed array

## Methods

### getNewArray

```
public Object[] getNewArray()
```

Retrieves the new array reference. Note: The old and new array references may be the same array and may be the same or different sizes.

### getOldArray

```
public Object[] getOldArray()
```

Retrieves the old array reference. Note: The old and new array references may be the same array and may be the same or different sizes.

## com.cafean.client.event Class ComponentChangedEvent

```

java.lang.Object
  |
  +- java.util.EventObject
        |
        +- com.cafean.client.event.ComponentChangedEvent
  
```

Direct Known Subclasses:

[ArrayChangedEvent](#)

```

public class ComponentChangedEvent
extends EventObject
  
```

An event object describing a general change to an AbstractComponent. Derivative classes can and should describe a more specific change.

### Constructor Summary

|        |   |
|--------|---|
| public | ComponentChangedEvent(AbstractComponent component)<br>Creates a new instance of ComponentChangedEvent |
|--------|---|

### Method Summary

|                                   |  |
|-----------------------------------|--|
| <a href="#">AbstractComponent</a> | getComponent()<br>retrieves the component whos change is described by this event |
|-----------------------------------|--|

#### Methods inherited from class java.util.EventObject

getSource, toString

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### ComponentChangedEvent

```

public ComponentChangedEvent(AbstractComponent component)
    Creates a new instance of ComponentChangedEvent
  
```

### Methods

(continued from last page)

## getComponent

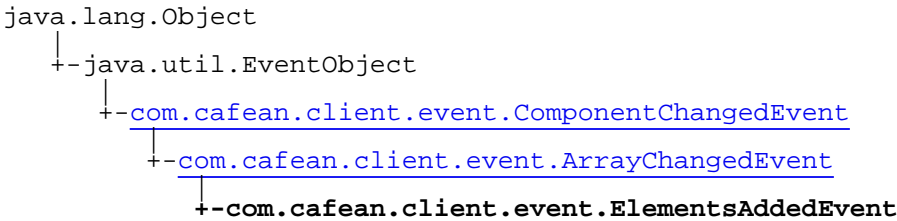
public [AbstractComponent](#) **getComponent**()

retrieves the component whos change is described by this event



com.cafean.client.event

# Class ElementsAddedEvent



```
public class ElementsAddedEvent
extends ArrayChangedEvent
```

An event object describing a change to an array contained in a component.

## Constructor Summary

|        |   |
|--------|---|
| public | <code>ElementsAddedEvent(AbstractComponent component, Object[] oldArr, Object[] newArr, int[] indexes)</code><br><br>Creates a new event describing the addition of elements to an array in a particular component. |
|--------|---|

## Method Summary

|       |   |
|-------|---|
| int[] | <code>getIndexes()</code><br><br>Retrieves the indexes (in the old array), of the elements that were removed. |
|-------|---|

|  |
|--|
| Methods inherited from class <a href="#">com.cafean.client.event.ArrayChangedEvent</a> |
| <a href="#">getNewArray</a> , <a href="#">getOldArray</a>                              |

|  |
|--|
| Methods inherited from class <a href="#">com.cafean.client.event.ComponentChangedEvent</a> |
| <a href="#">getComponent</a>   |

|   |
|---|
| Methods inherited from class <code>java.util.EventObject</code> |
| <code>getSource</code> , <code>toString</code>                  |

|  |
|--|
| Methods inherited from class <code>java.lang.Object</code>   |
| <code>equals</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code> |

## Constructors

(continued from last page)

## ElementsAddedEvent

```
public ElementsAddedEvent(AbstractComponent component,  
                           Object[] oldArr,  
                           Object[] newArr,  
                           int[] indexes)
```

Creates a new event describing the addition of elements to an array in a particular component.

### Parameters:

component - the AbstractComponent that has changed  
oldArr - an Object[] reference to the original array  
newArr - an Object[] reference to the newly changed array

## Methods

### getIndexes

```
public int[] getIndexes()
```

Retrieves the indexes (in the old array), of the elements that were removed.

## com.cafean.client.event Class ElementsRemovedEvent

```

java.lang.Object
  |
  +- java.util.EventObject
        |
        +- com.cafean.client.event.ComponentChangedEvent
              |
              +- com.cafean.client.event.ArrayChangedEvent
                    |
                    +- com.cafean.client.event.ElementsRemovedEvent
  
```

public class **ElementsRemovedEvent**  
 extends [ArrayChangedEvent](#)

An event object describing the removal of elements from an array in a given component.

### Constructor Summary

|        |   |
|--------|---|
| public | <b>ElementsRemovedEvent</b> (AbstractComponent component, Object[] oldArr, Object[] newArr, int[] indexes)<br>Creates a new event describing the removal of elements from an array in a particular component. |
|--------|---|

### Method Summary

|       |  |
|-------|--|
| int[] | <b>getIndexes</b> ()<br>Retrieves the indexes (in the old array), of the elements that were removed. |
|-------|--|

#### Methods inherited from class [com.cafean.client.event.ArrayChangedEvent](#)

[getNewArray](#), [getOldArray](#)

#### Methods inherited from class [com.cafean.client.event.ComponentChangedEvent](#)

[getComponent](#)

#### Methods inherited from class java.util.EventObject

[getSource](#), [toString](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructors

(continued from last page)

## ElementsRemovedEvent

```
public ElementsRemovedEvent(AbstractComponent component,  
                             Object[] oldArr,  
                             Object[] newArr,  
                             int[] indexes)
```

Creates a new event describing the removal of elements from an array in a particular component.

### Parameters:

component - the AbstractComponent that has changed  
oldArr - an Object[] reference to the original array  
newArr - an Object[] reference to the newly changed array

## Methods

### getIndexes

```
public int[] getIndexes()
```

Retrieves the indexes (in the old array), of the elements that were removed.

---

## Package

# com.cafean.client.io

Provides the several utility classes to assist in loading and storing ModelEditor models.

Important classes to note are: Contains a set of static methods used directly to store and load PIB blocks for a given model. An interface describing objects that have an ASCII representation that can be to a `java.io.PrintWriter` for display or export.

- [MEDReader](#)
- [Writeable](#)

## com.cafean.client.io Class MEDReader

java.lang.Object

└─com.cafean.client.io.MEDReader

public class **MEDReader**  
extends Object

This file contains the necessary operations for reading in ModelEditor PibBlocks. These blocks are generic across plugins, and can be read and written using static methods in this class.

### Constructor Summary

|        |             |
|--------|-------------|
| public | MEDReader() |
|--------|-------------|

### Method Summary

|                                |   |
|--------------------------------|---|
| static<br>UserDefinedConstant  | loadUserConstant(UserConstantRec rec,AbstractModel model)<br><br>Loads the given UserConstantRec into a UserDefinedConstant object created in the given model.  |
| static<br>UserDefinedFunction  | loadUserFunction(UserFunctionRec rec,AbstractModel model)<br><br>Loads the given UserFunctionRec into a UserDefinedFunction object created in the given model.  |
| static<br>UserDefinedVariable  | loadUserVariable(UserVariableRec rec,AbstractModel model)<br><br>Loads the given UserVariableRec into a UserDefinedVariable object created in the given model.  |
| static void                    | loadVisualComponents(Vector drawingBlocks,Vector viewBlocks,AbstractModel model)<br><br>Loads the ViewComponent, DrawnComponent and DrawnAnnotation records from the given Vectors.                             |
| static<br>com.apl.xdr.PibBlock | readDrawingBlock(com.apl.xdr.PibFile file,String blockname,int[] blockparm)<br><br>Reads a PibBlock of the required type from the given PibFile if blockname specified is one that is handled by the MEDReader. |
| static<br>UserConstantRec      | storeUserConstant(UserDefinedConstant var)<br><br>Stores the given UserDefinedConstant into a UserConstantRec object for use in saving it into a PIB formatted file.  |
| static<br>UserFunctionRec      | storeUserFunction(UserDefinedFunction var)<br><br>Stores the given UserDefinedFunction into a UserFunctionRec object for use in saving it into a PIB formatted file.  |
| static<br>UserVariableRec      | storeUserVariable(UserDefinedVariable var)<br><br>Stores the given UserDefinedVariable into a UserVariableRec object for use in saving it into a PIB formatted file.  |

**Methods inherited from class** `java.lang.Object``equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructors

### MEDReader

```
public MEDReader()
```

## Methods

### loadVisualComponents

```
public static void loadVisualComponents(Vector drawingBlocks,
    Vector viewBlocks,
    AbstractModel model)
```

Loads the ViewComponent, DrawnComponent and DrawnAnnotation records from the given Vectors. Plugin provided drawing types must be loaded after this method is called and added to the ViewComponent at that time.

**Parameters:**

`drawingBlocks` - a Vector containing the PibBlock derivative drawing blocks to load and add to the appropriate view.  
`viewBlocks` - a Vector containing the ViewCompRec's to create views for.  
`model` - the AbstractModel to add the loaded views to.

**See Also:**

```
ViewComponent.addPibBlock()
```

### readDrawingBlock

```
public static com.appt.xdr.PibBlock readDrawingBlock(com.appt.xdr.PibFile file,
    String blockname,
    int[] blockparm)
```

Reads a PibBlock of the required type from the given PibFile if blockname specified is one that is handled by the MEDReader.

### storeUserVariable

```
public static UserVariableRec storeUserVariable(UserDefinedVariable var)
```

Stores the given UserDefinedVariable into a UserVariableRec object for use in saving it into a PIB formatted file.

### loadUserVariable

```
public static UserDefinedVariable loadUserVariable(UserVariableRec rec,
    AbstractModel model)
```

Loads the given UserVariableRec into a UserDefinedVariable object created in the given model. The loaded variable must be added to the model before use.

### storeUserConstant

```
public static UserConstantRec storeUserConstant(UserDefinedConstant var)
```

(continued from last page)

Stores the given `UserDefinedConstant` into a `UserConstantRec` object for use in saving it into a PIB formatted file.

---

## **loadUserConstant**

```
public static UserDefinedConstant loadUserConstant(UserConstantRec rec,  
                                                    AbstractModel model)
```

Loads the given `UserConstantRec` into a `UserDefinedConstant` object created in the given model. The loaded variable must be added to the model before use.

---

## **storeUserFunction**

```
public static UserFunctionRec storeUserFunction(UserDefinedFunction var)
```

Stores the given `UserDefinedFunction` into a `UserFunctionRec` object for use in saving it into a PIB formatted file.

---

## **loadUserFunction**

```
public static UserDefinedFunction loadUserFunction(UserFunctionRec rec,  
                                                    AbstractModel model)
```

Loads the given `UserFunctionRec` into a `UserDefinedFunction` object created in the given model. The loaded variable must be added to the model before use.



## com.cafean.client.io Interface Writable

---

public interface **Writable**

Interface implemented by components that know how to write themselves.

---

### Method Summary

|      |   |
|------|---|
| void | <code>write(java.io.PrintWriter out)</code><br>Write the ascii output for the component to the given PrintWriter. |
|------|---|

---

### Methods

#### **write**

public void **write**(java.io.PrintWriter out)

Write the ascii output for the component to the given PrintWriter.

#### **Parameters:**

out - the PrintWriter to write this object to.

---

## Package

# com.cafean.client.ui

Provides the basic user interface classes for the ModelEditor.

Classes of note to plugin writers are:

- [ComponentSelector](#) - A selection dialog for components in a model
- [DrawnComponent](#) - A renderer for an [AbstractComponent](#)
- [MainFrame](#) - The central main class for the ModelEditor.
- [NamedValueSelector](#) - A selector for values with a related string.
- [RealTextField](#) - a JTextField that is specialized for working with [Real](#) values
- [TableSorter](#) - A sorter utility for JTable's.

## com.cafean.client.ui Class AsciiViewer

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- java.awt.Window
                    |
                    +- java.awt.Dialog
                          |
                          +- javax.swing.JDialog
                                |
                                +- com.cafean.client.ui.AsciiViewer
  
```

### All Implemented Interfaces:

[RefreshableDialog](#), java.awt.event.MouseListener, [ComponentListener](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, RootPaneContainer, javax.accessibility.Accessible, WindowConstants

```

public class AsciiViewer
  extends JDialog
  implements WindowConstants, javax.accessibility.Accessible, RootPaneContainer,
  javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer,
  java.io.Serializable, ComponentListener, java.awt.event.MouseListener, RefreshableDialog
  
```

A viewer for the ASCII export of a Writeable component.

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | AsciiViewer(java.awt.Frame parent, Writeable comp, MECodePlugin plugin)<br>Creates new viewer to show the given Writeable component with the given Document. |
|--------|--|

## Method Summary

|      |   |
|------|---|
| void | componentChanged(ComponentChangedEvent evt) |
| void | componentConnected(Connection con)          |
| void | componentDeleted()                          |
| void | componentDisconnected(Connection con)       |
| void | createPopupMenu(int x, int y)               |

|                           |   |
|---------------------------|---|
| <a href="#">Writeable</a> | getWriteable()<br>Retrieves the object that this viewer is viewing.                         |
| void                      | mouseClicked(java.awt.event.MouseEvent e)   |
| void                      | mouseEntered(java.awt.event.MouseEvent e)   |
| void                      | mouseExited(java.awt.event.MouseEvent e)  |
| void                      | mousePressed(java.awt.event.MouseEvent e)   |
| void                      | mouseReleased(java.awt.event.MouseEvent e)  |
| void                      | refresh()<br>Updates this AsciiViewer's data  |
| void                      | setTargetComponent(AbstractComponent comp)<br>Sets the listening component for this viewer. |
| void                      | setVisible(boolean visible)   |
| void                      | unitsChanged()  |

**Methods inherited from class javax.swing.JDialog**

getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setJMenuBar, setLayeredPane, setLayout, update

**Methods inherited from class java.awt.Dialog**

addNotify, getAccessibleContext, getTitle, hide, isModal, isResizable, isUndecorated, setModal, setResizable, setTitle, setUndecorated, show

**Methods inherited from class java.awt.Window**

addNotify, addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getAccessibleContext, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, show, toBack, toFront

**Methods inherited from class java.awt.Container**

```

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt,
getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners,
getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets,
invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove,
remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy,
setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward,
transferFocusDownCycle, update, validate

```

#### Methods inherited from class java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

(continued from last page)

## AsciiViewer

```
public AsciiViewer(java.awt.Frame parent,  
                   Writeable comp,  
                   MECodePlugin plugin)
```

Creates new viewer to show the given Writeable component with the given Document.

## Methods

### setTargetComponent

```
public void setTargetComponent(AbstractComponent comp)
```

Sets the listening component for this viewer. This viewer will be added as a { @link [AbstractComponent#addComponentListener](#) component listener} to the given component and will be registered for refresh events.

**Parameters:**

comp - the AbstractComponent target for this viewer

### getWriteable

```
public Writeable getWriteable()
```

Retrieves the object that this viewer is viewing.

### setVisible

```
public void setVisible(boolean visible)
```

### refresh

```
public void refresh()
```

Updates this AsciiViewer's data

### componentChanged

```
public void componentChanged(ComponentChangedEvent evt)
```

### componentConnected

```
public void componentConnected(Connection con)
```

### componentDeleted

```
public void componentDeleted()
```

### componentDisconnected

```
public void componentDisconnected(Connection con)
```

---

## **createPopupMenu**

```
public void createPopupMenu(int x,  
                             int y)
```

---

## **mouseClicked**

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

---

## **mouseEntered**

```
public void mouseEntered(java.awt.event.MouseEvent e)
```

---

## **mouseExited**

```
public void mouseExited(java.awt.event.MouseEvent e)
```

---

## **mousePressed**

```
public void mousePressed(java.awt.event.MouseEvent e)
```

---

## **mouseReleased**

```
public void mouseReleased(java.awt.event.MouseEvent e)
```

---

## **unitsChanged**

```
public void unitsChanged()
```

---

## com.cafean.client.ui Class AsciiViewer.Updater

```

java.lang.Object
  |
  +- java.lang.Thread
        |
        +- com.cafean.client.ui.AsciiViewer.Updater
  
```

public class **AsciiViewer.Updater**  
extends Thread

A background updater thread for ASCII Views

### Fields inherited from class java.lang.Thread

MAX\_PRIORITY, MIN\_PRIORITY, NORM\_PRIORITY

## Constructor Summary

|        |                       |
|--------|-----------------------|
| public | AsciiViewer.Updater() |
|--------|-----------------------|

## Method Summary

|      |   |
|------|---|
| void | disable()<br>Disables and deactivates this Updater.   |
| void | run()<br>Continuously updates and sleeps while enabled to update the AsciiViewer by calling writeComponent. |
| void | update()<br>Requests that an update of the AsciiViewer be performed on a background thread.                 |
| void | Updater()<br>Creates a new Updater as a minimum priority daemon task.                                       |

### Methods inherited from class java.lang.Thread

activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, run, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



## Constructors

### AsciiViewer.Updater

```
public AsciiViewer.Updater()
```

## Methods

### Updater

```
public void Updater()
```

Creates a new Updater as a minimum priority daemon task.

---

### run

```
public void run()
```

Continuously updates and sleeps while enabled to update the AsciiViewer by calling writeComponent.

#### See Also:

```
.disable()  
.update()
```

---

### disable

```
public void disable()
```

Disables and deactivates this Updater.

---

### update

```
public void update()
```

Requests that an update of the AsciiViewer be performed on a background thread.

## com.cafean.client.ui

### Class BeanBox

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JPanel
                          |-- com.cafean.client.ui.BeanBox

```

#### All Implemented Interfaces:

StateEditable, java.io.Serializable, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

public class **BeanBox**

extends JPanel

implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.io.Serializable, StateEditable

The BeanBox is the actual canvas that contains all of the components and [annotations](#) inside the { @link ZoomablePanel } of a { @link DrawnView }. All the methods that manipulate the components in a view, including selecting the components and building the popup menu for right-clicking are included here.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |   |
|--------|---|
| public | BeanBox(AbstractModel model)<br>Constructs a new BeanBox. |
|--------|---|

## Method Summary

|                    |   |
|--------------------|---|
| java.awt.Component | add(java.awt.Component comp)<br>Adds the specified component to this container.   |
| void               | addBoxSelectionListener(BoxSelectionListener listener)<br>Adds the given listener to the list that is notified when the BeanBox selection set may have changed. |

|                                 |   |
|---------------------------------|---|
| <a href="#">DrawnConnection</a> | addDrawnConnection(Connection con)<br>Adds a DrawnConnection object to this BeanBox for the given connection if and only if both sides of the connection are present in this view.  |
| void                            | addNotify()<br>Notifies this component that it now has a parent.  |
| void                            | addSelectedComponent(Object selected)<br>Add a single object to the selection.  |
| void                            | addSelectedComponents(Object[] sel)<br>Add an array of beans to the current selection.  |
| java.awt.Component              | addToFront(java.awt.Component comp)<br>Adds the specified component to this container.  |
| void                            | boundsChanged(DrawnComponent component)<br>This is used to inform the BeanBox that the bounds of a DrawnComponent may have changed because of a component's internal values being edited.                                   |
| void                            | calcSelectionBounds()<br>Calculate the bounds of selected beans.  |
| boolean                         | canAlign()<br>The align operation can only be performed if two or more components are selected.   |
| boolean                         | canCopy()<br>The Copy or cut command can only be performed if there is at least one object selected.  |
| boolean                         | canDelete()<br>The delete operation can only be performed if a copy operation could be performed.   |
| boolean                         | canPaste()<br>The Paste command can only be executed if there are contents on the clipboard, and the Data on the clipboard is appropriate for the ModelEditor.  |
| void                            | clearSelection()<br>Clear the current selection.  |
| void                            | connectionPointRemoved(DrawnComponent component)<br>This is used to inform the BeanBox that the connecting points of a DrawnComponent may have been added or removed because of a component's internal values being edited. |
| boolean                         | copy()<br>Serialize the current selection to the system clipboard.  |
| JPopupMenu                      | createPopupMenu(java.awt.event.MouseEvent evt)<br>Creates a popup menu suitable for this GlassPanel and the model it is viewing.  |
| JMenu                           | createZoomMenu()<br>Creates a Zoom menu suitable for this view.   |

|                                |   |
|--------------------------------|---|
| void                           | cut()<br>Serialize the current selection to the system clipboard, and then remove it from the BeanBox.  |
| void                           | delete()<br>Remove the current selection from the model.  |
| java.awt.Component[]           | findComponentsInside(int x1,int y1,int x2,int y2)<br>Retreive the list of components located within a rectangular region bounded by the points (x1,y1) and (x2,y2). |
| Vector                         | getAnnotations()<br>This gets fills a vector with all of the <a href="#">annotations</a> it contains.   |
| Vector                         | getComponents(boolean includeConnections)<br>Retrieves a Vector of all the Components in this BeanBox.  |
| <a href="#">DrawnComponent</a> | getDrawnComponent(AbstractComponent comp)<br>Retrieves the DrawnComponent rendering the given component in this BeanBox.  |
| <a href="#">DrawnComponent</a> | getDrawnComponent(int compIdent)<br>Retrieves the DrawnComponent rendering the given component in this BeanBox.   |
| Vector                         | getDrawnComponents()<br>Retrieves a Vector of all the Components in this BeanBox, including <a href="#">Annotations</a> , DrawnComponents and DrawnConnections.     |
| java.awt.Dimension             | getMaximumSize()<br>Return the maximum size of this component.  |
| java.awt.Dimension             | getMinimumSize()<br>Return the minimum size of this component.  |
| <a href="#">AbstractModel</a>  | getModel()<br>Getter for the AbstractModel that contains the <a href="#">com.cafean.client.analysis.ViewComponent</a> this BeanBox represents.                      |
| int                            | getNumSelected()<br>Return the number of selected components.   |
| java.awt.Dimension             | getPanelSize()<br>Return the panelSize parameter from the ZoomablePanel parent class.   |
| java.awt.Dimension             | getPreferredSize()<br>Return the preferred size of this component.  |
| Double                         | getScale()<br>Return the scale parameter from the ZoomablePanel parent class.   |
| java.awt.Component             | getSelected(int i)<br>Return a selected bean.   |
| java.awt.Component[]           | getSelection()<br>Return the array of selected beans.   |

|  |   |
|--|---|
| <code>java.awt.Component[]</code>            | <code>getSelection(Vector selComps)</code><br>Retreive the list of selected components in the drawing order.  |
| <code>java.awt.Rectangle</code>              | <code>getSelectionBounds()</code><br>Return the bounds of selected beans.   |
| <a href="#">ViewComponent</a>                | <code>getViewComponent()</code><br>Retrieves the ViewComponent that corresponds with the DrawnView that contains this BeanBox.  |
| <code>double</code>                          | <code>getWidthScaleFactor()</code><br>Gets this view's width scale factor.  |
| <code>boolean</code>                         | <code>isSelected(java.awt.Component component)</code><br>Returns true if the given component is part of the current selection.  |
| <code>static<br/>java.awt.Component[]</code> | <code>loadComponents(AbstractModel model,com.appt.xdr.PibBlock[] blocks)</code><br>Loads annotations and drawn components from the given array of PibBlock's.                   |
| <code>static<br/>java.awt.Component[]</code> | <code>loadComponents(AbstractModel model,com.appt.xdr.PibBlock[] blocks,boolean loadDrawn)</code><br>Loads annotations and drawn components from the given array of PibBlock's. |
| <code>void</code>                            | <code>minimizeView()</code><br>Minimizes the current view to the minimum possible with the current components.  |
| <code>void</code>                            | <code>organizeSelection()</code><br>Organize the selected components.   |
| <code>void</code>                            | <code>organizeView(Vector drawnComponents,boolean relative)</code><br>Calls <code>AbstractModel.layoutComponents</code> with the given components.                              |
| <code>void</code>                            | <code>paint(java.awt.Graphics g)</code>   |
| <code>void</code>                            | <code>paintImmediately(int x,int y,int w,int h)</code>  |
| <code>void</code>                            | <code>paintImmediately(JComponent component)</code><br>Immediately paints the region of the (scaled) BeanBox that the given component is located.                               |
| <code>void</code>                            | <code>paste()</code>  |
| <code>void</code>                            | <code>print(java.awt.Graphics g)</code>   |
| <code>void</code>                            | <code>redrawSelection()</code><br>Redraw the selected components, or all components in this view if none are selected   |
| <code>void</code>                            | <code>refresh(java.awt.Rectangle r)</code><br>This refreshses a given Rectangle that indicates a dirty region.  |
| <code>void</code>                            | <code>remove(java.awt.Component comp)</code><br>Removes the specified component from this container.  |

|      |  |
|------|--|
| void | <code>remove(int index)</code><br>Removes the component, specified by <code>index</code> , from this container.  |
| void | <code>removeBoxSelectionListener(BoxSelectionListener listener)</code><br>Removes the given listener from the list that is notified when the BeanBox selection set may have changed. |
| void | <code>removeSelectedComponent(Object sel)</code><br>This removes the Object from the list of selected components.  |
| void | <code>renumberSelectedComponents()</code><br>This takes the selected components and has the model renumber their component numbers.  |
| void | <code>repaint(long tm,int x,int y,int width,int height)</code>   |
| void | <code>resetConnections()</code><br>This resets all the connections that connect to the selected components   |
| void | <code>restoreState(Hashtable state)</code><br>Restore the state of the bean from an earlier edit.  |
| void | <code>selectCategory(Category cat)</code><br>This clears the current selection, and adds all the components that are of the given categories subset to the "selected" list.          |
| void | <code>setPaintEnabled(boolean b)</code><br>Enables or disables the repainting of this BeanBox.   |
| void | <code>setScale(Double scale)</code><br>Set the scale parameter.  |
| void | <code>setSelectedComponent(java.awt.Component focus)</code><br>Clear the current selection and add a single object to the selection.   |
| void | <code>showAllConnections()</code><br>This goes through all of the components currently in the view, and tries to render their connections.   |
| void | <code>showAllConnections(boolean undoable)</code><br>This goes through all of the components currently in the view, and tries to render their connections.                           |
| void | <code>storeState(Hashtable state)</code><br>Store the state of the bean to permit undo.  |
| void | <code>updateComponentList(Vector components,boolean organize)</code><br>Synchronizes the component list in this BeanBox to the given list of components.                             |
| void | <code>updateSelection()</code><br>Updates this BeanBox's current selection and notifies MainFrame and the main property view of the new current model and current selection.         |

**Methods inherited from class** `javax.swing.JPanel`

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

#### Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### BeanBox

public **BeanBox**([AbstractModel](#) model)

Constructs a new BeanBox. Each BeanBox has a reference to the AbstractModel that contains the [com.cafean.client.analysis.ViewComponent](#) that constructed the DrawnView.

#### Parameters:

model - the AbstractModel.

## Methods



(continued from last page)

---

## addNotify

```
public void addNotify()
```

Notifies this component that it now has a parent. Overridden here to add cursor update key-listeners to the new parent

---

## getModel

```
public AbstractModel getModel()
```

Getter for the `AbstractModel` that contains the [com.cafean.client.analysis.ViewComponent](#) this `BeanBox` represents.

**Returns:**

the `AbstractModel`.

---

## getAnnotations

```
public Vector getAnnotations()
```

This gets fills a vector with all of the [annotations](#) it contains.

**Returns:**

the Vector containing annotations.

---

## addDrawnConnection

```
public DrawnConnection addDrawnConnection(Connection con)
```

Adds a `DrawnConnection` object to this `BeanBox` for the given connection if and only if both sides of the connection are present in this view.

**Parameters:**

con - the `Connection` that is being added to the view.

**Returns:**

the `DrawnConnection` created from the `Connection`.

---

## getDrawnComponent

```
public DrawnComponent getDrawnComponent(int compIdent)
```

Retrieves the `DrawnComponent` rendering the given component in this `BeanBox`.

**Parameters:**

compIdent - the [com.cafean.client.analysis.GenericObject.getIdent\(\)](#) of the `AbstractComponent` to retrieve a `DrawnComponent` for.

**Returns:**

the `DrawnComponent` requested or null.

---

## getDrawnComponent

```
public DrawnComponent getDrawnComponent(AbstractComponent comp)
```

Retrieves the `DrawnComponent` rendering the given component in this `BeanBox`.

**Parameters:**

comp - the `AbstractComponent` to retrieve a `DrawnComponent` for.

**Returns:**

the `DrawnComponent` requested or null.

---

## remove

```
public void remove(int index)
```

Removes the component, specified by `index`, from this container.

**Parameters:**

`index` - the index of the component to be removed.

**See Also:**

`.add()`

---

## remove

```
public void remove(java.awt.Component comp)
```

Removes the specified component from this container.

**Parameters:**

`comp` - the component to be removed

**See Also:**

`.add()`

`java.awt.Container.remove()`

---

## boundsChanged

```
public void boundsChanged(DrawnComponent component)
```

This is used to inform the BeanBox that the bounds of a `DrawnComponent` may have changed because of a component's internal values being edited. This informs all of the `DrawnConnections` that the component has changed shape.

**Parameters:**

`component` - the `DrawnComponent` whose shape has changed.

---

## connectionPointRemoved

```
public void connectionPointRemoved(DrawnComponent component)
```

This is used to inform the BeanBox that the connecting points of a `DrawnComponent` may have been added or removed because of a component's internal values being edited. This informs all of the `DrawnConnections` that the component's connection points have changed.

**Parameters:**

`component` - the `DrawnComponent` whose shape has changed.

---

## add

```
public java.awt.Component add(java.awt.Component comp)
```

Adds the specified component to this container. If the component is a `DrawnConnection` it will be prepended; otherwise it will be appended. Components are painted in the reverse order that they exist in the BeanBox, thus `DrawnConnection` objects must be first.

**Parameters:**

`comp` - the Component to be added

**Returns:**

the Component argument

**See Also:**

---

(continued from last page)

---

`Container.add()`

---

## addToFront

```
public java.awt.Component addToFront( java.awt.Component comp)
```

Adds the specified component to this container. If the component is a DrawnConnection it will be prepended; otherwise it will be appended. Components are painted in the reverse order that they exist in the BeanBox, thus DrawnConnection objects must be first.

**Parameters:**

comp - the Component to be added

**Returns:**

the Component argument

**See Also:**

.add()

---

## getMinimumSize

```
public java.awt.Dimension getMinimumSize()
```

Return the minimum size of this component. Calculated from the bounds of all child components.

**Returns:**

A dimension object indicating this component's minimum size.

---

## getMaximumSize

```
public java.awt.Dimension getMaximumSize()
```

Return the maximum size of this component. The panelSize component of the zoomable panel containing this beanbox.

**Returns:**

A dimension object indicating this component's maximum size.

---

## getPreferredSize

```
public java.awt.Dimension getPreferredSize()
```

Return the preferred size of this component. This is the panel size scaled by the current scale factor.

**Returns:**

A dimension object indicating this component's preferred size.

---

## getPanelSize

```
public java.awt.Dimension getPanelSize()
```

Return the panelSize parameter from the ZoomablePanel parent class.

**Returns:**

the Dimension size of the ZoomablePanel.

**See Also:**

ZoomablePanel.getPanelSize()

---

## setScale

```
public void setScale(Double scale)
```

---

(continued from last page)

Set the scale parameter. This property is contained in the `ZoomablePanel` parent class.

**Parameters:**

`scale` - The new value for the scale parameter.

---

## getScale

```
public Double getScale()
```

Return the scale parameter from the `ZoomablePanel` parent class.

**Returns:**

the `Double` scale factor of the `ZoomablePanel`.

**See Also:**

`ZoomablePanel.getScale()`

---

## refresh

```
public void refresh(java.awt.Rectangle r)
```

This refreshes a given `Rectangle` that indicates a dirty region.

**Parameters:**

`r` - the `Rectangle` indicating the dirty region to be refreshed.

**See Also:**

`GlassPanel.repaint()`

---

## repaint

```
public void repaint(long tm,  
                    int x,  
                    int y,  
                    int width,  
                    int height)
```

---

## print

```
public void print(java.awt.Graphics g)
```

---

## setPaintEnabled

```
public void setPaintEnabled(boolean b)
```

Enables or disables the repainting of this `BeanBox`.

---

## paint

```
public void paint(java.awt.Graphics g)
```

---

## paintImmediately

```
public void paintImmediately(int x,  
                              int y,  
                              int w,  
                              int h)
```

(continued from last page)

---

## paintImmediately

```
public void paintImmediately(JComponent component)
```

Immediately paints the region of the (scaled) BeanBox that the given component is located.

**See Also:**

```
.paintImmediately(int,int,int,int)()
```

---

## delete

```
public void delete()
```

Remove the current selection from the model.

---

## cut

```
public void cut()
```

Serialize the current selection to the system clipboard, and then remove it from the BeanBox.

---

## copy

```
public boolean copy()
```

Serialize the current selection to the system clipboard.

**Returns:**

true if the copy command is successful.

---

## canPaste

```
public boolean canPaste()
```

The Paste command can only be executed if there are contents on the clipboard, and the Data on the clipboard is appropriate for the ModelEditor.

**Returns:**

true if a paste operation can be performed

---

## canCopy

```
public boolean canCopy()
```

The Copy or cut command can only be performed if there is at least one object selected.

**Returns:**

true if a cut or copy operation can be performed

---

## canDelete

```
public boolean canDelete()
```

The delete operation can only be performed if a copy operation could be performed.

**Returns:**

true if a delete operation can be performed

**See Also:**

```
.canCopy()
```

---

---

## canAlign

```
public boolean canAlign()
```

The align operation can only be performed if two or more components are selected.

### Returns:

true if an align operation can be performed

---

## paste

```
public void paste()
```

---

## loadComponents

```
public static java.awt.Component[] loadComponents(AbstractModel model,  
com.apt.xdr.PibBlock[] blocks)
```

Loads annotations and drawn components from the given array of PibBlock's.

### Parameters:

model - the AbstractModel to load components for  
blocks - the PibBlock[] to load

### See Also:

```
.loadComponents(AbstractModel, PibBlock[], boolean)()
```

---

## loadComponents

```
public static java.awt.Component[] loadComponents(AbstractModel model,  
com.apt.xdr.PibBlock[] blocks,  
boolean loadDrawn)
```

Loads annotations and drawn components from the given array of PibBlock's.

### Parameters:

model - the AbstractModel to load components for  
blocks - the PibBlock[] to load  
loadDrawn - if true, DrawnComponents will be loaded; if false, only annotations will be loaded.

---

## findComponentsInside

```
public java.awt.Component[] findComponentsInside(int x1,  
int y1,  
int x2,  
int y2)
```

Retrieve the list of components located within a rectangular region bounded by the points (x1,y1) and (x2,y2).

### Parameters:

x1 - the int x coordinate of the first corner.  
y1 - the int y coordinate of the first corner.  
x2 - the int x coordinate of the second corner.  
y2 - the int y coordinate of the second corner.

### Returns:

the Component[] of Components that exist inside the given region.

---

(continued from last page)

## getSelection

```
public java.awt.Component[] getSelection(Vector selComps)
```

Retrieve the list of selected components in the drawing order. This makes sure that the BeanBox is not one of the selected components.

**Parameters:**

selComps - the Vector containing the selected components.

**Returns:**

a Component[] containing the selected components in drawing order.

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

**Parameters:**

state - A hash table containing modified parameters.

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

**Parameters:**

state - A hash table containing modified parameters.

---

## setSelectedComponent

```
public void setSelectedComponent(java.awt.Component focus)
```

Clear the current selection and add a single object to the selection.

**Parameters:**

focus - The bean to add to the selection. If null, the current beanbox is used.

---

## removeSelectedComponent

```
public void removeSelectedComponent(Object sel)
```

This removes the Object from the list of selected components. If the Object is a DrawnComponent, it has it's selected flag set false.

**Parameters:**

sel - the Object that will be removed from the selection.

**See Also:**

`DrawnComponent.setSelected()`

---

## addSelectedComponent

```
public void addSelectedComponent(Object selected)
```

Add a single object to the selection. If the object is already selected it will be removed from the selection, otherwise it will be added.

**Parameters:**

selected - the Object that will be added to the selection.

(continued from last page)

**See Also:**`DrawnComponent.setSelected()`

---

## addSelectedComponents

```
public void addSelectedComponents(Object[] sel)
```

Add an array of beans to the current selection. If the bean is already selected, it will be removed from the current selection.

**Parameters:**

`sel` - The beans to add to the selection.

---

## getNumSelected

```
public int getNumSelected()
```

Return the number of selected components.

---

## clearSelection

```
public void clearSelection()
```

Clear the current selection.

---

## getSelected

```
public java.awt.Component getSelected(int i)
```

Return a selected bean.

**Parameters:**

`i` - The index of the bean in the selection.

---

## getSelection

```
public java.awt.Component[] getSelection()
```

Return the array of selected beans.

---

## isSelected

```
public boolean isSelected(java.awt.Component component)
```

Returns true if the given component is part of the current selection.

---

## getSelectionBounds

```
public java.awt.Rectangle getSelectionBounds()
```

Return the bounds of selected beans.

---

## calcSelectionBounds

```
public void calcSelectionBounds()
```

Calculate the bounds of selected beans.

---

## createPopupMenu

```
public JPopupMenu createPopupMenu(java.awt.event.MouseEvent evt)
```

Creates a popup menu suitable for this GlassPanel and the model it is viewing.



---

(continued from last page)

**Returns:**

a JPopupMenu containing appropriate items such as cut, copy paste and view properties.

**See Also:**

`AbstractComponent.getCustomPopupItems()`

---

## renumberSelectedComponents

```
public void renumberSelectedComponents()
```

This takes the selected components and has the model renumber their component numbers.

---

## organizeView

```
public void organizeView(Vector drawnComponents,  
    boolean relative)
```

Calls `AbstractModel.layoutComponents` with the given components.

**Parameters:**

`relative` - if true, the minnum x and y coordinates will be preserved.

---

## getComponents

```
public Vector getComponents(boolean includeConnections)
```

Retrieves a Vector of all the Components in this BeanBox.

**Parameters:**

`includeConnections` - if true, DrawnConnection objects will be included in the list.

**Returns:**

a Vector containing all the Components in the BeanBox.

---

## getDrawnComponents

```
public Vector getDrawnComponents()
```

Retrieves a Vector of all the Components in this BeanBox, including [Annotations](#), DrawnComponents and DrawnConnections.

**Returns:**

a Vector containing all the Components in the BeanBox.

---

## createZoomMenu

```
public JMenu createZoomMenu()
```

Creates a Zoom menu suitable for this view.

**Returns:**

the JMenu containing all of the possible zoom options for this view.

---

## getViewComponent

```
public ViewComponent getViewComponent()
```

Retrieves the ViewComponent that corresponds with the DrawnView that contains this BeanBox.

---

## getWidthScaleFactor

```
public double getWidthScaleFactor()
```

---

(continued from last page)

Gets this view's width scale factor. This factor is intended for use in scaling the diameter or width of components that may have one dimension much larger than the other; such as a 10 meter long, 0.1 meter wide pipe.

---

## updateComponentList

```
public void updateComponentList(Vector components,  
    boolean organize)
```

Synchronizes the component list in this BeanBox to the given list of components. This is used from the InsertComponentDialog.

### Parameters:

`components` - the Vector of components that should be rendered in this view.  
`organize` - true if all the components in the view should be organized.

---

## showAllConnections

```
public void showAllConnections()
```

This goes through all of the components currently in the view, and tries to render their connections.

### See Also:

```
.addDrawnConnection()
```

---

## showAllConnections

```
public void showAllConnections(boolean undoable)
```

This goes through all of the components currently in the view, and tries to render their connections.

### Parameters:

`undoable` - if true, an undoable event will be posted for this change

### See Also:

```
.addDrawnConnection()
```

---

## organizeSelection

```
public void organizeSelection()
```

Organize the selected components. At least two components must be selected for the organize routine to make any sense.

---

## redrawSelection

```
public void redrawSelection()
```

Redraw the selected components, or all components in this view if none are selected

---

## selectCategory

```
public void selectCategory(Category cat)
```

This clears the current selection, and adds all the components that are of the given categories subset to the "selected" list.

---

## resetConnections

```
public void resetConnections()
```

This resets all the connections that connect to the selected components

(continued from last page)

---

## minimizeView

```
public void minimizeView()
```

Minimizes the current view to the minimum possible with the current components. This will translate all the components, until the left most is 20 from the left edge, and the upper most is 20 from the top edge, then it will clip the lower right edges to be 20 from the right most component and bottom most component.

---

## updateSelection

```
public void updateSelection()
```

Updates this BeaBox's current selection and notifies MainFrame and the main property view of the new current model and current selection.

---

## addBoxSelectionListener

```
public void addBoxSelectionListener(BoxSelectionListener listener)
```

Adds the given listener to the list that is notified when the BeanBox selection set may have changed. Note that notification does not ensure that the selection has changed, only that it may have changed.

**Parameters:**

listener - the BoxSelectionListener to add for notification of selection changes

**See Also:**

.removeBoxSelectionListener()

---

## removeBoxSelectionListener

```
public void removeBoxSelectionListener(BoxSelectionListener listener)
```

Removes the given listener from the list that is notified when the BeanBox selection set may have changed.

**Parameters:**

listener - the BoxSelectionListener to remove from the list of selection listeners

**See Also:**

.addBoxSelectionListener()

---

---

## com.cafean.client.ui Interface BoxSelectionListener

---

public interface **BoxSelectionListener**

An interface describing a listener that is to be notified when the selection in a particular BeanBox instance may have changed.

---

### Method Summary

|      |   |
|------|---|
| void | <code>boxSelectionChanged(BeanBox box)</code><br>Notifies this listener that the selection in the given BeanBox may have changed. |
|------|---|

---

### Methods

#### **boxSelectionChanged**

public void **boxSelectionChanged**([BeanBox](#) box)

Notifies this listener that the selection in the given BeanBox may have changed.

**Parameters:**

box – the BeanBox who's selection has changed

## com.cafean.client.ui Class ClientPluginLoader

java.lang.Object

└─com.cafean.client.ui.ClientPluginLoader

All Implemented Interfaces:

PluginListener

public class **ClientPluginLoader**  
extends Object  
implements PluginListener

A delegate used to load client plugins into the given Vector

### Constructor Summary

|        |   |
|--------|---|
| public | <b>ClientPluginLoader</b> (Vector plugins)<br>Creates a new instance of ClientPluginLoader to load client plugins into the given Vectors. |
|--------|---|

### Method Summary

|      |  |
|------|--|
| void | <b>pluginException</b> (Exception ex)<br>Called when there is an exception thrown reading in a plugin.             |
| void | <b>pluginLoaded</b> (Object plugin)<br>Adds the given plugin to the list of plugin's to be loaded in #loadPlugins. |

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### ClientPluginLoader

public **ClientPluginLoader**(Vector plugins)

Creates a new instance of ClientPluginLoader to load client plugins into the given Vectors.

### Methods

#### pluginLoaded

public void **pluginLoaded**(Object plugin)

Adds the given plugin to the list of plugin's to be loaded in #loadPlugins.

(continued from last page)

**Parameters:**

`plugin` - the `ClientCodePlugin` object that has been loaded by the `CodePluginClassLoader`; all other types are ignored.

---

**pluginException**

```
public void pluginException(Exception ex)
```

Called when there is an exception thrown reading in a plugin.

**Parameters:**

`ex` - the `Exception` thrown while reading in a plugin.

## com.cafean.client.ui Class ComponentSelector

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- java.awt.Window
                    |-- java.awt.Dialog
                          |-- javax.swing.JDialog
                                |-- com.cafean.client.ui.ComponentSelector
  
```

public class **ComponentSelector**  
extends JDialog

A dialog for selecting a single component from a list provided upon creation. Usage: `ComponentSelector dlg = new ComponentSelector(...); dlg.setVisible(true); AbstractComponent selection = dlg.getSelection(); dlg.dispose();`

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | ComponentSelector(java.awt.Frame parent, boolean modal, AbstractComponent[] comps)<br>Creates new form ComponentSelector |
| public | ComponentSelector(JDialog parent, boolean modal, AbstractComponent[] comps)<br>Creates new form ComponentSelector        |
| public | ComponentSelector(JDialog parent, boolean modal, Vector comps)<br>Creates new form ComponentSelector                     |
| public | ComponentSelector(java.awt.Frame parent, boolean modal, Vector comps)<br>Creates new form ComponentSelector              |

## Method Summary

|                                   |  |
|-----------------------------------|--|
| <a href="#">AbstractComponent</a> | getSelection()<br>Return the selected component or null if there no selection has been made. |
| void                              | hideComponentNumber()<br>Hides the component number field of this ComponentSelector.         |

|         |  |
|---------|--|
| void    | hideComponentType()<br>Hides the component type field of this ComponentSelector.   |
| boolean | isCanceled()<br>This returns true if the dialog is closed by any other means then by pressing the OK button.   |
| void    | setSelected(AbstractComponent comp)  |
| void    | showCancelButton(boolean b)<br>If this operation is not something that should be cancellable, this function can be called to prevent the cancel button from being displayed. |

**Methods inherited from class javax.swing.JDialog**

getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setJMenuBar, setLayeredPane, setLayout, update

**Methods inherited from class java.awt.Dialog**

addNotify, getAccessibleContext, getTitle, hide, isModal, isResizable, isUndecorated, setModal, setResizable, setTitle, setUndecorated, show

**Methods inherited from class java.awt.Window**

addNotify, addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getAccessibleContext, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, show, toBack, toFront

**Methods inherited from class java.awt.Container**

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

**Methods inherited from class java.awt.Component**



```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### ComponentSelector

```

public ComponentSelector(java.awt.Frame parent,
                        boolean modal,
                        AbstractComponent[] comps)

```

Creates new form ComponentSelector

#### Parameters:

- parent - the parent java.awt.Frame of this dialog.
- modal - true if this dialog should be modal.
- comps - the AbstractComponent[] list to choose from.

(continued from last page)

## ComponentSelector

```
public ComponentSelector(JDialog parent,
                        boolean modal,
                        AbstractComponent[] comps)
```

Creates new form ComponentSelector

### Parameters:

parent - the parent javax.swing.JDialog of this dialog.  
 modal - true if this dialog should be modal.  
 comps - the AbstractComponent[] list to choose from.

---

## ComponentSelector

```
public ComponentSelector(JDialog parent,
                        boolean modal,
                        Vector comps)
```

Creates new form ComponentSelector

### Parameters:

parent - the parent java.awt.Frame of this dialog.  
 modal - true if this dialog should be modal.  
 comps - the Vector of components to choose from.

---

## ComponentSelector

```
public ComponentSelector(java.awt.Frame parent,
                        boolean modal,
                        Vector comps)
```

Creates new form ComponentSelector

### Parameters:

parent - the parent javax.swing.JDialog of this dialog.  
 modal - true if this dialog should be modal.  
 comps - the Vector of components to choose from.

## Methods

### hideComponentType

```
public void hideComponentType()
```

Hides the component type field of this ComponentSelector.

---

### hideComponentNumber

```
public void hideComponentNumber()
```

Hides the component number field of this ComponentSelector.

---

### setSelected

```
public void setSelected(AbstractComponent comp)
```

---

### getSelection

```
public AbstractComponent getSelection()
```

(continued from last page)

Return the selected component or null if there no selection has been made.

**Returns:**

the AbstractComponent selected from the list by the user.

---

## **showCancelButton**

```
public void showCancelButton(boolean b)
```

If this operation is not something that should be cancellable, this function can be called to prevent the cancel button from being displayed.

**Parameters:**

b - boolean value determining if the cancel button should be visible.

---

## **isCanceled**

```
public boolean isCanceled()
```

This returns true if the dialog is closed by any other means then by pressing the OK button.

**Returns:**

false if the OK button was used to close this dialog.

## com.cafean.client.ui Class ConnectingPt

java.lang.Object

└─com.cafean.client.ui.ConnectingPt

### All Implemented Interfaces:

Cloneable

```
public class ConnectingPt
  extends Object
  implements Cloneable
```

The ConnectingPt are the small target and source points on [components](#) that are used to both connect two components using the [connect](#) tool, and to render a connection between two components using a { @link com.cafean.client.ui.DrawnConnection }

## Field Summary

|                  |  |
|------------------|--|
| static final int | CROSSFLOW_DROP_ZONE<br>A connection target for a crossflow connect<br>Value: <b>2</b>                  |
| static final int | INLET_DROP_ZONE<br>Connection target that points into the component.<br>Value: <b>0</b>                |
| static final int | INLET_MOVEABLE_CONNECTOR<br>A connection source that points into the component.<br>Value: <b>3</b>     |
| static final int | INVISIBLE_CONNECTOR<br>A connection point that does not draw on the drawn component<br>Value: <b>6</b> |
| static final int | OUTLET_DROP_ZONE<br>A Connection target that points out from the component<br>Value: <b>1</b>          |
| static final int | OUTLET_MOVEABLE_CONNECTOR<br>A connection source that points out of the component<br>Value: <b>4</b>   |
| static final int | STATIC_CONNECTOR<br>A connection source that cannot be moved<br>Value: <b>5</b>                        |

## Constructor Summary

|        |   |
|--------|---|
| public | ConnectingPt(int type,int linkNum)<br>The default constructor for a ConnectingPt.                     |
| public | ConnectingPt(int type,int linkNum,ConnectionData data)<br>The default constructor for a ConnectingPt. |

## Method Summary

|  |   |
|--|---|
| Object                                     | clone()<br><br>   |
| java.awt.Color[]                           | getBadgeColors()<br>Gets the colors of the Badges for this ConnectingPt   |
| java.awt.Shape[]                           | getBadges()<br>Gets the Badges for this ConnectingPt  |
| <a href="#">ConnectionData</a>             | getConnectionData()<br>Gets the ConnectionData that defines this ConnectingPt's position.   |
| java.awt.Shape                             | getHandle()<br><br>   |
| double                                     | getHandleSize()<br>Gets the size of the handle.   |
| boolean                                    | getNeedsTarget()<br>Returns true if connections started from this ConnectingPt will allow connections to whole components.  |
| Pad  | getPad()<br>This returns the actual drawn location on the panel where a connection can be initiated or completed with the connect tool.   |
| static<br>java.awt.geom.Point2D<br>.Double | getRotatedLocation(double theta,java.awt.geom.Point2D.Double pt,java.awt.geom.Point2D.Double ctr)<br>computes the new location of a point after it has been rotated.  |
| static java.awt.Point                      | getRotatedLocation(double theta,java.awt.Point pt,java.awt.Point ctr)<br>computes the new location of a point after it has been rotated.  |
| int  | getType()<br>Returns the type of this ConnectingPt.   |
| boolean                                    | isConnected()<br>Returns true if there is a Connection to the AbstractComponent that would be drawn to this ConnectingPt existst, even if there is no DrawnConnection for that Connection on this DrawnView |
| boolean                                    | isVisible()<br>Returns true if this ConnectingPt is a connectinPt that can be visible.  |
| void                                       | move(double diffx,double diffy)<br>Moves a ConnectingPt by the distance in each dimension.  |

|         |  |
|---------|--|
| void    | <code>rotate(double theta,double x,double y)</code><br>Rotates a ConnectingPt clockwize by the angle theta, around the point specified by x and y.   |
| void    | <code>setBadgeColors(java.awt.Color[] badgeColors)</code><br>Sets the colors of the Badges for this ConnectingPt   |
| void    | <code>setBadges(java.awt.Shape[] theBadges)</code><br>Sets the Badges for this ConnectingPt  |
| void    | <code>setConnected()</code><br>Should be called if there is a Connection to the AbstractComponent that would be drawn to this ConnectingPt exists, even if there is no DrawnConnection for that Connection on this DrawnView.                                    |
| void    | <code>setConnectPtType(int type)</code><br>Sets the type of this ConnectingPt.   |
| void    | <code>setDisconnected()</code><br>Should be called if there is a Connection to the AbstractComponent that would be drawn to this ConnectingPt has been disconnected.This specifically gets called by the DrawnComponent when a Connection has been disconnected. |
| void    | <code>setHandle(java.awt.Shape theHandle)</code>   |
| void    | <code>setHandleSize(double hsHeight)</code><br>Sets the size of the handle.  |
| void    | <code>setNeedsTarget(boolean val_)</code><br>Sets whether connections started from this ConnectingPt will allow connections to whole components.   |
| void    | <code>setPad(Pad thePad)</code><br>This sets the actual drawn location on the panel where a connection can be initiated or completed with the connect tool.  |
| String  | <code>toString()</code><br>Constructs an informative label for this ConnectingPt, including type, and pad coordinates  |
| boolean | <code>typeIsConnector()</code><br>This determines if this ConnectingPt is of a type that can initiate a connection.  |
| boolean | <code>typeIsCrossflow()</code><br>This determines if this ConnectingPt is the used to represent the outlet of a component.   |
| boolean | <code>typeIsDropZone()</code><br>This determines if this ConnectingPt is of a type that can complete a connection.   |
| boolean | <code>typeIsInlet()</code><br>This determines if this ConnectingPt is the used to represent the inlet of a component.  |
| boolean | <code>typeIsOutlet()</code><br>This determines if this ConnectingPt is the used to represent the outlet of a component.  |

boolean

typeIsStatic()

This determines if this ConnectingPt is of a type that can be moved around.

**Methods inherited from class** java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### INLET\_DROP\_ZONE

```
public static final int INLET_DROP_ZONE
```

Connection target that points into the component.

### OUTLET\_DROP\_ZONE

```
public static final int OUTLET_DROP_ZONE
```

A Connection target that points out from the component

### CROSSFLOW\_DROP\_ZONE

```
public static final int CROSSFLOW_DROP_ZONE
```

A connection target for a crossflow connect

### INLET\_MOVEABLE\_CONNECTOR

```
public static final int INLET_MOVEABLE_CONNECTOR
```

A connection source that points into the component.

### OUTLET\_MOVEABLE\_CONNECTOR

```
public static final int OUTLET_MOVEABLE_CONNECTOR
```

A connection source that points out of the component

### STATIC\_CONNECTOR

```
public static final int STATIC_CONNECTOR
```

A connection source that cannot be moved

### INVISIBLE\_CONNECTOR

```
public static final int INVISIBLE_CONNECTOR
```

A connection point that does not draw on the drawn component

## Constructors

### ConnectingPt

```
public ConnectingPt(int type,
                   int linkNum)
```

(continued from last page)

The default constructor for a `ConnectingPt`.

**Parameters:**

`type` - the enumerated type of `connectingPt`. Acceptable values are.

`#INLET_DROP_ZONE#OUTLET_DROP_ZONE#CROSSFLOW_DROP_ZONE#INLET_MOVEABLE_CONNECTOR#OUTLET_MOVEABLE_CONNECTOR#INVISIBLE_CONNECTOR`  
`linkNum` - the index of this `connectionPoint`.

## ConnectingPt

```
public ConnectingPt(int type,
                    int linkNum,
                    ConnectionData data)
```

The default constructor for a `ConnectingPt`.

**Parameters:**

`type` - the enumerated type of `connectingPt`. Acceptable values are.

`#INLET_DROP_ZONE#OUTLET_DROP_ZONE#CROSSFLOW_DROP_ZONE#INLET_MOVEABLE_CONNECTOR#OUTLET_MOVEABLE_CONNECTOR`  
`linkNum` - the index of this `connectionPoint`.  
`data` - the `ConnectionData` that defines this `ConnectingPt`'s position.

## Methods

### getConnectionData

```
public ConnectionData getConnectionData()
```

Gets the `ConnectionData` that defines this `ConnectingPt`'s position. This data is used by the `connect` tool to determine the details of the connection.

**Returns:**

the `ConnectionData`.

### toString

```
public String toString()
```

Constructs an informative label for this `ConnectingPt`, including type, and pad coordinates

**Returns:**

the `String` for quickly representing this `ConnectingPt`.

**See Also:**

`.getPad()`

### typeIsConnector

```
public boolean typeIsConnector()
```

This determines if this `ConnectingPt` is of a type that can initiate a connection.

**Returns:**

true if the type is one of the following:

`#INLET_MOVEABLE_CONNECTOR#OUTLET_MOVEABLE_CONNECTOR#STATIC_CONNECTOR`



(continued from last page)

---

## typeIsDropZone

```
public boolean typeIsDropZone()
```

This determines if this ConnectingPt is of a type that can complete a connection.

**Returns:**

true if the type is one of the following:

```
#INLET_DROP_ZONE#OUTLET_DROP_ZONE#CROSSFLOW_DROP_ZONE
```

---

## typeIsStatic

```
public boolean typeIsStatic()
```

This determines if this ConnectingPt is of a type that can be moved around.

**Returns:**

true if the type is one of the following:

```
#INLET_DROP_ZONE#OUTLET_DROP_ZONE#CROSSFLOW_DROP_ZONE#STATIC_CONNECTOR
```

---

## typeIsInlet

```
public boolean typeIsInlet()
```

This determines if this ConnectingPt is the used to represent the inlet of a component.

**Returns:**

true if the type is one of the following:

```
#INLET_DROP_ZONE#INLET_MOVEABLE_CONNECTOR
```

---

## clone

```
public Object clone()
```

---

## typeIsOutlet

```
public boolean typeIsOutlet()
```

This determines if this ConnectingPt is the used to represent the outlet of a component.

**Returns:**

true if the type is one of the following:

```
#OUTLET_DROP_ZONE#OUTLET_MOVEABLE_CONNECTOR
```

---

## typeIsCrossflow

```
public boolean typeIsCrossflow()
```

This determines if this ConnectingPt is the used to represent the outlet of a component.

**Returns:**

true if the type is #CROSSFLOW\_DROP\_ZONE.

---

## getType

```
public int getType()
```

Returns the type of this ConnectingPt.

---

(continued from last page)

**Returns:**

one of the following:

```
#INLET_DROP_ZONE#OUTLET_DROP_ZONE#CROSSFLOW_DROP_ZONE#INLET_MOVEABLE_CONNECTOR
#OUTLET_MOVEABLE_CONNECTOR
```

---

**isVisible**

```
public boolean isVisible()
```

Returns true if this ConnectingPt is a connectinPt that can be visible.

**Returns:**

true if the type is #INVISIBLE\_CONNECTOR.

---

**setConnectPtType**

```
public void setConnectPtType(int type)
```

Sets the type of this ConnectingPt.

**Parameters:**

type - the new type of this ConnectingPoint, acceptable values are:

```
#INLET_DROP_ZONE#OUTLET_DROP_ZONE#CROSSFLOW_DROP_ZONE#INLET_MOVEABLE_CONNECTOR
#OUTLET_MOVEABLE_CONNECTOR
```

---

**isConnected**

```
public boolean isConnected()
```

Returns true if there is a Connection to the AbstractComponent that would be drawn to this ConnectingP existst, even if there is no DrawnConnection for that Connection on this DrawnView

**Returns:**

true if this ConnectingPt is currently connected.

---

**setConnected**

```
public void setConnected()
```

Should be called if there is a {@link Connection} to the {@link AbstractComponent} that would be drawn to this ConnectingPt exists, even if there is no {@link DrawnConnection} for that {@link Connection} on this {@link DrawnView}. This specifically gets called by the connect tool after a connection has been completed.

---

**setDisconnected**

```
public void setDisconnected()
```

Should be called if there is a Connection to the AbstractComponent that would be drawn to this ConnectingPt has been disconnected.This specifically gets called by the DrawnComponent when a Connection has been disconnected.

---

**getPad**

```
public Pad getPad()
```

This returns the actual drawn location on the panel where a connection can be initiated or completed with the connect tool.

**Returns:**

the Pad for this ConnectingPt.

(continued from last page)

## setPad

```
public void setPad(Pad thePad)
```

This sets the actual drawn location on the panel where a connection can be initiated or completed with the connect tool.

### Parameters:

thePad - the Pad for this ConnectingPt.

---

## getHandle

```
public java.awt.Shape getHandle()
```

---

## setHandle

```
public void setHandle(java.awt.Shape theHandle)
```

---

## getBadges

```
public java.awt.Shape[] getBadges()
```

Gets the Badges for this ConnectingPt

### Returns:

the Shape[] containing the badges for this ConnectingPt

---

## setBadges

```
public void setBadges(java.awt.Shape[] theBadges)
```

Sets the Badges for this ConnectingPt

### Parameters:

theBadges - the Shape[] containing the badges for this ConnectingPt

---

## setBadgeColors

```
public void setBadgeColors(java.awt.Color[] badgeColors)
```

Sets the colors of the Badges for this ConnectingPt

### Parameters:

badgeColors - the Color[] containing the colors for the badges for this ConnectingPt

---

## getBadgeColors

```
public java.awt.Color[] getBadgeColors()
```

Gets the colors of the Badges for this ConnectingPt

### Returns:

the Color[] containing the colors for the badges for this ConnectingPt

---

## setHandleSize

```
public void setHandleSize(double hsHeight)
```

Sets the size of the handle.

---

(continued from last page)

**Parameters:**

`hsHeight` - the handle size for both width and height.

---

## getHandleSize

```
public double getHandleSize()
```

Gets the size of the handle.

**Returns:**

the handle size for both width and height.

---

## move

```
public void move(double diffx,  
                 double diffy)
```

Moves a `ConnectingPt` by the distance in each dimension. All badges, the Pad, and the Handle are translated by `diffx` and `diffy` ammount.

**Parameters:**

`diffx` - the distance to translate in the X direction.

`diffy` - the distance to translate in the Y direction.

**See Also:**

`AffineTransform.translate()`

---

## rotate

```
public void rotate(double theta,  
                  double x,  
                  double y)
```

Rotates a `ConnectingPt` clockwise by the angle `theta`, around the point specified by `x` and `y`. All badges, the Pad, and the Handle are rotated as well.

**Parameters:**

`theta` - the angular distance to rotate, clockwise in radians.

`x` - the x point to rotate about.

`y` - the y point to rotate about.

**See Also:**

`AffineTransform.rotate()`

---

## getRotatedLocation

```
public static java.awt.Point getRotatedLocation(double theta,  
                                                java.awt.Point pt,  
                                                java.awt.Point ctr)
```

computes the new location of a point after it has been rotated.

**Parameters:**

`theta` - the angle in radians of rotation (remember Java rotates clockwise)

`pt` - the initial location of the Point to be rotated

`ctr` - the Point that is the center of rotation

---

## getRotatedLocation

```
public static java.awt.geom.Point2D.Double getRotatedLocation(double theta,  
                                                              java.awt.geom.Point2D.Double pt,  
                                                              java.awt.geom.Point2D.Double ctr)
```

(continued from last page)

computes the new location of a point after it has been rotated.

**Parameters:**

`theta` - the angle in radians of rotation (remember Java rotates clockwise)  
`pt` - the initial location of the `Point2D#Double` to be rotated  
`ctr` - the `Point2D#Double` that is the center of rotation

---

**getNeedsTarget**

```
public boolean getNeedsTarget()
```

Returns true if connections started from this `ConnectingPt` will allow connections to whole components. If false, connections started from this `ConnectingPt` require a `ConnectingPt` as a target.

---

**setNeedsTarget**

```
public void setNeedsTarget(boolean val_)
```

Sets whether connections started from this `ConnectingPt` will allow connections to whole components. If false, connections started from this `ConnectingPt` require a `ConnectingPt` as a target.

## com.cafean.client.ui

### Class ConnectionSetPanel

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- javax.swing.JPanel
                          |
                          +- com.cafean.client.ui.ConnectionSetPanel

```

#### All Implemented Interfaces:

java.io.Serializable, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
java.io.Serializable, javax.accessibility.Accessible

#### public class **ConnectionSetPanel**

extends JPanel

implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver,  
java.awt.MenuContainer, java.io.Serializable, java.io.Serializable

A panel for displaying a set of connections to a given component.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED,  
WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | ConnectionSetPanel()<br>Creates new form ConnectionsPanel  |
| public | ConnectionSetPanel(AbstractComponent component, Connection[] connections)<br>Creates new form ConnectionsPanel |

## Method Summary

|      |  |
|------|--|
| void | addConnectionSelectionListener(ConnectionSelectionListener listener)<br>Adds a selection listener to the list of listeners in this panel |
| void | clearSelection()<br>Clears the currently selected connections  |

|      |   |
|------|---|
| void | <code>init(AbstractComponent component, Connection[] connections)</code><br>Initializes this panel for use in displaying the given connections                |
| void | <code>refresh(Connection[] connections)</code><br>Refreshes this panel with the given Connection list.  |
| void | <code>removeConnectionSelectionListener(ConnectionSelectionListener listener)</code><br>Removes a selection listener from the list of listeners in this panel |

**Methods inherited from class `javax.swing.JPanel`**

`getAccessibleContext`, `getUI`, `getUIClassID`, `setUI`, `updateUI`

**Methods inherited from class `javax.swing.JComponent`**

`addAncestorListener`, `addNotify`, `addVetoableChangeListener`, `computeVisibleRect`, `contains`, `createToolTip`, `disable`, `enable`, `firePropertyChange`, `firePropertyChange`, `firePropertyChange`, `getAccessibleContext`, `getActionForKeyStroke`, `getActionMap`, `getAlignmentX`, `getAlignmentY`, `getAncestorListeners`, `getAutoscrolls`, `getBorder`, `getBounds`, `getClientProperty`, `getComponentPopupMenu`, `getConditionForKeyStroke`, `getDebugGraphicsOptions`, `getDefaultLocale`, `getFontMetrics`, `getGraphics`, `getHeight`, `getInheritsPopupMenu`, `getInputMap`, `getInputMap`, `getInputVerifier`, `getInsets`, `getInsets`, `getListeners`, `getLocation`, `getMaximumSize`, `getMinimumSize`, `getNextFocusableComponent`, `getPopupLocation`, `getPreferredSize`, `getRegisteredKeyStrokes`, `getRootPane`, `getSize`, `getToolTipLocation`, `getToolTipText`, `getToolTipText`, `getTopLevelAncestor`, `getTransferHandler`, `getUIClassID`, `getVerifyInputWhenFocusTarget`, `getVetoableChangeListeners`, `getVisibleRect`, `getWidth`, `getX`, `getY`, `grabFocus`, `isDoubleBuffered`, `isLightweightComponent`, `isManagingFocus`, `isOpaque`, `isOptimizedDrawingEnabled`, `isPaintingTile`, `isRequestFocusEnabled`, `isValidateRoot`, `paint`, `paintImmediately`, `paintImmediately`, `print`, `printAll`, `putClientProperty`, `registerKeyboardAction`, `registerKeyboardAction`, `removeAncestorListener`, `removeNotify`, `removeVetoableChangeListener`, `repaint`, `repaint`, `requestDefaultFocus`, `requestFocus`, `requestFocus`, `requestFocusInWindow`, `resetKeyboardActions`, `reshape`, `revalidate`, `scrollRectToVisible`, `setActionMap`, `setAlignmentX`, `setAlignmentY`, `setAutoscrolls`, `setBackground`, `setBorder`, `setComponentPopupMenu`, `setDebugGraphicsOptions`, `setDefaultLocale`, `setDoubleBuffered`, `setEnabled`, `setFocusTraversalKeys`, `setFont`, `setForeground`, `setInheritsPopupMenu`, `setInputMap`, `setInputVerifier`, `setMaximumSize`, `setMinimumSize`, `setNextFocusableComponent`, `setOpaque`, `setPreferredSize`, `setRequestFocusEnabled`, `setToolTipText`, `setTransferHandler`, `setVerifyInputWhenFocusTarget`, `setVisible`, `unregisterKeyboardAction`, `update`, `updateUI`

**Methods inherited from class `java.awt.Container`**

`add`, `add`, `add`, `add`, `add`, `addContainerListener`, `addNotify`, `addPropertyChangeListener`, `addPropertyChangeListener`, `applyComponentOrientation`, `areFocusTraversalKeysSet`, `countComponents`, `deliverEvent`, `doLayout`, `findComponentAt`, `findComponentAt`, `getAlignmentX`, `getAlignmentY`, `getComponent`, `getComponentAt`, `getComponentAt`, `getComponentCount`, `getComponents`, `getComponentZOrder`, `getContainerListeners`, `getFocusTraversalKeys`, `getFocusTraversalPolicy`, `getInsets`, `getLayout`, `getListeners`, `getMaximumSize`, `getMinimumSize`, `getMousePosition`, `getPreferredSize`, `insets`, `invalidate`, `isAncestorOf`, `isFocusCycleRoot`, `isFocusCycleRoot`, `isFocusTraversalPolicyProvider`, `isFocusTraversalPolicySet`, `layout`, `list`, `list`, `locate`, `minimumSize`, `paint`, `paintComponents`, `preferredSize`, `print`, `printComponents`, `remove`, `remove`, `removeAll`, `removeContainerListener`, `removeNotify`, `setComponentZOrder`, `setFocusCycleRoot`, `setFocusTraversalKeys`, `setFocusTraversalPolicy`, `setFocusTraversalPolicyProvider`, `setFont`, `setLayout`, `transferFocusBackward`, `transferFocusDownCycle`, `update`, `validate`

**Methods inherited from class `java.awt.Component`**

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### ConnectionSetPanel

```

public ConnectionSetPanel()
    Creates new form ConnectionsPanel

```

### ConnectionSetPanel

```

public ConnectionSetPanel(AbstractComponent component,
    Connection[] connections)
    Creates new form ConnectionsPanel

```

## Methods



(continued from last page)

## **init**

```
public void init(AbstractComponent component,  
                Connection[] connections)  
    Initializes this panel for use in displaying the given connections
```

---

## **clearSelection**

```
public void clearSelection()  
    Clears the currently selected connections
```

---

## **addConnectionSelectionListener**

```
public void addConnectionSelectionListener(ConnectionSelectionListener listener)  
    Adds a selection listener to the list of listeners in this panel
```

---

## **removeConnectionSelectionListener**

```
public void removeConnectionSelectionListener(ConnectionSelectionListener listener)  
    Removes a selection listener from the list of listeners in this panel
```

---

## **refresh**

```
public void refresh(Connection[] connections)  
    Refreshes this panel with the given Connection list.
```

## com.cafean.client.ui Interface DeckWriter

public interface **DeckWriter**

This interface must be implemented by any class used for exporting ASCII decks. This allows for the ModelEditor to export ascii decks indipendently of the plugin.

### Method Summary

|         |   |
|---------|---|
| boolean | <code>performExport(boolean exportCheck)</code><br>This performs the export of an ascii deck. |
|---------|---|

### Methods

#### **performExport**

public boolean **performExport**(boolean exportCheck)

This performs the export of an ascii deck.

##### **Parameters:**

exportCheck - the boolean flag to turn on error checking.

##### **Returns:**

true if the export was successfull

## com.cafean.client.ui Class DrawnComponent

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   └── com.cafean.client.ui.DrawnComponent
```

### All Implemented Interfaces:

[ComponentListener](#), [StateEditable](#), [java.awt.event.MouseMotionListener](#), [java.awt.event.MouseListener](#), [Cloneable](#), [java.io.Serializable](#), [java.awt.MenuContainer](#), [java.awt.image.ImageObserver](#), [java.io.Serializable](#)

### Direct Known Subclasses:

[DrawnViewComponent](#), [DrawnConnection](#), [DrawnUserValue](#)

---

public abstract class **DrawnComponent**

extends [JComponent](#)

implements [java.io.Serializable](#), [java.awt.image.ImageObserver](#), [java.awt.MenuContainer](#), [java.io.Serializable](#), [Cloneable](#), [java.awt.event.MouseListener](#), [java.awt.event.MouseMotionListener](#), [StateEditable](#), [ComponentListener](#)

The DrawnComponent is a renderer for an AbstractComponent. It contains all of the information necessary to draw a representation of an AbstractComponent inside a DrawnView, or a NodeViewPanel.

The DrawnComponent is a ComponentListener on the AbstractComponent that it renders. This allows the DrawnComponent to update itself based on changes made to a component.

A DrawnComponent may be comprised of smaller shapes stored in an array. This is primarily used by Hydraulic Components but is available for any DrawnComponent to use.

---

## Field Summary

|                  |   |
|------------------|---|
| static final int | BOTTOM<br>Oriented towards the bottom of the screen<br>Value: 2 |
| static final int | CENTER<br>Align in the direct center.<br>Value: 5               |
| static final int | CENTER_H<br>align in the center horizontally.<br>Value: 7       |
| static final int | CENTER_V<br>Align in the center vertically.<br>Value: 6         |

|                                  |   |
|----------------------------------|---|
| <code>static final int</code>    | <p>CIRCLE</p> <p>Indicates the Pad for the ConnectingPt should be a Circle</p> <p>Value: <b>0</b></p>                                 |
| <code>static final int</code>    | <p>CROSSHATCH</p> <p>Indicates the Pad for the ConnectingPt should be a Crosshatch</p> <p>Value: <b>4</b></p>                         |
| <code>static final int</code>    | <p>DIAMOND</p> <p>Indicates the Pad for the ConnectingPt should be a Diamond</p> <p>Value: <b>3</b></p>                               |
| <code>static final int</code>    | <p>DOWN</p> <p>Oriented towards the bottom of the screen</p> <p>Value: <b>2</b></p>   |
| <code>static final int</code>    | <p>LEFT</p> <p>Oriented to the left hand of the screen</p> <p>Value: <b>0</b></p>   |
| <code>static final int</code>    | <p>max_positions</p> <p>The number of different orientation enumerations.</p> <p>Value: <b>4</b></p>                                  |
| <code>static final int</code>    | <p>NONE</p> <p>No Pad type is needed for the ConnectingPt</p> <p>Value: <b>-1</b></p>   |
| <code>static final double</code> | <p>PIXELS_P_METER</p> <p>This determines how many pixels are needed to display a single meter in length</p> <p>Value: <b>25.0</b></p> |
| <code>static final int</code>    | <p>RIGHT</p> <p>Oriented to the right-hand of the screen</p> <p>Value: <b>1</b></p>   |
| <code>static final int</code>    | <p>SEGMENT_BOTH</p> <p>Indicates a segment has both an inlet and outlet ConnectingPt.</p> <p>Value: <b>3</b></p>                      |
| <code>static final int</code>    | <p>SEGMENT_INLET</p> <p>Indicates a segment has only an inlet ConnectingPt.</p> <p>Value: <b>1</b></p>                                |
| <code>static final int</code>    | <p>SEGMENT_NONE</p> <p>Indicates a segment has neither an inlet nor an outlet.</p> <p>Value: <b>0</b></p>                             |
| <code>static final int</code>    | <p>SEGMENT_OUTLET</p> <p>Indicates a segment has only an outlet ConnectingPt</p> <p>Value: <b>2</b></p>                               |

|                  |  |
|------------------|--|
| static final int | SEGMENT_SPECIAL<br>Indicates a segment a special meaning.<br>Value: <b>4</b>               |
| static final int | SQUARE<br>Indicates the Pad for the ConnectingPt should be a Crosshatch<br>Value: <b>5</b> |
| static final int | TOP<br>Oriented towards the top of the screen<br>Value: <b>3</b>                           |
| static final int | TRIANGLE<br>Indicates the Pad for the ConnectingPt should be a Triangle<br>Value: <b>2</b> |
| static final int | UP<br>Oriented towards the top of the screen<br>Value: <b>3</b>                            |

**Fields inherited from class** javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

**Fields inherited from class** java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | DrawnComponent(AbstractComponent c)<br>Creates a default DrawnComponent of the provided AbstractComponent centered at (0,0) with an angle of 0 and an orientation of RIGHT |
|--------|--|

## Method Summary

|         |   |
|---------|---|
| void    | addNotify()<br>   |
| boolean | canBeResized()<br>All components can be resized as a default.   |
| void    | Clear()<br>Clears the current DrawnComponent's connections and labels                                   |
| void    | clearLinks()<br><b>Deprecated.</b> DrawnLinks have been converted to DrawnConnections.                  |
| Object  | clone(AbstractComponent cb)<br>Creates a clone of a DrawnComponent, including its links and connections |

|                              |   |
|------------------------------|---|
| void                         | <code>componentChanged(ComponentChangedEvent evt)</code><br><p>When a DrawnComponent's component changes, the drawn component needs to call <code>initDrawing</code> on itself { @inheritDoc }</p>  |
| void                         | <code>componentConnected(Connection con)</code><br><p>This should make sure that the ZoomablePanel tries to add the drawnLink to the view when a connection is established.</p>   |
| void                         | <code>componentDeleted()</code><br><p>When a DrawnComponent's component is deleted, the Drawn component needs to remove itself from the view.</p>   |
| void                         | <code>componentDisconnected(Connection con)</code><br><p>This should make sure that the zoomable panel tries to remove the drawnLink from the view when a connection is removed.</p>  |
| void                         | <code>connectLinks()</code><br><p><b>Deprecated.</b> <i>DrawnLinks are now DrawnConnection objects</i></p>  |
| boolean                      | <code>contains(double x,double y)</code><br><p>Checks whether this component "contains" the specified coordinates where <i>x</i> and <i>y</i> are defined to be relative to the coordinate system of this component.</p>                                |
| boolean                      | <code>contains(int x,int y)</code><br><p>{ @inheritDoc }</p>  |
| boolean                      | <code>contains(java.awt.Point p)</code><br><p>{ @inheritDoc }</p>   |
| java.awt.Shape               | <code>createBorderRegion(java.awt.Shape shape,int face,double shift)</code><br><p>Creates a border shape on the specified face of the given shape</p>   |
| java.awt.Shape               | <code>createCenterShape(java.awt.Shape shape,int face,double shift)</code><br><p>Creates a stripe down the center of a given shape</p>  |
| static void                  | <code>createConnectionPrototypes()</code><br><p>Creates the various archtypes of Connection Shapes to be used at connection points</p>  |
| <a href="#">ConnectingPt</a> | <code>createConnectionPt(double x,double y,double angle,int type,int face,int mark,int link_num,ConnectionData data)</code><br><p>Initializes and creates a new connecting point at the specified location and having the specified characteristics</p> |
| JComponent[]                 | <code>createDisplayBeans(int pixelsPerMeter,double widthScaleFactor,ClassLoader loader)</code><br><p>Creates and configures a set of AbstractDisplayBeans for use in displaying this DrawnComponent in an animatable view.</p>                          |
| JPopupMenu                   | <code>createPopupMenu()</code><br><p>Creates a custom popup context menu for this DrawnComponent that includes an optional Actions menu from sub components.</p>  |
| TemplateEntry                | <code>createTemplateEntry()</code><br><p>Creates a new TemplateEntry for this DrawnComponent that stores all of the location and state data for this DranwComponent.</p>  |

|                                   |   |
|-----------------------------------|---|
| void                              | <code>disconnectAllMyLinks()</code><br><b>Deprecated.</b> <i>DrawnLinks have been converted to DrawnConnections</i>   |
| void                              | <code>draw(java.awt.Graphics2D g,boolean selected)</code><br>Tints the current DrawnComponent according to its state.   |
| void                              | <code>drawLabelStrings(java.awt.Shape obj,java.awt.Graphics2D g,int offset)</code><br>This draws all of the Strings to at the center of the Object.               |
| void                              | <code>flip()</code><br>Flips the orientation of a DrawnComponent to the next one in the given rotation.   |
| <a href="#">BeanBox</a>           | <code>getBeanBox()</code><br>Retrieves the BeanBox that contains this DrawnComponent or null if this DrawnComponent exists outside of a BeanBox.                  |
| int                               | <code>getClockwiseFace(int face)</code><br>Retrieves the face value that is 90 degrees clockwise from the given face.   |
| <a href="#">AbstractComponent</a> | <code>getComponent()</code><br>Each DrawnComponent is a rendering object for a specific AbstractComponent.  |
| int                               | <code>getComponentID()</code><br>Retrieves the ident of this drawn component's component  |
| java.awt.Point                    | <code>getConnectingLocation(ConnectingPt point)</code><br>Returns the absolute screen location of the given ConnectingPt.   |
| <a href="#">ConnectingPt</a>      | <code>getConnectingPt(ConnectionData data)</code><br>Retrieves the ConnectingPt for the given ConnectionData or null if none is found.                            |
| <a href="#">ConnectingPt</a>      | <code>getConnectingPt(int i)</code><br>Returns a ConnectingPt for a given int.  |
| <a href="#">ConnectingPt</a>      | <code>getConnectingPtAt(double x,double y)</code><br>Retrieves the connector at the given coordinates   |
| int                               | <code>getConnectSize()</code><br>Retrieves the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent.      |
| int                               | <code>getCounterFace(int face)</code><br>Retrieves the face value that is 90 degrees counter-clockwise from the given face.                                       |
| int                               | <code>getCrossflowIndex(ConnectingPt cp)</code><br>Gets the number of ConnectingPt components that are crossflow Points that occur before the given ConnectingPt. |
| Action[]                          | <code>getCustomPopupActions()</code><br>Retrieves the curstom popup actions from this drawn component's AbstractComponent.  |
| Vector                            | <code>getCustomPopupItems()</code><br>Returns the custom popup items of this DrawnComponent's target component  |

|                            |  |
|----------------------------|--|
| float                      | getDefaultDrawLength()<br>The Default Draw Length for a DrawnComponent is 50 pixels.   |
| float                      | getDefaultDrawWidth()<br>The Default Draw Width for a DrawnComponent is 50 pixels.   |
| double                     | getDrawAngle()<br>Gets the angle, in radians describing the angle of this DrawnComponent   |
| int                        | getDrawingFace(java.awt.geom.Point2D.Double pt)<br>Returns the face the DrawnComponent is connected to when the object is a source, ie, LEFT RIGHT TOP BOTTOM. |
| java.awt.Shape             | getDrawingObject()<br>Gets the Shape that should be drawn when this DrawnComponent is painted.   |
| int                        | getFaceByAngle(double ang,boolean invert)<br>Determines the appropriate orientation for a given angle.   |
| java.awt.Color             | getFillColor()<br>Returns the color to be used for filling this DrawnComponent's normal shape.   |
| <a href="#">GlassPanel</a> | getGlassPane()<br>Retrieves the GlassPanel from the ZoomablePanel that contains this DrawnComponent.   |
| int                        | getHandleSize()<br>This gets the user defined size of the handles from the user preferences.   |
| static java.awt.Color      | getIndicatorColor()<br>Gets the user preference for the Connection Color.  |
| double                     | getLength()<br>The length of a drawn component is the same as it's height.   |
| double                     | getLenScaleFactor()<br>This gets the Length Scale Factor from the AbstractModel.   |
| float                      | getMaxHeight()<br>This gets the maximum height for this DrawnComponent.  |
| float                      | getMaxWidth()<br>This gets the maximum width for this DrawnComponent.  |
| float                      | getMinWidth()<br>This gets the minimum width for this DrawnComponent.  |
| java.awt.Shape             | getMirrorImageShape(java.awt.Shape shape,int mirrorCode)<br>Creates a mirror image of a given shape s  |
| java.awt.Shape             | getNormalObj()<br>Gets the Shape that is used to drawn the component in the DrawnPanel   |



|                              |   |
|------------------------------|---|
| int                          | getNumberConnections()<br>Retrieves the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent. |
| int                          | getOppositeFace(int face)<br>Retrieves the face value that is the exact opposite of the given face.   |
| int                          | getOrientation()<br>Returns the orientation of the DrawnComponent   |
| JMenu                        | getOrientationMenu()<br>Creates a menu appropriate for selecting the desired orientation for this drawn component.                                    |
| static String                | getOrientationName(int orientation)<br>Retrieves the name of the given orientation.   |
| java.awt.Container           | getParent()<br>This either gets the actual parent of the DrawnComponent, if it exists inside a Container.   |
| java.awt.Dimension           | getPreferredSize()<br>Provides the preferred size for this component.   |
| <a href="#">ConnectingPt</a> | getSelectedConnector(double x,double y)<br>Retrieves the connector at the given coordinates   |
| <a href="#">ConnectingPt</a> | getSelectedDropZone(double x,double y)<br>Retrieves the drop zone at the given coordinates  |
| DrawnSubComponent            | getSubComponentAt(int x,int y)<br>Returns the sub-component at the given coordinates, or null if no sub-component exists at that location.            |
| String                       | getToolTipText()  |
| String                       | getToolTipText(int x,int y)   |
| double                       | getWidthScaleFactor()<br>This gets the width scale factor from the AbstractModel.   |
| double                       | getX_Pos()<br>Gets the X coordinate of the center point of this DrawnComponent.   |
| double                       | getXDistBetweenCPs()<br>This distance is used in the layout algorithm to allow for drawn objects which are not square, such as segmented pipes.       |
| double                       | getXDistBetweenXflowCPs()<br>This distance is used in the layout algorithm to allow for drawn objects which are not square, such as segmented pipes.  |
| double                       | getY_Pos()<br>Gets the Y coordinate of the center point of this DrawnComponent.   |

|                               |  |
|-------------------------------|--|
| <a href="#">ZoomablePanel</a> | <code>getZoomablePanel()</code><br>Retrieves the ZoomablePanel that contains this DrawnComponent or null if this DrawnComponent exists outside of a ZoomablePanel. |
| boolean                       | <code>hasSubComponents()</code><br>Returns true if this DrawnComponent has DrawnSubComponents contained within it  |
| void                          | <code>InitDrawing()</code><br>Initializes and scales this component to prepare it for painting.  |
| boolean                       | <code>isAutoScale()</code><br>Getter for property autoScale that determines whether scaleIt calls scale this Drawn Component.                                      |
| boolean                       | <code>isDrawBadges()</code><br>Getter for property drawBadges.   |
| boolean                       | <code>isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)</code><br>Determines if an object is inside the given rectangle                                  |
| boolean                       | <code>isPlenumShaped()</code><br>Plenums are traditionally shaped differently from other components.   |
| boolean                       | <code>isPosnSet()</code><br>Determines if the DrawnComponent's position has been initialized.  |
| boolean                       | <code>isScalable()</code><br>All components can be scaled as a default.  |
| static boolean                | <code>isSegmentSet(int test,int segment)</code><br>Returns true if the given segment bit set is included in the given test value.                                  |
| boolean                       | <code>isSelected()</code><br>Getter for property selected.   |
| boolean                       | <code>isValveShaped()</code><br>Valves are traditionally shaped differently from other components.   |
| void                          | <code>loadDrawnComponent(com.apl.xdr.PibBlock block)</code><br>This function loads a DrawnComponent from a DrawnComponentRec.                                      |
| void                          | <code>mouseClicked(java.awt.event.MouseEvent e)</code>   |
| void                          | <code>mouseDragged(java.awt.event.MouseEvent evt)</code>   |
| void                          | <code>mouseEntered(java.awt.event.MouseEvent e)</code>   |
| void                          | <code>mouseExited(java.awt.event.MouseEvent e)</code>  |
| void                          | <code>mouseMoved(java.awt.event.MouseEvent e)</code>   |
| void                          | <code>mousePressed(java.awt.event.MouseEvent e)</code>   |

|                |  |
|----------------|--|
| void           | mouseReleased( java.awt.event.MouseEvent e )   |
| void           | moveRel(double x,double y,boolean last)<br>Move the drawing relative to its previous position  |
| void           | moveTo(double x,double y,boolean last)<br>Move the drawing so the center is the specified position   |
| void           | paint( java.awt.Graphics g )<br>Paints this component.   |
| void           | paintComponent( java.awt.Graphics g )<br>Tints the current DrawnComponent according to its state.  |
| void           | print( java.awt.Graphics g )<br>Prints this component.   |
| void           | readTemplateEntry( TemplateEntry entry )<br>Sets the data on this DrawnComponent from a TemplateEntry read in from a view template file.   |
| void           | removeNotify()   |
| void           | repositionLinks()<br>Adjusts the position of all links coming out of an object   |
| void           | repositionLinks(boolean last)<br><b>Deprecated.</b> <i>DrawnLinks have been converted to DrawConnections.</i>  |
| void           | resetPosition()<br>resetPosition can be used to set the correct positions of components when they become visible, such as when the display layer is changed.   |
| void           | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.   |
| java.awt.Shape | rotateTo(double theta, java.awt.Shape s, java.awt.geom.Point2D.Double pt, Vector ConnectPts)<br>Rotate the specified shape around a given point by the given angle.                                  |
| java.awt.Shape | rotateTo(double theta, java.awt.Shape s, java.awt.Point pt, Vector ConnectPts)<br>Rotate the specified shape around a given point by the given angle.  |
| boolean        | scaleIt()<br>Scale the drawing so that it reflects the true relative size  |
| void           | setAutoScale(boolean autoScale)<br>Setter for property autoScale that determines whether scaleIt calls scale this Drawn Component.   |
| void           | setBackupComponent( AbstractComponent backup )<br>When a DrawnComponent is rendering a cloned object for displaying changes, the backup component is the source of external data for that component. |

|         |   |
|---------|---|
| void    | <code>setBounds(int x,int y,int width,int height)</code><br>This is used to set the current scale factors on a component when the bounds are changed by the user.           |
| void    | <code>setComponent(AbstractComponent comp)</code><br>This sets the AbstractComponent that is being rendered by this DrawnComponent.   |
| void    | <code>setDrawAngle()</code><br>sets the drawAngle value depending on the DrawnComponent's current orientation   |
| void    | <code>setDrawBadges(boolean drawBadges)</code><br>Setter for property drawBadges.   |
| void    | <code>setDrawHeight(double l)</code><br>Sets the drawing height of the DrawnComponent.  |
| void    | <code>setDrawWidth(double w)</code><br>Sets the drawing width of the DrawnComponent.  |
| void    | <code>setEqualTo(DrawnComponent dc)</code><br>Sets the position and orientation of the current DrawnComponent to be the same as the DrawnComponent provided as an argument. |
| void    | <code>setLabelString(String str,int index)</code><br>Adds or replaces a String in the labels array.   |
| void    | <code>setLenScaleFactor(double factor)</code>   |
| void    | <code>setOrientation(int orientation)</code><br>Sets the orientation of the current DrawnComponent to the specified orientation.  |
| void    | <code>setOrientationByAngle(double a)</code><br>Sets the orientation by an angle measure, in radians  |
| void    | <code>setParent(java.awt.Container parent)</code><br>Setter for the parent value of this DrawnComponent.  |
| void    | <code>setSelected(boolean selected)</code><br>Setter for property selected.   |
| boolean | <code>setSizeTo(double len,double wid)</code><br>Resets the drawing size to the given dimension and then reinitializes the DrawnComponent.                                  |
| void    | <code>setWidthScaleFactor(double factor)</code>   |
| void    | <code>setX_Pos(double position)</code><br>Sets the location of this DrawnComponent's center to the given position in the X dimension.                                       |
| void    | <code>setY_Pos(double position)</code><br>Sets the location of this DrawnComponent's center to the given position in the Y dimension.                                       |

|                       |   |
|-----------------------|---|
| boolean               | showConnections()<br><br>   |
| com.appt.xdr.PibBlock | store(int viewNum)<br><br>This function returns a PibBlock for a drawn component.   |
| void                  | storeState(Hashtable state)<br><br>Store the state of the bean to permit undo.  |
| String                | toString()<br><br>The String produced is based on the toString of the component.  |
| java.awt.Point        | translateConnectionToScreen(ConnectingPt pt)<br><br>This converts the center of a passed connecting Point into the coordinates on the zoomable panel. |
| java.awt.Point        | translatePointToScreen(java.awt.Point point)<br><br>This converts a Point from local coordinates into coordinates on the zoomable panel.              |

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate



(continued from last page)

---

## TRIANGLE

```
public static final int TRIANGLE
```

Indicates the Pad for the ConnectingPt should be a Triangle

---

## DIAMOND

```
public static final int DIAMOND
```

Indicates the Pad for the ConnectingPt should be a Diamond

---

## CROSSHATCH

```
public static final int CROSSHATCH
```

Indicates the Pad for the ConnectingPt should be a Crosshatch

---

## SQUARE

```
public static final int SQUARE
```

Indicates the Pad for the ConnectingPt should be a Crosshatch

---

## SEGMENT\_NONE

```
public static final int SEGMENT_NONE
```

Indicates a segment has neither an inlet nor an outlet.

---

## SEGMENT\_INLET

```
public static final int SEGMENT_INLET
```

Indicates a segment has only an inlet ConnectingPt.

---

## SEGMENT\_OUTLET

```
public static final int SEGMENT_OUTLET
```

Indicates a segment has only an outlet ConnectingPt

---

## SEGMENT\_BOTH

```
public static final int SEGMENT_BOTH
```

Indicates a segment has both an inlet and outlet ConnectingPt.

---

## SEGMENT\_SPECIAL

```
public static final int SEGMENT_SPECIAL
```

Indicates a segment a special meaning.

---

## LEFT

```
public static final int LEFT
```

Oriented to the left hand of the screen

---

## RIGHT

```
public static final int RIGHT
```

Oriented to the right-hand of the screen

---

---

## DOWN

```
public static final int DOWN
```

Oriented towards the bottom of the screen

---

## BOTTOM

```
public static final int BOTTOM
```

Oriented towards the bottom of the screen

---

## TOP

```
public static final int TOP
```

Oriented towards the top of the screen

---

## UP

```
public static final int UP
```

Oriented towards the top of the screen

---

## max\_positions

```
public static final int max_positions
```

The number of different orientation enumerations.

---

## CENTER

```
public static final int CENTER
```

Align in the direct center.

---

## CENTER\_V

```
public static final int CENTER_V
```

Align in the center vertically.

---

## CENTER\_H

```
public static final int CENTER_H
```

align in the center horizontally.

---

## PIXELS\_P\_METER

```
public static final double PIXELS_P_METER
```

This determines how many pixels are needed to display a single meter in length

---

## Constructors

### DrawnComponent

```
public DrawnComponent(AbstractComponent c)
```

Creates a default DrawnComponent of the provided AbstractComponent centered at (0,0) with an angle of 0 and an orientation of RIGHT

---



(continued from last page)

**Parameters:**

c - the AbstractComponent of the component to be modeled

## Methods

### isSegmentSet

```
public static boolean isSegmentSet(int test,
                                     int segment)
```

Returns true if the given segment bit set is included in the given test value.

**Parameters:**

test - the bitset to check the segment mask against  
segment - the SEGMENT\_\* mask to check against test

### setBackupComponent

```
public void setBackupComponent(AbstractComponent backup)
```

When a DrawnComponent is rendering a cloned object for displaying changes, the backup component is the source of external data for that component. For example, the Backup component is used to display externally defined edge angles.

**Parameters:**

backup - the AbstractComponent that contains the original component data.

### getOrientationName

```
public static String getOrientationName(int orientation)
```

Retrieves the name of the given orientation.

**Parameters:**

orientation - the enumeration orientation type.

**Returns:**

the String that best describes that orientation.

### InitDrawing

```
public void InitDrawing()
```

Initializes and scales this component to prepare it for painting. ( Not cell length, just length ) All the Connections of the Component are examined. If the ConnectionData from the Connection has a ConnectingPt, that ConnectingPt is set selected. DrawnComponent derivatives should call this as the last step in their InitDrawing method.

### createDisplayBeans

```
public JComponent[] createDisplayBeans(int pixelsPerMeter,
                                         double widthScaleFactor,
                                         ClassLoader loader)
```

Creates and configures a set of AbstractDisplayBeans for use in displaying this DrawnComponent in an animatable view. Either a set of beans or a single bean may be returned. If multiple beans are returned they should be positioned appropriately in relation to one another. Note: The base implementation returns null.

**Returns:**

a javax.swing.JComponent[] array or null if no beans are available for this DrawnComponent type

(continued from last page)

---

## getIndicatorColor

```
public static java.awt.Color getIndicatorColor()
```

Gets the user preference for the Connection Color.

**Returns:**

the Color selected by the user for the Connections

---

## getLenScaleFactor

```
public double getLenScaleFactor()
```

This gets the Length Scale Factor from the AbstractModel.

**Returns:**

the double length scale factor from the AbstractModel.

**See Also:**

```
AbstractModel.getLenScaleFactor()
```

---

## getWidthScaleFactor

```
public double getWidthScaleFactor()
```

This gets the width scale factor from the AbstractModel.

**Returns:**

the double width scale factor from the AbstractModel.

**See Also:**

```
AbstractModel.getLenScaleFactor()
```

---

## setLenScaleFactor

```
public void setLenScaleFactor(double factor)
```

---

## setWidthScaleFactor

```
public void setWidthScaleFactor(double factor)
```

---

## isValveShaped

```
public boolean isValveShaped()
```

Valves are traditionally shaped differently from other components. This flag indicates that this DrawnComponent is drawn like a valve. This is used by the Organizer.

**Returns:**

false as a default.

---

## isPlenumShaped

```
public boolean isPlenumShaped()
```

Plenums are traditionally shaped differently from other components. This flag indicates that this DrawnComponent is drawn like a plenum. This is used by the Organizer.

**Returns:**

(continued from last page)

false as a default.

---

## createConnectionPrototypes

```
public static void createConnectionPrototypes()
```

Creates the various archtypes of Connection Shapes to be used at connection points

---

## setBounds

```
public void setBounds(int x,  
    int y,  
    int width,  
    int height)
```

This is used to set the current scale factors on a component when the bounds are changed by the user. If autoscale is on and the model preference for autoscaling components is on the scale factor is set to 1, otherwise it is calculated based on the minimum bounds of the component.

### Parameters:

- x - the upper left corner X position.
  - y - the upper left corner Y position.
  - width - the width of the bounds.
  - height - the height of the bounds. {@inheritDoc}
- 

## getNormalObj

```
public java.awt.Shape getNormalObj()
```

Gets the Shape that is used to drawn the component in the DrawnPanel

### Returns:

Shape the Shape rendered by InitDrawing.

---

## toString

```
public String toString()
```

The String produced is based on the toString of the component.

### Returns:

the String for quickly identifying this Drawing

---

## clone

```
public Object clone(AbstractComponent cb)
```

Creates a clone of a DrawnComponent, including its links and connections

### Parameters:

- cb - the AbstractComponent of the DrawnComponent that is being cloned

### Returns:

o an Object representing the clone of the DrawnComponent

---

(continued from last page)

## createConnectionPt

```
public ConnectingPt createConnectionPt(double x,
    double y,
    double angle,
    int type,
    int face,
    int mark,
    int link_num,
    ConnectionData data)
```

Initializes and creates a new connecting point at the specified location and having the specified characteristics

### Parameters:

`data` - the ConnectionData that defines the new ConnectingPt actual connection information.  
`x` - the x-value of the coordinate of the desired connection point  
`y` - the y-value of the coordinate of the desired connection point  
`angle` - the desired drawn angle, in radians, of the shape which will be drawn at this connection point  
`type` - the type of ConnectingPt this is to represent. Should be one of the following:  
 ConnectingPt.INLET\_DROP\_ZONEConnectingPt.OUTLET\_DROP\_ZONEConnectingPt.CROSSFLOW\_DROP\_ZONEConnectingPt.INLET\_MOVEABLE\_CONNECTORConnectingPt.OUTLET\_MOVEABLE\_CONNECTORConnectingPt.STATIC\_CONNECTORConnectingPt.INVISIBLE\_CONNECTOR  
`face` - the face on which the point is to be drawn  
`mark` - the shape to be drawn at this connecting point. Should be one of the following values: (usually for INLET\_MOVEABLE\_CONNECTOR and OUTLET\_MOVEABLE\_CONNECTOR)(usually for INLET\_DROP\_ZONE and OUTLET\_DROP\_ZONE) DrawnComponent.NONEDrawnComponent.CIRCLEDrawnComponent.TRIANGLE DrawnComponent.DIAMONDDrawnComponent.CROSSHATCH  
`link_num` - the number of the link associated with this point; -1 if there is no link associated with this point

### Returns:

The ConnectingPt that has been created.

---

## setEqualTo

```
public void setEqualTo(DrawnComponent dc)
```

Sets the position and orientation of the current DrawnComponent to be the same as the DrawnComponent provided as an argument. The connections and the owner composite base are not changed.

### Parameters:

`dc` - a DrawnComponent to which the current DrawnComponent's position and orientation are to be set

---

## Clear

```
public void Clear()
```

Clears the current DrawnComponent's connections and labels

---

## clearLinks

```
public void clearLinks()
```

**Deprecated.** *DrawnLinks have been converted to DrawnConnections.*

Clears all of the DrawnLinks off of this DrawnComponent

---

## getDefaultDrawWidth

```
public float getDefaultDrawWidth()
```

The Default Draw Width for a DrawnComponent is 50 pixels.

### Returns:

50

## getDefaultDrawLength

```
public float getDefaultDrawLength()
```

The Default Draw Length for a DrawnComponent is 50 pixels.

**Returns:**

50

---

## getLength

```
public double getLength()
```

The length of a drawn component is the same as it's height.

**Returns:**

the length of this drawn component

**See Also:**

`.getHeight()`

---

## getDrawAngle

```
public double getDrawAngle()
```

Gets the angle, in radians describing the angle of this DrawnComponent

**Returns:**

the angle of this DrawnComponent.

---

## getDrawingObject

```
public java.awt.Shape getDrawingObject()
```

Gets the Shape that should be drawn when this DrawnComponent is painted. This defaults to the current normalObject

**Returns:**

the Shape for drawing.

---

## connectLinks

```
public void connectLinks()
```

**Deprecated.** *DrawnLinks are now DrawnConnection objects*

This reconnects the DrawnLinks of this DrawnComponent

---

## setDrawAngle

```
public void setDrawAngle()
```

sets the drawAngle value depending on the DrawnComponent's current orientation

---

## getToolTipText

```
public String getToolTipText()
```

---

(continued from last page)

---

## getToolTipText

```
public String getToolTipText(int x,  
                             int y)
```

---

## paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

Tints the current DrawnComponent according to its state. Also draws the normalObj of the DrawnComponent, as well as the component's labels, connection points, and links

**Parameters:**

g - the Graphics2D object which will do the painting

---

## paint

```
public void paint(java.awt.Graphics g)
```

Paints this component. Overridden here to prevent failures in drawn component painting code to cause the UI to become unresponsive.

---

## print

```
public void print(java.awt.Graphics g)
```

Prints this component. Overridden here to prevent failures in drawn component painting code to cause the UI to become unresponsive.

---

## getFillColor

```
public java.awt.Color getFillColor()
```

Returns the color to be used for filling this DrawnComponent's normal shape.

**Returns:**

a java.awt.Color object appropriate for filling this component.

---

## draw

```
public void draw(java.awt.Graphics2D g,  
                boolean selected)
```

Tints the current DrawnComponent according to its state. Also draws the normalObj of the DrawnComponent, as well as the component's labels, connection points, and links

**Parameters:**

g - the Graphics2D object which will do the painting

selected - indicates whether the current DrawnComponent is currently selected

---

## drawLabelStrings

```
public void drawLabelStrings(java.awt.Shape obj,  
                             java.awt.Graphics2D g,  
                             int offset)
```

This draws all of the Strings to at the center of the Object.

**Parameters:**

obj - the Shape that is being drawn.

g - the Graphics2D object.

offset - the vertical offset for the strings.

---

## setSizeTo

```
public boolean setSizeTo(double len,  
    double wid)
```

Resets the drawing size to the given dimension and then reinitializes the DrawnComponent.

**Parameters:**

len - the new length in pixels.  
wid - the new width in pixels

**Returns:**

true if the size given is valid.

**See Also:**

.InitDrawing()

---

## getXDistBetweenCPs

```
public double getXDistBetweenCPs()
```

This distance is used in the layout algorithm to allow for drawn objects which are not square, such as segmented pipes. The absolute value of the distance is returned to allow for the first layout pass when the object may or may not have been flipped to its final orientation yet.

**Returns:**

The distance in pixels between centers of the first and second ConnectingPts.

---

## getXDistBetweenXflowCPs

```
public double getXDistBetweenXflowCPs()
```

This distance is used in the layout algorithm to allow for drawn objects which are not square, such as segmented pipes. The absolute value of the distance is returned to allow for the first layout pass when the object may or may not have been flipped to its final orientation yet.

**Returns:**

The distance in pixels between the third and fourth ConnectingPts.

---

## scaleIt

```
public boolean scaleIt()
```

Scale the drawing so that it reflects the true relative size

**Returns:**

true if the component is scaled.

---

## canBeResized

```
public boolean canBeResized()
```

All components can be resized as a default. Override this to return false if this comp should not be resizable.

**Returns:**

true if this DrawnComponent can be scaled.

---

## isScalable

```
public boolean isScalable()
```

All components can be scaled as a default. Override this to return false if this comp should not be scalable.

---

(continued from last page)

**Returns:**

true if the DrawnComponent can be scaled.

---

**moveTo**

```
public void moveTo(double x,  
                  double y,  
                  boolean last)
```

Move the drawing so the center is the specified position

**Parameters:**

**x** - the new center x position  
**y** - the new center y position  
**last** - false while moving, true on the final move

---

**moveRel**

```
public void moveRel(double x,  
                   double y,  
                   boolean last)
```

Move the drawing relative to its previous position

**Parameters:**

**x** - the change in x position in pixels.  
**y** - the change in y position in pixels  
**last** - true when this move is the final movement.

---

**resetPosition**

```
public void resetPosition()
```

resetPosition can be used to set the correct positions of components when they become visible, such as when the display layer is changed. It should be overloaded by classes that need it. In the base class it does nothing.

---

**rotateTo**

```
public java.awt.Shape rotateTo(double theta,  
                                java.awt.Shape s,  
                                java.awt.Point pt,  
                                Vector ConnectPts)
```

Rotate the specified shape around a given point by the given angle.

**Parameters:**

**theta** - the angular distance to rotate in radians.  
**s** - the Shape to be rotated  
**pt** - the Point around which the shape is rotated  
**ConnectPts** - the vector of ConnectingPt that must also be moved.

**Returns:**

the Shape after it has been rotated.

---

**rotateTo**

```
public java.awt.Shape rotateTo(double theta,  
                                java.awt.Shape s,  
                                java.awt.geom.Point2D.Double pt,  
                                Vector ConnectPts)
```

Rotate the specified shape around a given point by the given angle.

---



(continued from last page)

**Parameters:**

theta - the angular distance to rotate in radians.  
s - the Shape to be rotated  
pt - the Point2D.Double around which the shape is rotated  
ConnectPts - the vector of ConnectingPt that must also be moved.

**Returns:**

the Shape after it has been rotated.

---

## getMirrorImageShape

```
public java.awt.Shape getMirrorImageShape(java.awt.Shape shape,  
int mirrorCode)
```

Creates a mirror image of a given shape s

**Parameters:**

shape - the Shape to mirror  
mirrorCode - the direction to do the mirroring; same as orientation flags: 0 or 1 for left/right or 2 or 3 for top/bottom

**Returns:**

the mirrored Shape

---

## createBorderRegion

```
public java.awt.Shape createBorderRegion(java.awt.Shape shape,  
int face,  
double shift)
```

Creates a border shape on the specified face of the given shape

**Parameters:**

shape - the shape which will have a border added  
face - the face to add the border to  
shift - width of border (pixels)

**Returns:**

the Shape representing the border region

---

## createCenterShape

```
public java.awt.Shape createCenterShape(java.awt.Shape shape,  
int face,  
double shift)
```

Creates a stripe down the center of a given shape

**Parameters:**

shape - the shape which will have the center strip added  
face - the face to add the border to  
shift - width of border (pixels)

**Returns:**

the center Shape

---

## isObjectInsideBounds

```
public boolean isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)
```

Determines if an object is inside the given rectangle

(continued from last page)

**Parameters:**

`rect` - the bounding rectangle

**Returns:**

true if the object is inside the rectangle false otherwise.

---

**contains**

```
public boolean contains(java.awt.Point p)
```

---

**contains**

```
public boolean contains(int x,  
                        int y)
```

---

**contains**

```
public boolean contains(double x,  
                        double y)
```

Checks whether this component "contains" the specified coordinates where `x` and `y` are defined to be relative to the coordinate system of this component. Note that `getBounds()` includes the position of this component within the BeanBox where the given `x` and `y` should **not**.

**Parameters:**

`x` - the x coordinate  
`y` - the y coordinate

**Returns:**

true if the given coordinates fall within the bounds of this DrawnComponent

---

**getConnectingPtAt**

```
public ConnectingPt getConnectingPtAt(double x,  
                                       double y)
```

Retrieves the connector at the given coordinates

**Parameters:**

`x` - the x position  
`y` - the y position

**Returns:**

the `ConnectingPt` at the given coordinates that is a connector

---

**getSelectedConnector**

```
public ConnectingPt getSelectedConnector(double x,  
                                       double y)
```

Retrieves the connector at the given coordinates

**Parameters:**

`x` - the x position  
`y` - the y position

**Returns:**

(continued from last page)

the ConnectingPt at the given coordinates that is a connector

---

## getSelectedDropZone

```
public ConnectingPt getSelectedDropZone(double x,  
double y)
```

Retrieves the drop zone at the given coordinates

### Parameters:

x - the x position  
y - the y position

### Returns:

the ConnectingPt at the given coordinates that is a drop zone

---

## getCrossflowIndex

```
public int getCrossflowIndex(ConnectingPt cp)
```

Gets the number of ConnectingPt components that are crossflow Points that occur before the given ConnectingPt. Returns -1 if the ConnectingPt is not found.

### Parameters:

cp - the ConnectingPoint.

### Returns:

the number of crossflow ConnectingPts that were created before the given point.

---

## repositionLinks

```
public void repositionLinks()
```

Adjusts the position of all links coming out of an object

---

## repositionLinks

```
public void repositionLinks(boolean last)
```

**Deprecated.** *DrawnLinks have been converted to DrawConnections.*

Adjusts the position of all links coming out of an object

### Parameters:

last - true if this was the final move.

---

## getDrawingFace

```
public int getDrawingFace(java.awt.geom.Point2D.Double pt)
```

Returns the face the DrawnComponent is connected to when the object is a source, ie, LEFT RIGHT TOP BOTTOM. Used for drawing a component on the main canvas.

### Parameters:

pt - the argument point

### Returns:

an int representing the face

(continued from last page)

---

## flip

```
public void flip()
```

Flips the orientation of a DrawnComponent to the next one in the given rotation. #LEFT#RIGHT#DOWN#UP

---

## getOrientation

```
public int getOrientation()
```

Returns the orientation of the DrawnComponent

**Returns:**

The enumeration of the current orientation of this DrawnComponent. Acceptable values are:

---

## setOrientation

```
public void setOrientation(int orientation)
```

Sets the orientation of the current DrawnComponent to the specified orientation. If the int specified is invalid, the orientation will be set to #RIGHT.

**Parameters:**

orientation - the int enumeration for the orientation.

**See Also:**

.getOrientation()

---

## setOrientationByAngle

```
public void setOrientationByAngle(double a)
```

Sets the orientation by an angle measure, in radians

**Parameters:**

a - the angle measure, in radians

---

## getFaceByAngle

```
public int getFaceByAngle(double ang,  
                           boolean invert)
```

Determines the appropriate orientation for a given angle. This is used to determine the angle of a Pad facing for a ConnectingPt.

**Parameters:**

angle - The angle in radians that is in question.  
invert - Will return the opposite face if this is true.

**Returns:**

Calculates which face the angle is closest to representing. Possible values are:

#BOTTOM#TOP#RIGHT#LEFT

---

## getOppositeFace

```
public int getOppositeFace(int face)
```

Retrieves the face value that is the exact opposite of the given face.

**Parameters:**

---

(continued from last page)

face - the face enumeration.

**Returns:**

the oposite face. Possible values are:

#BOTTOM#TOP#RIGHT#LEFT

---

## getCounterFace

```
public int getCounterFace(int face)
```

Retrieves the face value that is 90 degrees counter-clockwise from the given face.

**Parameters:**

face - the face enumeration.

**Returns:**

the face 90 degrees counter-clockwise from the given face. Possible values are:

#BOTTOM#TOP#RIGHT#LEFT

---

## getClockwiseFace

```
public int getClockwiseFace(int face)
```

Retrieves the face value that is 90 degrees clockwise from the given face.

**Parameters:**

face - the face enumeration.

**Returns:**

the face 90 degrees clockwise from the given face. Possible values are:

#BOTTOM#TOP#RIGHT#LEFT

---

## getMaxWidth

```
public float getMaxWidth()
```

This gets the maximum width for this DrawnComponent.

**Returns:**

the maximum width of this drawn component.

---

## getMaxHeight

```
public float getMaxHeight()
```

This gets the maximum height for this DrawnComponent.

**Returns:**

the maximum height of this drawn component.

---

## getMinWidth

```
public float getMinWidth()
```

This gets the minimum width for this DrawnComponent.

**Returns:**

the minimum width of this drawn component.

---

(continued from last page)

---

## getComponent

```
public AbstractComponent getComponent()
```

Each DrawnComponent is a rendering object for a specific AbstractComponent. This function is used to gain access to the component that is being rendered.

**Returns:**

the AbstractComponent associated with this DrawnComponent.

---

## setComponent

```
public void setComponent(AbstractComponent comp)
```

This sets the AbstractComponent that is being rendered by this DrawnComponent. This component must be able to be rendered by this DrawnComponent.

**Parameters:**

comp - the AbstractComponent to be rendered.

---

## getComponentID

```
public int getComponentID()
```

Retrieves the ident of this drawn component's component

**Returns:**

the primary key of the component.

---

## getX\_Pos

```
public double getX_Pos()
```

Gets the X coordinate of the center point of this DrawnComponent.

**Returns:**

the X coordinate of the center of the bounds.

---

## getY\_Pos

```
public double getY_Pos()
```

Gets the Y coordinate of the center point of this DrawnComponent.

**Returns:**

the Y coordinate of the center of the bounds.

---

## setX\_Pos

```
public void setX_Pos(double position)
```

Sets the location of this DrawnComponent's center to the given position in the X dimension.

**Parameters:**

position - the new x coordinate for the center of this DrawnComponent.

---

## setY\_Pos

```
public void setY_Pos(double position)
```

Sets the location of this DrawnComponent's center to the given position in the Y dimension.

**Parameters:**

(continued from last page)

`position` - the new x coordinate for the center of this DrawnComponent.

---

## setDrawWidth

```
public void setDrawWidth(double w)
```

Sets the drawing width of the DrawnComponent. If the specified width is invalid, the drawing width will be set to the default drawing width.

**Parameters:**

`w` - the requested drawing width

---

## setDrawHeight

```
public void setDrawHeight(double l)
```

Sets the drawing height of the DrawnComponent. If the specified length is invalid, the drawing height will be set to the default drawing height.

**Parameters:**

`l` - the requested drawing height

---

## disconnectAllMyLinks

```
public void disconnectAllMyLinks()
```

**Deprecated.** *DrawnLinks have been converted to DrawnConnections*

Disconnects all DrawnLinks of the current DrawnComponent

---

## getHandleSize

```
public int getHandleSize()
```

This gets the user defined size of the handles from the user preferences.

**Returns:**

the size of the Resize Handle specified by the user in the Snap Preferences section

---

## getConnectSize

```
public int getConnectSize()
```

Retrieves the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent. Note: Do not confuse with `getNumberConnections()`

**Returns:**

the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent. Checks the value specified by the user in the Snap Preferences set.

---

## getNumberConnections

```
public int getNumberConnections()
```

Retrieves the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent.

---

## getConnectingPt

```
public ConnectingPt getConnectingPt(ConnectionData data)
```

Retrieves the ConnectingPt for the given ConnectionData or null if none is found. This is used to find which ConnectingPt a DrawnConnection should connect to.

(continued from last page)

**Parameters:**

`data` - the `ConnectionData` to find a `ConnectionPt` for.

**Returns:**

the `ConnectingPt` or null

---

## getConnectingPt

```
public ConnectingPt getConnectingPt(int i)
```

Returns a `ConnectingPt` for a given `int`. If the specified argument is too large, will return the last `ConnectingPt` in the `Connections` vector.

**Parameters:**

`i` - the index of the required `ConnectingPt` in the `Connections` vector

**Returns:**

the `ConnectingPt` at the specified index

---

## getOrientationMenu

```
public JMenu getOrientationMenu()
```

Creates a menu appropriate for selecting the desired orientation for this drawn component.

**Returns:**

a `JMenu` with appropriate items or null if no orientations are appropriate

---

## setLabelString

```
public void setLabelString(String str,  
                           int index)
```

Adds or replaces a `String` in the labels array. If there is already a `String` at index `n`, the `String` is replaced. If the index is greater than the number of items currently in the array the `String` is added to the end.

**Parameters:**

`str` - the `String` to be added

`index` - the `int` index into the label array.

---

## isPosnSet

```
public boolean isPosnSet()
```

Determines if the `DrawnComponent`'s position has been initialized.

**Returns:**

false if the center of the `DrawnComponent` is (0.0, 0.0) true if the center of the `DrawnComponent` is not (0.0, 0.0)

---

## createPopupMenu

```
public JPopupMenu createPopupMenu()
```

Creates a custom popup context menu for this `DrawnComponent` that includes an optional `Actions` menu from sub components.

**Returns:**

the `JPopupMenu` created from the component.

**See Also:**

`.getComponent()`

---



## getCustomPopupActions

```
public Action[] getCustomPopupActions()
```

Retrieves the custom popup actions from this drawn component's AbstractComponent.

**Returns:**

the Action[] returned from the component.

**See Also:**

`AbstractComponent.getCustomPopupActions()`

---

## isDrawBadges

```
public boolean isDrawBadges()
```

Getter for property drawBadges.

**Returns:**

Value of property drawBadges.

---

## setDrawBadges

```
public void setDrawBadges(boolean drawBadges)
```

Setter for property drawBadges.

**Parameters:**

`drawBadges` - New value of property drawBadges.

---

## isAutoScale

```
public boolean isAutoScale()
```

Getter for property autoScale that determines whether scaleIt calls scale this Drawn Component.

**Returns:**

Value of property autoScale.

---

## setAutoScale

```
public void setAutoScale(boolean autoScale)
```

Setter for property autoScale that determines whether scaleIt calls scale this Drawn Component.

**Parameters:**

`autoScale` - If true, scaleIt calls will resize the visual representation of this drawn component.

---

## getPreferredSize

```
public java.awt.Dimension getPreferredSize()
```

Provides the preferred size for this component. Overridden here to maintain size during cut&paste operations.

**Returns:**

the Dimension containing the un-scaled width and height.

---

## mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent e)
```

---

(continued from last page)

---

## mouseExited

```
public void mouseExited(java.awt.event.MouseEvent e)
```

---

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent e)
```

---

---

## mousePressed

```
public void mousePressed(java.awt.event.MouseEvent e)
```

---

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

---

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent e)
```

---

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent evt)
```

---

---

## showConnections

```
public boolean showConnections()
```

---

---

## getBeanBox

```
public BeanBox getBeanBox()
```

Retrieves the BeanBox that contains this DrawnComponent or null if this DrawnComponent exists outside of a BeanBox.

**Returns:**

the BeanBox at the top of the ancestor list.

---

---

## getGlassPane

```
public GlassPanel getGlassPane()
```

Retrieves the GlassPanel from the ZoomablePanel that contains this DrawnComponent.

**Returns:**

the GlassPanel from the ZoomablePanel.

**See Also:**

[.getZoomablePanel\(\)](#)

---

## getZoomablePanel

```
public ZoomablePanel getZoomablePanel()
```

Retrieves the ZoomablePanel that contains this DrawnComponent or null if this DrawnComponent exists outside of a ZoomablePanel.

**Returns:**

the ZoomablePanel at the top of the ancestor list.

---

## translateConnectionToScreen

```
public java.awt.Point translateConnectionToScreen(ConnectingPt pt)
```

This converts the center of a passed connecting Point into the coordinates on the zoomable panel. This assumes that the passed point is contained within this DrawnComponent.

**Parameters:**

pt - The ConnectingPt whose center is to be translated.

**Returns:**

The coordinates on the zoomable panel for the connecting Point's center.

---

## translatePointToScreen

```
public java.awt.Point translatePointToScreen(java.awt.Point point)
```

This converts a Point from local coordinates into coordinates on the zoomable panel. Local coordinates are based around the center of the DrawnComponent.

**Parameters:**

point - the Point that is getting translated.

**Returns:**

The coordinates on the zoomable panel for the Point.

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

**Parameters:**

state - A hash table containing modified parameters.

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

**Parameters:**

state - A hash table containing modified parameters.

---

## componentChanged

```
public void componentChanged(ComponentChangedEvent evt)
```

When a DrawnComponent's component changes, the drawn component needs to call initDrawing on itself { @inheritDoc }

---

(continued from last page)

---

## componentDeleted

```
public void componentDeleted()
```

When a DrawnComponent's component is deleted, the Drawn component needs to remove itself from the view.  
{ @inheritDoc }

---

## componentConnected

```
public void componentConnected(Connection con)
```

This should make sure that the ZoomablePanel tries to add the drawnLink to the view when a connection is established.  
{ @inheritDoc }

---

## removeNotify

```
public void removeNotify()
```

---

## addNotify

```
public void addNotify()
```

---

## componentDisconnected

```
public void componentDisconnected(Connection con)
```

This should make sure that the zoomable panel tries to remove the drawnLink from the view when a connection is removed. { @inheritDoc }

---

## setParent

```
public void setParent(java.awt.Container parent)
```

Setter for the parent value of this DrawnComponent. This is used for situations where the DrawnComponent needs a parent reference to a panel that it doesn't exist inside. For example: The NodeViewPanel doesn't actually own the DrawnComponent it displays.

### Parameters:

parent - the Container that owns this DrawnComponent.

---

## getParent

```
public java.awt.Container getParent()
```

This either gets the actual parent of the DrawnComponent, if it exists inside a Container. Otherwise, it gets the locally held parent value.

### See Also:

.setParent()

---

## isSelected

```
public boolean isSelected()
```

Getter for property selected. This is true if this DrawnComponent is currently selected in the BeanBox.

### Returns:

Value of property selected.

---

(continued from last page)

---

## setSelected

```
public void setSelected(boolean selected)
```

Setter for property selected.

### Parameters:

selected - New value of property selected.

---

## getCustomPopupItems

```
public Vector getCustomPopupItems()
```

Returns the custom popup items of this DrawnComponent's target component

---

## store

```
public com.apt.xdr.PibBlock store(int viewNum)
```

This function returns a PibBlock for a drawn component.

### Parameters:

comp - the DrawnComponent to be converted to PibBlock format.  
viewid - the unique identifier of a ViewComponent.

### Returns:

the DrawnComponentRec to store the given component.

---

## loadDrawnComponent

```
public void loadDrawnComponent(com.apt.xdr.PibBlock block)
```

This function loads a DrawnComponent from a DrawnComponentRec.

### Parameters:

base - the AbstractComponent whose DrawnComponent is being loaded.  
rec - the DrawnComponentRec being loaded.

### Returns:

the DrawnComponent loaded from base.

---

## hasSubComponents

```
public boolean hasSubComponents()
```

Returns true if this DrawnComponent has DrawnSubComponents contained within it

---

## getSubComponentAt

```
public DrawnSubComponent getSubComponentAt(int x,  
                                             int y)
```

Returns the sub-component at the given coordinates, or null if no sub-component exists at that location.

---

## createTemplateEntry

```
public TemplateEntry createTemplateEntry()
```

Creates a new TemplateEntry for this DrawnComponent that stores all of the location and state data for this DrawnComponent. Any DrawnComponent that has unique data associated with it should store that data in an extension of TemplateEntry and overwrite this method.

---

(continued from last page)

## **readTemplateEntry**

```
public void readTemplateEntry(TemplateEntry entry)
```

Sets the data on this DrawnComponent from a TemplateEntry read in from a view template file. Any DrawnComponent that has unique data associated with it should read that data in, assuming a TemplateEntry.

---

## **getConnectingLocation**

```
public java.awt.Point getConnectingLocation(ConnectingPt point)
```

Returns the absolute screen location of the given ConnectingPt.

## com.cafean.client.ui Class DrawnConnection

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- com.cafean.client.ui.DrawnComponent
                          |
                          +- com.cafean.client.ui.DrawnConnection

```

### All Implemented Interfaces:

[FullScreenDrawing](#), [Cloneable](#), [java.io.Serializable](#), [java.awt.MenuContainer](#), [java.awt.image.ImageObserver](#), [java.io.Serializable](#), [ComponentListener](#), [StateEditable](#), [java.awt.event.MouseMotionListener](#), [java.awt.event.MouseListener](#), [Cloneable](#)

public class **DrawnConnection**

extends [DrawnComponent](#)

implements [Cloneable](#), [java.awt.event.MouseListener](#), [java.awt.event.MouseMotionListener](#), [StateEditable](#), [ComponentListener](#), [java.io.Serializable](#), [java.awt.image.ImageObserver](#), [java.awt.MenuContainer](#), [java.io.Serializable](#), [Cloneable](#), [FullScreenDrawing](#)

A renderrer for a Connectionbetween two components that draws a segmented line between the DrawnComponentrenderers of the two sides of the connection.

Junction Knees algorithm adapted from Graphic Gems II This algorithm for determining the best "neat" path between two objects is described in Graphics Gems \* gem number IV.2. By Claudio Rosato, A Simple Connection Algorithm for 2D

## Field Summary

|                     |                                    |
|---------------------|------------------------------------|
| static final double | DEFAULT_SIZE<br>Value: <b>60.0</b> |
| static final double | HALFWIDTH<br>Value: <b>7.0</b>     |

### Fields inherited from class [com.cafean.client.ui.DrawnComponent](#)

[BOTTOM](#), [CENTER](#), [CENTER\\_H](#), [CENTER\\_V](#), [CIRCLE](#), [CROSSHATCH](#), [DIAMOND](#), [DOWN](#), [LEFT](#), [max\\_positions](#), [NONE](#), [PIXELS\\_P\\_METER](#), [RIGHT](#), [SEGMENT\\_BOTH](#), [SEGMENT\\_INLET](#), [SEGMENT\\_NONE](#), [SEGMENT\\_OUTLET](#), [SEGMENT\\_SPECIAL](#), [SQUARE](#), [TOP](#), [TRIANGLE](#), [UP](#)

### Fields inherited from class [javax.swing.JComponent](#)

[TOOL\\_TIP\\_TEXT\\_KEY](#), [UNDEFINED\\_CONDITION](#), [WHEN\\_ANCESTOR\\_OF\\_FOCUSED\\_COMPONENT](#), [WHEN\\_FOCUSED](#), [WHEN\\_IN\\_FOCUSED\\_WINDOW](#)

### Fields inherited from class [java.awt.Component](#)

[BOTTOM\\_ALIGNMENT](#), [CENTER\\_ALIGNMENT](#), [LEFT\\_ALIGNMENT](#), [RIGHT\\_ALIGNMENT](#), [TOP\\_ALIGNMENT](#)

## Constructor Summary

|        |  |
|--------|--|
| public | DrawnConnection(Connection connection)<br>Creates a renderer for the given Connection object.  |
| public | DrawnConnection(Connection connection, DrawnComponent source, DrawnComponent target)<br>Creates a renderer for the given Connection object and initializes its' source and target DrawnComponent references. |

## Method Summary

|   |  |
|---|--|
| boolean                                   | addPoint(java.awt.Point p)<br>Adds a path point at the given point, if the point falls on an existing line segment.                                  |
| boolean                                   | canAddPoint(java.awt.Point p)<br>Return true if a point can be added to the path.  |
| boolean                                   | canRemovePoint(java.awt.Point p)<br>Return true if a point can be removed from the path.   |
| void                                      | componentChanged(ComponentChangedEvent evt)  |
| void                                      | componentRemoved(DrawnComponent comp)<br>This should be called by the view when a drawn component has been removed.                                  |
| void                                      | componentReshaped(DrawnComponent comp)<br>This should be called by the view when a drawn component has been removed.                                 |
| void                                      | connectionPointRemoved(DrawnComponent comp)  |
| boolean                                   | contains(double x, double y)   |
| boolean                                   | contains(int x, int y)   |
| boolean                                   | contains(java.awt.Point p)   |
| JComponent[]                              | createDisplayBeans(int pixelsPerMeter, double widthScaleFactor, ClassLoader loader)  |
| static<br><a href="#">DrawnConnection</a> | createDrawnConnection(Connection connection, DrawnComponent source, DrawnComponent target)   |
| TemplateEntry                             | createTemplateEntry()<br>Creates a new TemplateEntry for this DrawnComponent that stores all of the location and state data for this DranwComponent. |
| void                                      | draw(java.awt.Graphics2D g, boolean selected)<br>Draws indicator colored lines between each of this DrawnConnection's plotted points.                |



|                                |  |
|--------------------------------|--|
| java.awt.Point[]               | getPath()<br>Retrieves a copy of the plotted path points that this DrawnConnection is painting.  |
| <a href="#">DrawnComponent</a> | getSource()  |
| <a href="#">DrawnComponent</a> | getTarget()  |
| String                         | getToolTipText()   |
| java.awt.Rectangle             | getUsedBounds()  |
| void                           | InitDrawing()  |
| boolean                        | isGeneratePoints()<br>Getter for property generatePoints.  |
| boolean                        | isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)  |
| void                           | loadDrawnComponent(com.appt.xdr.PibBlock block)<br>This function loads a DrawnComponent from a DrawnComponentRec.                      |
| void                           | mouseClicked(java.awt.event.MouseEvent e)  |
| void                           | mouseDragged(java.awt.event.MouseEvent e)<br>Handle mouseDragged events for segment and point manipulation                             |
| void                           | mouseEntered(java.awt.event.MouseEvent e)  |
| void                           | mouseExited(java.awt.event.MouseEvent e)   |
| void                           | mouseMoved(java.awt.event.MouseEvent e)  |
| void                           | mousePressed(java.awt.event.MouseEvent e)<br>Handle mousePressed events to support segment and point manipulation                      |
| void                           | mouseReleased(java.awt.event.MouseEvent e)<br>Handle mouseReleased events to support segment and point manipulation                    |
| void                           | readTemplateEntry(TemplateEntry entry)<br>Sets the data on this DrawnComponent from a TemplateEntry read in from a view template file. |
| void                           | removeClosestPoint(java.awt.Point p)<br>Attempts to remove the closest point in this DrawnConnection's set of points.                  |
| boolean                        | removePoint(java.awt.Point p)<br>Attempts to remove the path point at the given location.  |
| void                           | repaint()  |

|                       |  |
|-----------------------|--|
| void                  | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.                       |
| void                  | setGeneratePoints(boolean generatePoints)<br>Setter for property generatePoints.                           |
| void                  | setLocation(int x,int y)   |
| void                  | setPath( java.awt.Point[] path)<br>Sets the path points that this DrawnConnection will use for connecting. |
| void                  | setSelected(boolean selected)  |
| com.appt.xdr.PibBlock | store(int viewNum)<br>This function returns a PibBlock for a drawn component.                              |
| void                  | storeState(Hashtable state)<br>Store the state of the bean to permit undo.                                 |
| String                | toString()   |
| void                  | translate(int dx,int dy)   |
| void                  | validate()   |

#### Methods inherited from class [com.cafean.client.ui.DrawnComponent](#)

[addNotify](#), [canBeResized](#), [Clear](#), [clearLinks](#), [clone](#), [componentChanged](#),  
[componentConnected](#), [componentDeleted](#), [componentDisconnected](#), [connectLinks](#), [contains](#),  
[contains](#), [contains](#), [createBorderRegion](#), [createCenterShape](#), [createConnectionPrototypes](#),  
[createConnectionPt](#), [createDisplayBeans](#), [createPopupMenu](#), [createTemplateEntry](#),  
[disconnectAllMyLinks](#), [draw](#), [drawLabelStrings](#), [flip](#), [getBeanBox](#), [getClockwiseFace](#),  
[getComponent](#), [getComponentID](#), [getConnectingLocation](#), [getConnectingPt](#), [getConnectingPt](#),  
[getConnectingPtAt](#), [getConnectionSize](#), [getCounterFace](#), [getCrossflowIndex](#),  
[getCustomPopupActions](#), [getCustomPopupItems](#), [getDefaultDrawLength](#), [getDefaultDrawWidth](#),  
[getDrawAngle](#), [getDrawingFace](#), [getDrawingObject](#), [getFaceByAngle](#), [getFillColor](#),  
[getGlassPane](#), [getHandleSize](#), [getIndicatorColor](#), [getLength](#), [getLenScaleFactor](#),  
[getMaxHeight](#), [getMaxWidth](#), [getMinWidth](#), [getMirrorImageShape](#), [getNormalObj](#),  
[getNumberConnections](#), [getOppositeFace](#), [getOrientation](#), [getOrientationMenu](#),  
[getOrientationName](#), [getParent](#), [getPreferredSize](#), [getSelectedConnector](#),  
[getSelectedDropZone](#), [getSubComponentAt](#), [getToolTipText](#), [getToolTipText](#),  
[getWidthScaleFactor](#), [getX\\_Pos](#), [getXDistBetweenCPs](#), [getXDistBetweenXflowCPs](#), [getY\\_Pos](#),  
[getZoomablePanel](#), [hasSubComponents](#), [InitDrawing](#), [isAutoScale](#), [isDrawBadges](#),  
[isObjectInsideBounds](#), [isPlenumShaped](#), [isPosnSet](#), [isScalable](#), [isSegmentSet](#), [isSelected](#),  
[isValveShaped](#), [loadDrawnComponent](#), [mouseClicked](#), [mouseDragged](#), [mouseEntered](#),  
[mouseExited](#), [mouseMoved](#), [mousePressed](#), [mouseReleased](#), [moveRel](#), [moveTo](#), [paint](#),  
[paintComponent](#), [print](#), [readTemplateEntry](#), [removeNotify](#), [repositionLinks](#),  
[repositionLinks](#), [resetPosition](#), [restoreState](#), [rotateTo](#), [rotateTo](#), [scaleIt](#),  
[setAutoScale](#), [setBackupComponent](#), [setBounds](#), [setComponent](#), [setDrawAngle](#),  
[setDrawBadges](#), [setDrawHeight](#), [setDrawWidth](#), [setEqualTo](#), [setLabelString](#),  
[setLenScaleFactor](#), [setOrientation](#), [setOrientationByAngle](#), [setParent](#), [setSelected](#),  
[setSizeTo](#), [setWidthScaleFactor](#), [setX\\_Pos](#), [setY\\_Pos](#), [showConnections](#), [store](#), [storeState](#),  
[toString](#), [translateConnectionToScreen](#), [translatePointToScreen](#)

#### Methods inherited from class [javax.swing.JComponent](#)

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate
```

#### Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Fields

### DEFAULT\_SIZE

```
public static final double DEFAULT_SIZE
```

### HALFWIDTH

```
public static final double HALFWIDTH
```

## Constructors

(continued from last page)

## DrawnConnection

```
public DrawnConnection(Connection connection)
```

Creates a renderer for the given Connection object. The source and target DrawnComponent references will be initialized automatically at draw time.

### Parameters:

connection - the Connection that this DrawnConnection is rendering.

---

## DrawnConnection

```
public DrawnConnection(Connection connection,  
                      DrawnComponent source,  
                      DrawnComponent target)
```

Creates a renderer for the given Connection object and initializes its' source and target DrawnComponent references.

### Parameters:

connection - the Connection that this DrawnConnection is rendering.

source - the DrawnComponent that is the left of connection

target - the DrawnComponent that is the right of connection

## Methods

### InitDrawing

```
public void InitDrawing()
```

---

### validate

```
public void validate()
```

---

### setLocation

```
public void setLocation(int x,  
                       int y)
```

---

### setPath

```
public void setPath(java.awt.Point[] path)
```

Sets the path points that this DrawnConnection will use for connecting.

---

### createDisplayBeans

```
public JComponent[] createDisplayBeans(int pixelsPerMeter,  
                                       double widthScaleFactor,  
                                       ClassLoader loader)
```

---

### mousePressed

```
public void mousePressed(java.awt.event.MouseEvent e)
```

Handle mousePressed events to support segment and point manipulation

---

(continued from last page)

**See Also:**

`.mouseReleased()`

---

**mouseClicked**

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

---

**removeClosestPoint**

```
public void removeClosestPoint(java.awt.Point p)
```

Attempts to remove the closest point in this DrawnConnection's set of points.

**Parameters:**

p - the Point to remove the closest path point.

---

**removePoint**

```
public boolean removePoint(java.awt.Point p)
```

Attempts to remove the path point at the given location.

**Parameters:**

p - the Point at which to delete a path point.

**Returns:**

true if the point was deleted.

---

**canRemovePoint**

```
public boolean canRemovePoint(java.awt.Point p)
```

Return true if a point can be removed from the path.

---

**canAddPoint**

```
public boolean canAddPoint(java.awt.Point p)
```

Return true if a point can be added to the path.

---

**addPoint**

```
public boolean addPoint(java.awt.Point p)
```

Adds a path point at the given point, if the point falls on an existing line segment.

**Parameters:**

p - the Point at which to add a path point.

**Returns:**

true if the point is actually added.

---

**mouseReleased**

```
public void mouseReleased(java.awt.event.MouseEvent e)
```

Handle mouseReleased events to support segment and point manipulation

**See Also:**

(continued from last page)

```
.mousePressed()
```

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent e)
```

Handle mouseDragged events for segment and point manipulation

### See Also:

```
.mousePressed()
```

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent e)
```

---

## mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent e)
```

---

## mouseExited

```
public void mouseExited(java.awt.event.MouseEvent e)
```

---

## getToolTipText

```
public String getToolTipText()
```

---

## isObjectInsideBounds

```
public boolean isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)
```

---

## contains

```
public boolean contains(double x,  
                        double y)
```

---

## contains

```
public boolean contains(java.awt.Point p)
```

---

## contains

```
public boolean contains(int x,  
                        int y)
```

---

(continued from last page)

---

## draw

```
public void draw(java.awt.Graphics2D g,  
                 boolean selected)
```

Draws indicator colored lines between each of this DrawnConnection's plotted points.

### Parameters:

- g - the Graphics2D object which will do the painting
- selected - indicates whether the current DrawnComponent is currently selected

---

## getSource

```
public DrawnComponent getSource()
```

### Returns:

the source drawn component

---

## getTarget

```
public DrawnComponent getTarget()
```

### Returns:

the target drawn component

---

## componentRemoved

```
public void componentRemoved(DrawnComponent comp)
```

This should be called by the view when a drawn component has been removed. If the given DrawnComponent is this connection's source or target, it removes itself as well.

---

## componentReshaped

```
public void componentReshaped(DrawnComponent comp)
```

This should be called by the view when a drawn component has been removed. If the given DrawnComponent is this connection's source or target, it removes itself as well.

---

## connectionPointRemoved

```
public void connectionPointRemoved(DrawnComponent comp)
```

---

## repaint

```
public void repaint()
```

---

## translate

```
public void translate(int dx,  
                     int dy)
```

---



(continued from last page)

---

## getUsedBounds

```
public java.awt.Rectangle getUsedBounds()
```

---

## componentChanged

```
public void componentChanged(ComponentChangedEvent evt)
```

---

## toString

```
public String toString()
```

---

## setSelected

```
public void setSelected(boolean selected)
```

---

## isGeneratePoints

```
public boolean isGeneratePoints()
```

Getter for property generatePoints.

**Returns:**

Value of property generatePoints.

---

## setGeneratePoints

```
public void setGeneratePoints(boolean generatePoints)
```

Setter for property generatePoints.

**Parameters:**

generatePoints - New value of property generatePoints.

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

**Parameters:**

state - A hash table containing modified parameters.

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

**Parameters:**

state - A hash table containing modified parameters.

---

(continued from last page)

---

## getPath

```
public java.awt.Point[] getPath()
```

Retrieves a copy of the plotted path points that this DrawnConnection is painting.

**Returns:**

a Point[] containing the coordinates of each point.

---

## store

```
public com.apl.xdr.PibBlock store(int viewNum)
```

This function returns a PibBlock for a drawn component.

**Parameters:**

comp - the DrawnComponent to be converted to PibBlock format.

viewid - the unique identifier of a ViewComponent.

**Returns:**

the DrawnComponentRec to store the given component.

---

## loadDrawnComponent

```
public void loadDrawnComponent(com.apl.xdr.PibBlock block)
```

This function loads a DrawnComponent from a DrawnComponentRec.

**Parameters:**

base - the AbstractComponent whose DrawnComponent is being loaded.

rec - the DrawnComponentRec being loaded.

**Returns:**

the DrawnComponent loaded from base.

---

## createDrawnConnection

```
public static DrawnConnection createDrawnConnection(Connection connection,  
    DrawnComponent source,  
    DrawnComponent target)
```

---

## createTemplateEntry

```
public TemplateEntry createTemplateEntry()
```

Creates a new TemplateEntry for this DrawnComponent that stores all of the location and state data for this DrawnComponent. Any DrawnComponent that has unique data associated with it should store that data in an extension of TemplateEntry and overwrite this method.

---

## readTemplateEntry

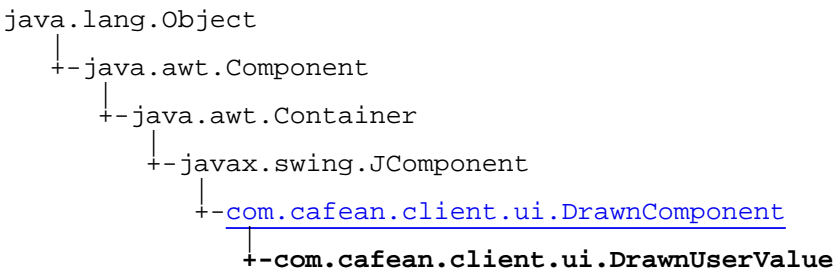
```
public void readTemplateEntry(TemplateEntry entry)
```

Sets the data on this DrawnComponent from a TemplateEntry read in from a view template file. Any DrawnComponent that has unique data associated with it should read that data in, assuming a TemplateEntry.

---

com.cafean.client.ui

Class DrawnUserValue



All Implemented Interfaces:

java.io.Serializable, StateEditable, java.awt.event.ActionListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, [ComponentListener](#), StateEditable, java.awt.event.MouseMotionListener, java.awt.event.MouseListener, Cloneable

public class **DrawnUserValue**  
extends [DrawnComponent](#)  
implements Cloneable, java.awt.event.MouseListener, java.awt.event.MouseMotionListener, StateEditable, [ComponentListener](#), java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.awt.event.ActionListener, StateEditable, java.io.Serializable

A Representation of a DrawnUserValue in a DrawnView

|   |
|---|
| Fields inherited from class <a href="#">com.cafean.client.ui.DrawnComponent</a>   |
| <a href="#">BOTTOM</a> , <a href="#">CENTER</a> , <a href="#">CENTER_H</a> , <a href="#">CENTER_V</a> , <a href="#">CIRCLE</a> , <a href="#">CROSSHATCH</a> , <a href="#">DIAMOND</a> , <a href="#">DOWN</a> , <a href="#">LEFT</a> , <a href="#">max_positions</a> , <a href="#">NONE</a> , <a href="#">PIXELS_P_METER</a> , <a href="#">RIGHT</a> , <a href="#">SEGMENT_BOTH</a> , <a href="#">SEGMENT_INLET</a> , <a href="#">SEGMENT_NONE</a> , <a href="#">SEGMENT_OUTLET</a> , <a href="#">SEGMENT_SPECIAL</a> , <a href="#">SQUARE</a> , <a href="#">TOP</a> , <a href="#">TRIANGLE</a> , <a href="#">UP</a> |
| Fields inherited from class javax.swing.JComponent  |
| <a href="#">TOOL_TIP_TEXT_KEY</a> , <a href="#">UNDEFINED_CONDITION</a> , <a href="#">WHEN_ANCESTOR_OF_FOCUSED_COMPONENT</a> , <a href="#">WHEN_FOCUSED</a> , <a href="#">WHEN_IN_FOCUSED_WINDOW</a>  |
| Fields inherited from class java.awt.Component  |
| <a href="#">BOTTOM_ALIGNMENT</a> , <a href="#">CENTER_ALIGNMENT</a> , <a href="#">LEFT_ALIGNMENT</a> , <a href="#">RIGHT_ALIGNMENT</a> , <a href="#">TOP_ALIGNMENT</a>  |

| Constructor Summary |   |
|---------------------|---|
| public              | <code>DrawnUserValue(UserDefinedValue value)</code><br>Creates a new instance of DrawnUserValue |

Method Summary

|                               |  |
|-------------------------------|--|
| void                          | actionPerformed( java.awt.event.ActionEvent evt )<br>Responds to an ActionEvent produced when hitting 'enter' in the text field contained in this drawing. |
| void                          | draw( java.awt.Graphics2D g,boolean selected)  |
| JLabel                        | getLabel()<br>Getter for property label.   |
| JPanel                        | getPanel()<br>Getter for property panel.   |
| java.awt.Dimension            | getPreferredSize()   |
| JLabel                        | getUnits()<br>Getter for property units.   |
| JLabel                        | getValueLabel()<br>Getter for property valueLabel.   |
| <a href="#">RealTextField</a> | getValueText()<br>Getter for property valueText.   |
| void                          | InitDrawing()<br>Updates the values inside this component, and updates it's bounds   |
| boolean                       | isEditable()<br>Getter for property editable.  |
| boolean                       | isShowLabel()<br>Getter for property showLabel.  |
| boolean                       | isShowUnits()<br>Getter for property showUnits.  |
| void                          | loadDrawnComponent( com.appt.xdr.PibBlock block)   |
| void                          | popupEditor()<br>Opens a dialog for editing the properties of this drawing.  |
| void                          | readByteArray(byte[] byteArr)<br>Reads this DrawnUserValue's data from the given byte[] of XML Encoded information.  |
| void                          | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.   |
| void                          | setBackground( java.awt.Color color)   |
| void                          | setEditable(boolean editable)<br>Setter for property editable.   |
| void                          | setForeground( java.awt.Color color)   |

|                       |  |
|-----------------------|--|
| void                  | setLabel(JLabel label)<br>Setter for property label.                       |
| void                  | setPanel(JPanel panel)<br>Setter for property panel.                       |
| void                  | setShowLabel(boolean showLabel)<br>Setter for property showLabel.          |
| void                  | setShowUnits(boolean showUnits)<br>Setter for property showUnits.          |
| void                  | setUnits(JLabel units)<br>Setter for property units.                       |
| void                  | setValueLabel(JLabel valueLabel)<br>Setter for property valueLabel.        |
| void                  | setValueText(RealTextField valueText)<br>Setter for property valueText.    |
| com.appt.xdr.PibBlock | store(int viewNum)   |
| void                  | storeState(Hashtable state)<br>Store the state of the bean to permit undo. |

#### Methods inherited from class [com.cafean.client.ui.DrawnComponent](#)

[addNotify](#), [canBeResized](#), [Clear](#), [clearLinks](#), [clone](#), [componentChanged](#),  
[componentConnected](#), [componentDeleted](#), [componentDisconnected](#), [connectLinks](#), [contains](#),  
[contains](#), [contains](#), [createBorderRegion](#), [createCenterShape](#), [createConnectionPrototypes](#),  
[createConnectionPt](#), [createDisplayBeans](#), [createPopupMenu](#), [createTemplateEntry](#),  
[disconnectAllMyLinks](#), [draw](#), [drawLabelStrings](#), [flip](#), [getBeanBox](#), [getClockwiseFace](#),  
[getComponent](#), [getComponentID](#), [getConnectingLocation](#), [getConnectingPt](#), [getConnectingPt](#),  
[getConnectingPtAt](#), [getConnectionSize](#), [getCounterFace](#), [getCrossflowIndex](#),  
[getCustomPopupActions](#), [getCustomPopupItems](#), [getDefaultDrawLength](#), [getDefaultDrawWidth](#),  
[getDrawAngle](#), [getDrawingFace](#), [getDrawingObject](#), [getFaceByAngle](#), [getFillColor](#),  
[getGlassPane](#), [getHandleSize](#), [getIndicatorColor](#), [getLength](#), [getLenScaleFactor](#),  
[getMaxHeight](#), [getMaxWidth](#), [getMinWidth](#), [getMirrorImageShape](#), [getNormalObj](#),  
[getNumberConnections](#), [getOppositeFace](#), [getOrientation](#), [getOrientationMenu](#),  
[getOrientationName](#), [getParent](#), [getPreferredSize](#), [getSelectedConnector](#),  
[getSelectedDropZone](#), [getSubComponentAt](#), [getToolTipText](#), [getToolTipText](#),  
[getWidthScaleFactor](#), [getX\\_Pos](#), [getXDistBetweenCPs](#), [getXDistBetweenXflowCPs](#), [getY\\_Pos](#),  
[getZoomablePanel](#), [hasSubComponents](#), [InitDrawing](#), [isAutoScale](#), [isDrawBadges](#),  
[isObjectInsideBounds](#), [isPlenumShaped](#), [isPosnSet](#), [isScalable](#), [isSegmentSet](#), [isSelected](#),  
[isValveShaped](#), [loadDrawnComponent](#), [mouseClicked](#), [mouseDragged](#), [mouseEntered](#),  
[mouseExited](#), [mouseMoved](#), [mousePressed](#), [mouseReleased](#), [moveRel](#), [moveTo](#), [paint](#),  
[paintComponent](#), [print](#), [readTemplateEntry](#), [removeNotify](#), [repositionLinks](#),  
[repositionLinks](#), [resetPosition](#), [restoreState](#), [rotateTo](#), [rotateTo](#), [scaleIt](#),  
[setAutoScale](#), [setBackupComponent](#), [setBounds](#), [setComponent](#), [setDrawAngle](#),  
[setDrawBadges](#), [setDrawHeight](#), [setDrawWidth](#), [setEqualTo](#), [setLabelString](#),  
[setLenScaleFactor](#), [setOrientation](#), [setOrientationByAngle](#), [setParent](#), [setSelected](#),  
[setSizeTo](#), [setWidthScaleFactor](#), [setX\\_Pos](#), [setY\\_Pos](#), [showConnections](#), [store](#), [storeState](#),  
[toString](#), [translateConnectionToScreen](#), [translatePointToScreen](#)

#### Methods inherited from class [javax.swing.JComponent](#)

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### DrawnUserValue

```
public DrawnUserValue(UserDefinedValue value)
```

Creates a new instance of DrawnUserValue

## Methods

### InitDrawing

```
public void InitDrawing()
```

Updates the values inside this component, and updates it's bounds

(continued from last page)

---

## actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent evt)
```

Responds to an ActionEvent produced when hitting 'enter' in the text field contained in this drawing.

---

## draw

```
public void draw(java.awt.Graphics2D g,  
    boolean selected)
```

---

## getPreferredSize

```
public java.awt.Dimension getPreferredSize()
```

---

## isEditable

```
public boolean isEditable()
```

Getter for property editable.

**Returns:**

Value of property editable.

---

## setEditable

```
public void setEditable(boolean editable)
```

Setter for property editable.

**Parameters:**

editable - New value of property editable.

---

## isShowLabel

```
public boolean isShowLabel()
```

Getter for property showLabel.

**Returns:**

Value of property showLabel.

---

## setShowLabel

```
public void setShowLabel(boolean showLabel)
```

Setter for property showLabel.

**Parameters:**

showLabel - New value of property showLabel.

---

## isShowUnits

```
public boolean isShowUnits()
```

Getter for property showUnits.

**Returns:**

Value of property showUnits.

---



---

## setShowUnits

```
public void setShowUnits(boolean showUnits)
```

Setter for property showUnits.

### Parameters:

showUnits - New value of property showUnits.

---

## getLabel

```
public JLabel getLabel()
```

Getter for property label.

### Returns:

Value of property label.

---

## setLabel

```
public void setLabel(JLabel label)
```

Setter for property label.

### Parameters:

label - New value of property label.

---

## getUnits

```
public JLabel getUnits()
```

Getter for property units.

### Returns:

Value of property units.

---

## setUnits

```
public void setUnits(JLabel units)
```

Setter for property units.

### Parameters:

units - New value of property units.

---

## getPanel

```
public JPanel getPanel()
```

Getter for property panel.

### Returns:

Value of property panel.

---

## setPanel

```
public void setPanel(JPanel panel)
```

Setter for property panel.

### Parameters:

(continued from last page)

panel - New value of property panel.

---

## getValueText

```
public RealTextField getValueText()
```

Getter for property valueText.

### Returns:

Value of property valueText.

---

## setValueText

```
public void setValueText(RealTextField valueText)
```

Setter for property valueText.

### Parameters:

valueText - New value of property valueText.

---

## getValueLabel

```
public JLabel getValueLabel()
```

Getter for property valueLabel.

### Returns:

Value of property valueLabel.

---

## setLabel

```
public void setLabel(JLabel valueLabel)
```

Setter for property valueLabel.

### Parameters:

valueLabel - New value of property valueLabel.

---

## popupEditor

```
public void popupEditor()
```

Opens a dialog for editing the properties of this drawing.

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

### Parameters:

state - A hash table containing modified parameters.

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

### Parameters:

state - A hash table containing modified parameters.

---

---

## setForeground

```
public void setForeground(java.awt.Color color)
```

---

## setBackground

```
public void setBackground(java.awt.Color color)
```

---

## readByteArray

```
public void readByteArray(byte[] byteArr)
```

Reads this DrawnUserValue's data from the given byte[] of XML Encoded information.

### Parameters:

byteArr - a byte[] containing the result of using storeState to produce a Hashtable of a DrawnUserValue's current state then using `java.beans.XMLEncoder` to encode that hashtable.

---

## store

```
public com.apt.xdr.PibBlock store(int viewNum)
```

---

## loadDrawnComponent

```
public void loadDrawnComponent(com.apt.xdr.PibBlock block)
```

## com.cafean.client.ui

### Class DrawnView

```

java.lang.Object
|
+- java.awt.Component
|   |
|   +- java.awt.Container
|       |
|       +- javax.swing.JComponent
|           |
|           +- javax.swing.JPanel
|               |
|               +- com.cafean.client.ui.DrawnView
  
```

#### All Implemented Interfaces:

MenuListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

public class **DrawnView**

extends JPanel

implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, MenuListener

The DrawnView is the panel that displays AbstractComponents and their connections. Each DrawnView contains a {@link ZoomablePanel} that allows the user to zoom into or out of a view. Each DrawnView also corresponds to a {@link ViewComponent} inside the {@link AbstractModel}. The ViewComponent stores all the important information for creating a DrawnView while loading and saving a model. The DrawnView is a MenuListener on the edit menu. The Tools menu of a DrawnView is filled by both the {@link MECodePlugin} of it's {@link AbstractModel} and by the {@link MEFeaturePlugin} Feature plugins} in the ModelEditor.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | DrawnView(AbstractModel model, ViewComponent viewComp)<br>Creates a new instance of DrawnView. |
|--------|--|

## Method Summary

|      |  |
|------|--|
| void | addAnnotation(Annotation comp)<br>This adds an {@link Annotation} to the center of the panel 's viewport.                  |
| void | addAnnotation(Annotation comp, boolean tofront)<br>This adds an {@link Annotation} to the center of the panel 's viewport. |

|                                  |  |
|----------------------------------|--|
| void                             | addComponents(Iterator itr)<br>Adds a list of components by an iterator on a list of components.   |
| void                             | addComponents(Iterator itr,boolean select)<br>Adds a list of components by an iterator on a list of components.                                  |
| <a href="#">DrawnComponent</a>   | addDrawnComponent(AbstractComponent comp,boolean select)<br>This adds a specific AbstractComponent to a drawn view.                              |
| void                             | addDrawnComponent(DrawnComponent comp,boolean select)<br>This adds a specific DrawnComponent to this DrawnView.                                  |
| void                             | addDrawnComponent(DrawnComponent comp,boolean select,boolean withCons)<br>This adds a specific DrawnComponent to this DrawnView.                 |
| void                             | addMenuItem(JMenuItem item)<br>This allows <a href="#">plugins</a> to add { @link JMenuItem menu items } to the "Tools" menu for this DrawnView. |
| void                             | addOverlapPanel(OverlapPanel panel)<br>Adds an overlapping internal panel to this view for use as a user input sub-dialog.                       |
| void                             | addToolBar(JToolBar toolbar,String prefix,boolean visible)<br>Adds the given toolbar to this DrawnView.  |
| void                             | clearSelection()<br>Clears the current selection from the zoom panel's bean box  |
| Vector                           | getAnnotations()<br>Gets all of the annotations in the panel for this Drawn View.  |
| <a href="#">BeanBox</a>          | getBeanBox()<br>An accessor for the BeanBox from the ZoomablePanel   |
| java.awt.Dimension               | getCanvasSize()<br>Getter for the size of the ZoomablePanel  |
| Vector                           | getComponents(boolean includeConnections)<br>Returns all of the components stored inside the beanbox in an array.                                |
| <a href="#">DrawnComponent</a>   | getDrawnComponentAt(int i)<br>Returns the DrawnComponent at a given index inside the bean Box.   |
| int                              | getDrawnComponentCount()<br>Returns the number of DrawnComponents that exist inside this drawn view.   |
| <a href="#">DrawnComponent[]</a> | getDrawnComponents()<br>Returns all of the components stored inside the beanbox in an array.   |
| JMenuBar                         | getMenubar()<br>   |
| <a href="#">AbstractModel</a>    | getModel()<br>Gets the AbstractModel for this DrawnView.   |

|                                  |   |
|----------------------------------|---|
| <a href="#">DrawnComponent[]</a> | <p><code>getSelectedComponents()</code></p> <p>Returns all of the components that are selected inside the beanbox in an array.</p>  |
| String                           | <p><code>getTitle()</code></p> <p>Returns the title of the dialog containing this view or the name of the ViewComponent.</p>  |
| <a href="#">Toolbox</a>          | <p><code>getToolbox()</code></p> <p>Retrieves this DrawnView's Toolbox.</p>   |
| <a href="#">ViewComponent</a>    | <p><code>getViewComponent()</code></p> <p>Gets the ViewComponent for this DrawnView.</p>  |
| java.awt.Point                   | <p><code>getViewPosition()</code></p> <p>Returns the upper left coordinate of the viewport in the main scroll pane this should make sure that the view stores this location for a local save.</p> |
| java.awt.Dimension               | <p><code>getViewSize()</code></p> <p>Returns the dimensions of the viewport in the main scroll pane.</p>  |
| <a href="#">ZoomablePanel</a>    | <p><code>getZoomablePanel()</code></p> <p>An accessor for the ZoomablePanel</p>   |
| double                           | <p><code>getZoomScale()</code></p> <p>Getter for the scale of the {@link ZoomablePanel}</p>   |
| void                             | <p><code>layoutView()</code></p> <p>This organizes all of the components within the view</p>  |
| void                             | <p><code>menuCanceled(MenuEvent e)</code></p> <p>This implements the functionality of the MenuListener interface.</p>   |
| void                             | <p><code>menuDeselected(MenuEvent e)</code></p> <p>This implements the functionality of the MenuListener interface.</p>   |
| void                             | <p><code>menuSelected(MenuEvent e)</code></p> <p>This implements the functionality of the MenuListener interface.</p>   |
| boolean                          | <p><code>printView()</code></p> <p>Prints this view</p>   |
| boolean                          | <p><code>printView(org.apache.batik.transcoder.print.PrintTranscoder transcoder)</code></p> <p>Prints this view into the given transcoder.</p>  |
| void                             | <p><code>resetZoomMenu()</code></p> <p>Reset the zoom menu to ensure the current zoom factor is selected or none if the zoom is between selections.</p>   |
| void                             | <p><code>setCanvasSize(java.awt.Dimension size)</code></p> <p>Setter for the size of the ZoomablePanel</p>  |
| void                             | <p><code>setLocked(boolean locked)</code></p> <p>Locks this View ( and updates the ViewComponent ).</p>   |

|              |  |
|--------------|--|
| void         | setTitle(String title)<br>Sets the title of the dialog containing this view or does nothing if no dialog exists  |
| void         | setViewPosition(java.awt.Point p)<br>Sets the upper left coordinate of the viewport in the main scroll pane this should make sure that the view shows exactly what was shown when last open. |
| void         | setVisible(boolean b)  |
| void         | setZoomScale(double scale)<br>Setter for the scale of the { @link ZoomablePanel }  |
| java.io.File | snapshotView()<br>Creates an image snapshot of this view for use in the DrawnViewComponent.  |
| void         | toFront()<br>Selects this view in the tabbed pane or toFront's the dialog that contains it.  |
| void         | updateToolbars()<br>Enables and/or disables toolbar buttons based on such things as the current BeanBox selection.   |

**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Page A-264



(continued from last page)

## DrawnView

```
public DrawnView(AbstractModel model,  
                ViewComponent viewComp)
```

Creates a new instance of DrawnView. This should only be done by the ViewComponent that is opening.

## Methods

### getMenubar

```
public JMenuBar getMenubar()
```

### addOverlapPanel

```
public void addOverlapPanel(OverlapPanel panel)
```

Adds an overlapping internal panel to this view for use as a user input sub-dialog.

### addToolbar

```
public void addToolbar(JToolBar toolbar,  
                      String prefix,  
                      boolean visible)
```

Adds the given toolbar to this DrawnView. The toolbar's visibility will default to the given visibility and controlled by a menu item and user preference, both named by the toolbar's current name.

#### Parameters:

- toolbar - the JToolBar instance to add to this view's toolbar panel. Buttons may be added before or after this method is called.
- prefix - a String containing the prefix to the ModelEditor preference to use when showing or hiding the toolbar
- visible - if true, the toolbar will be visible by default

### setLocked

```
public void setLocked(boolean locked)
```

Locks this View ( and updates the ViewComponent ).

### getToolbox

```
public Toolbox getToolbox()
```

Retrieves this DrawnView's Toolbox.

#### Returns:

the Toolbox for this DrawnView.

### getViewComponent

```
public ViewComponent getViewComponent()
```

Gets the ViewComponent for this DrawnView.

#### Returns:

the ViewComponent.

(continued from last page)

---

## getModel

```
public AbstractModel getModel()
```

Gets the AbstractModel for this DrawnView.

**Returns:**

the AbstractModel.

---

## addComponents

```
public void addComponents(Iterator itr)
```

Adds a list of components by an iterator on a list of components. These components are not selected after they are added to the view. Any connection between components in the view is rendered with a { @link DrawnConnection }.

**Parameters:**

itr - the Iterator on a list of components.

---

## addComponents

```
public void addComponents(Iterator itr,  
    boolean select)
```

Adds a list of components by an iterator on a list of components. Any connection between components in the view is rendered with a { @link DrawnConnection }.

**Parameters:**

itr - the Iterator on a list of components.

select - true if the components should be added to the selection after they are added.

---

## addDrawnComponent

```
public void addDrawnComponent(DrawnComponent comp,  
    boolean select,  
    boolean withCons)
```

This adds a specific DrawnComponent to this DrawnView. The DrawnComponent's position will not be set.

**Parameters:**

comp - the DrawnComponent to be added.

select - true if the component should be added to the selection after it is added.

withCons - if true, DrawnConnection will be created for each connection in the component whos opposite side exists in the view.

---

## addDrawnComponent

```
public void addDrawnComponent(DrawnComponent comp,  
    boolean select)
```

This adds a specific DrawnComponent to this DrawnView. The DrawnComponent's position will not be set. DrawnConnection will be created for each connection in the component whos opposite side exists in the view.

**Parameters:**

comp - the DrawnComponent to be added.

select - true if the component should be added to the selection after it is added.

---

## addDrawnComponent

```
public DrawnComponent addDrawnComponent(AbstractComponent comp,  
    boolean select)
```

(continued from last page)

This adds a specific `AbstractComponent` to a drawn view. A new `DrawnComponent` is created from the component, and is inserted into the center of the viewport. If the component has a connection to or from any components already in the view are rendered with a `@link DrawnConnection`.

**Parameters:**

- `comp` - the `AbstractComponent` to be added.
- `select` - true if the component should be added to the selection after it is added.

---

## addAnnotation

```
public void addAnnotation(Annotation comp)
```

This adds an `{ @link Annotation }` to the center of the panel 's viewport. After the Annotation is added, it resets it's bounds.

**Parameters:**

- `comp` - the Annotation being added to the view.

---

## addAnnotation

```
public void addAnnotation(Annotation comp,  
    boolean tofront)
```

This adds an `{ @link Annotation }` to the center of the panel 's viewport. After the Annotation is added, it resets it's bounds.

**Parameters:**

- `comp` - the Annotation being added to the view.

---

## layoutView

```
public void layoutView()
```

This organizes all of the components within the view

---

## getDrawnComponentCount

```
public int getDrawnComponentCount()
```

Returns the number of `DrawnComponents` that exist inside this drawn view.

**Returns:**

The number of components inside the beanbox

---

## getDrawnComponentAt

```
public DrawnComponent getDrawnComponentAt(int i)
```

Returns the `DrawnComponent` at a given index inside the bean Box.

**Parameters:**

- `i` - The index of the drawn component.

**Returns:**

the `DrawnComponent` at `i`.

---

## getDrawnComponents

```
public DrawnComponent[] getDrawnComponents()
```

Returns all of the components stored inside the beanbox in an array. This includes connections but not annotations.

(continued from last page)

**Returns:**

the DrawnComponent[] of components in the beanbox.

---

## getComponents

```
public Vector getComponents(boolean includeConnections)
```

Returns all of the components stored inside the beanbox in an array. This may include connections but not annotations.

**Parameters:**

`includeConnections` - if true DrawnConnections will be included in the selection.

**Returns:**

the Vector containing the components in the beanbox.

---

## getSelectedComponents

```
public DrawnComponent[] getSelectedComponents()
```

Returns all of the components that are selected inside the beanbox in an array. This includes connections but not annotations.

**Returns:**

the DrawnComponent[] of selected components in the beanbox.

---

## setVisible

```
public void setVisible(boolean b)
```

---

## snapshotView

```
public java.io.File snapshotView()
```

Creates an image snapshot of this view for use in the DrawnViewComponent.

**Returns:**

a java.io.File that refers to the temporary file that the snapshot is stored in.

---

## printView

```
public boolean printView()
```

Prints this view

**Returns:**

true if successful

---

## printView

```
public boolean printView(org.apache.batik.transcoder.print.PrintTranscoder transcoder)
```

Prints this view into the given transcoder.

**Parameters:**

`transcoder` - the PrintTranscoder to print into.

**Returns:**

true if successful

---

## resetZoomMenu

```
public void resetZoomMenu()
```

Reset the zoom menu to ensure the current zoom factor is selected or none if the zoom is between selections.

---

## getCanvasSize

```
public java.awt.Dimension getCanvassize()
```

Getter for the size of the ZoomablePanel

**Returns:**

the Dimension containing the size of the ZoomablePanel's canvas.

---

## setCanvasSize

```
public void setCanvassize(java.awt.Dimension size)
```

Setter for the size of the ZoomablePanel

**Parameters:**

size - the Dimension containing the new size of the ZoomablePanel's canvas.

**See Also:**

`ZoomablePanel.setPanelSize()`

---

## getZoomScale

```
public double getZoomScale()
```

Getter for the scale of the { @link ZoomablePanel }

**Returns:**

the double scale factor of the ZoomablePanel.

**See Also:**

`ZoomablePanel.getScale()`

---

## setZoomScale

```
public void setZoomScale(double scale)
```

Setter for the scale of the { @link ZoomablePanel }

**Parameters:**

scale - The new scale of the ZoomablePanel

---

## getViewPosition

```
public java.awt.Point getViewPosition()
```

Returns the upper left coordinate of the viewport in the main scroll pane this should make sure that the view stores this location for a local save.

**Returns:**

the view position of the main scroll pane's view port.

---

(continued from last page)

---

## getViewSize

```
public java.awt.Dimension getViewSize()
```

Returns the dimensions of the viewport in the main scroll pane. This is used to determine if a component is inside the current view.

---

## setViewPosition

```
public void setViewPosition(java.awt.Point p)
```

Sets the upper left coordinate of the viewport in the main scroll pane this should make sure that the view shows exactly what was shown when last open.

### Parameters:

p - The Point setting the view position of the main scroll pane's viewport.

---

## clearSelection

```
public void clearSelection()
```

Clears the current selection from the zoom panel's bean box

---

## updateToolbars

```
public void updateToolbars()
```

Enables and/or disables toolbar buttons based on such things as the current BeanBox selection.

---

## menuSelected

```
public void menuSelected(MenuEvent e)
```

This implements the functionality of the `MenuListener` interface. Invoked when the edit menu is selected. This allows the BeanBox to determine if the various events can occur with the given selection.

### Parameters:

e - the MenuEvent object

### See Also:

`BeanBox.canCopy()`  
`BeanBox.canPaste()`  
`BeanBox.canDelete()`  
`ComponentPaster.canPasteSpecial()`

---

## menuDeselected

```
public void menuDeselected(MenuEvent e)
```

This implements the functionality of the `MenuListener` interface. Invoked when the menu is deselected.

### Parameters:

e - the MenuEvent object { @inheritDoc }

---

## menuCanceled

```
public void menuCanceled(MenuEvent e)
```

This implements the functionality of the `MenuListener` interface. Invoked when the menu is cancelled.

### Parameters:

e - the MenuEvent object { @inheritDoc }

## addMenuItem

```
public void addMenuItem(JMenuItem item)
```

This allows [plugins](#) to add { @link JMenuItem menu items} to the "Tools" menu for this DrawnView. The item is appended to the bottom of the "Tools" menu.

**Parameters:**

item - the JMenuItem being added to the view.

---

## getAnnotations

```
public Vector getAnnotations()
```

Gets all of the annotations in the panel for this Drawn View.

**Returns:**

the Vector containing all of the annotations in this view.

**See Also:**

[BeanBox.getAnnotations\(\)](#)

---

## getBeanBox

```
public BeanBox getBeanBox()
```

An accessor for the BeanBox from the ZoomablePanel

**Returns:**

the BeanBox

---

## getZoomablePanel

```
public ZoomablePanel getZoomablePanel()
```

An accessor for the ZoomablePanel

**Returns:**

the ZoomablePanel

---

## toFront

```
public void toFront()
```

Selects this view in the tabbed pane or toFront's the dialog that contains it.

---

## getTitle

```
public String getTitle()
```

Returns the title of the dialog containing this view or the name of the ViewComponent.

---

## setTitle

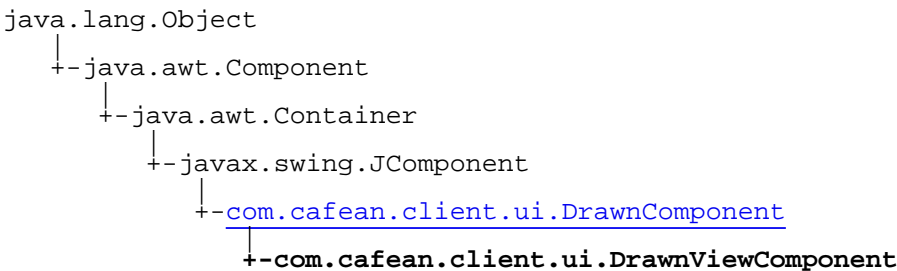
```
public void setTitle(String title)
```

Sets the title of the dialog containing this view or does nothing if no dialog exists

---

# com.cafean.client.ui

## Class DrawnViewComponent



public class **DrawnViewComponent**  
extends [DrawnComponent](#)

A DrawnComponent extension used to render one ViewComponent inside another View. This includes embedded connections and an image.

**See Also:**  
[DrawnEmbeddedConnection](#), [ViewComponent](#)

|   |
|---|
| <b>Fields inherited from class</b> <a href="#">com.cafean.client.ui.DrawnComponent</a>  |
| <a href="#">BOTTOM</a> , <a href="#">CENTER</a> , <a href="#">CENTER_H</a> , <a href="#">CENTER_V</a> , <a href="#">CIRCLE</a> , <a href="#">CROSSHATCH</a> , <a href="#">DIAMOND</a> , <a href="#">DOWN</a> , <a href="#">LEFT</a> , <a href="#">max_positions</a> , <a href="#">NONE</a> , <a href="#">PIXELS_P_METER</a> , <a href="#">RIGHT</a> , <a href="#">SEGMENT_BOTH</a> , <a href="#">SEGMENT_INLET</a> , <a href="#">SEGMENT_NONE</a> , <a href="#">SEGMENT_OUTLET</a> , <a href="#">SEGMENT_SPECIAL</a> , <a href="#">SQUARE</a> , <a href="#">TOP</a> , <a href="#">TRIANGLE</a> , <a href="#">UP</a> |

|  |
|--|
| <b>Fields inherited from class</b> <a href="#">javax.swing.JComponent</a>  |
| <a href="#">TOOL_TIP_TEXT_KEY</a> , <a href="#">UNDEFINED_CONDITION</a> , <a href="#">WHEN_ANCESTOR_OF_FOCUSED_COMPONENT</a> , <a href="#">WHEN_FOCUSED</a> , <a href="#">WHEN_IN_FOCUSED_WINDOW</a> |

|  |
|--|
| <b>Fields inherited from class</b> <a href="#">java.awt.Component</a>  |
| <a href="#">BOTTOM_ALIGNMENT</a> , <a href="#">CENTER_ALIGNMENT</a> , <a href="#">LEFT_ALIGNMENT</a> , <a href="#">RIGHT_ALIGNMENT</a> , <a href="#">TOP_ALIGNMENT</a> |

| Constructor Summary |   |
|---------------------|---|
| public              | <a href="#">DrawnViewComponent</a> ( <a href="#">AbstractComponent</a> c)<br>Creates a new instance of <a href="#">DrawnComponentView</a> |

| Method Summary                          |   |
|---|---|
| void                                    | <a href="#">draw</a> ( <a href="#">java.awt.Graphics2D</a> g,boolean selected)<br>Draw a simple X'd box if there is no image. |
| <a href="#">DrawnEmbeddedConnection</a> | <a href="#">findEmbedded</a> ( <a href="#">DrawnEmbeddedConnection[]</a> array,int comp, <a href="#">ConnectionData</a> data) |



|                              |  |
|------------------------------|--|
| <a href="#">ConnectingPt</a> | getConnectingPt(int compIdent, ConnectionData data)                                  |
| Vector                       | getCustomPopupItems()  |
| java.awt.Dimension           | getMinimumSize()   |
| DrawnSubComponent            | getSubComponentAt(int x, int y)  |
| boolean                      | hasSubComponents()   |
| void                         | InitDrawing()<br>Creates a draw view component and puts it in the drawing vectors.   |
| boolean                      | isRepresenting(int compid)   |
| void                         | loadDrawnComponent(com.appt.xdr.PibBlock block)                                      |
| void                         | mouseClicked(java.awt.event.MouseEvent e)  |
| void                         | mouseMoved(java.awt.event.MouseEvent e)  |
| void                         | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit. |
| void                         | setBounds(java.awt.Rectangle r)  |
| com.appt.xdr.PibBlock        | store(int viewNum)   |
| void                         | storeState(Hashtable state)<br>Store the state of the bean to permit undo.           |

Methods inherited from class [com.cafean.client.ui.DrawnComponent](#)

[addNotify](#), [canBeResized](#), [Clear](#), [clearLinks](#), [clone](#), [componentChanged](#),  
[componentConnected](#), [componentDeleted](#), [componentDisconnected](#), [connectLinks](#), [contains](#),  
[createConnectionPt](#), [createDisplayBeans](#), [createPopupMenu](#), [createTemplateEntry](#),  
[disconnectAllMyLinks](#), [draw](#), [drawLabelStrings](#), [flip](#), [getBeanBox](#), [getClockwiseFace](#),  
[getComponent](#), [getComponentID](#), [getConnectingLocation](#), [getConnectingPt](#), [getConnectingPt](#),  
[getConnectingPtAt](#), [getConnectSize](#), [getCounterFace](#), [getCrossflowIndex](#),  
[getCustomPopupActions](#), [getCustomPopupItems](#), [getDefaultDrawLength](#), [getDefaultDrawWidth](#),  
[getDrawAngle](#), [getDrawingFace](#), [getDrawingObject](#), [getFaceByAngle](#), [getFillColor](#),  
[getGlassPane](#), [getHandleSize](#), [getIndicatorColor](#), [getLength](#), [getLenScaleFactor](#),  
[getMaxHeight](#), [getMaxWidth](#), [getMinWidth](#), [getMirrorImageShape](#), [getNormalObj](#),  
[getNumberConnections](#), [getOppositeFace](#), [getOrientation](#), [getOrientationMenu](#),  
[getOrientationName](#), [getParent](#), [getPreferredSize](#), [getSelectedConnector](#),  
[getSelectedDropZone](#), [getSubComponentAt](#), [getToolTipText](#), [getToolTipText](#),  
[getWidthScaleFactor](#), [getX\\_Pos](#), [getXDistBetweenCPS](#), [getXDistBetweenXflowCPS](#), [getY\\_Pos](#),  
[getZoomablePanel](#), [hasSubComponents](#), [InitDrawing](#), [isAutoScale](#), [isDrawBadges](#),  
[isObjectInsideBounds](#), [isPlenumShaped](#), [isPosnSet](#), [isScalable](#), [isSegmentSet](#), [isSelected](#),  
[isValveShaped](#), [loadDrawnComponent](#), [mouseClicked](#), [mouseDragged](#), [mouseEntered](#),  
[mouseExited](#), [mouseMoved](#), [mousePressed](#), [mouseReleased](#), [moveRel](#), [moveTo](#), [paint](#),  
[paintComponent](#), [print](#), [readTemplateEntry](#), [removeNotify](#), [repositionLinks](#),  
[repositionLinks](#), [resetPosition](#), [restoreState](#), [rotateTo](#), [rotateTo](#), [scaleIt](#),  
[setAutoScale](#), [setBackupComponent](#), [setBounds](#), [setComponent](#), [setDrawAngle](#),  
[setDrawBadges](#), [setDrawHeight](#), [setDrawWidth](#), [setEqualTo](#), [setLabelString](#),  
[setLenScaleFactor](#), [setOrientation](#), [setOrientationByAngle](#), [setParent](#), [setSelected](#),  
[setSizeTo](#), [setWidthScaleFactor](#), [setX\\_Pos](#), [setY\\_Pos](#), [showConnections](#), [store](#), [storeState](#),  
[toString](#), [translateConnectionToScreen](#), [translatePointToScreen](#)

#### Methods inherited from class javax.swing.JComponent

[addAncestorListener](#), [addNotify](#), [addVetoableChangeListener](#), [computeVisibleRect](#),  
[contains](#), [createToolTip](#), [disable](#), [enable](#), [firePropertyChange](#), [firePropertyChange](#),  
[firePropertyChange](#), [getAccessibleContext](#), [getActionForKeyStroke](#), [getActionMap](#),  
[getAlignmentX](#), [getAlignmentY](#), [getAncestorListeners](#), [getAutoscrolls](#), [getBorder](#),  
[getBounds](#), [getClientProperty](#), [getComponentPopupMenu](#), [getConditionForKeyStroke](#),  
[getDebugGraphicsOptions](#), [getDefaultLocale](#), [getFontMetrics](#), [getGraphics](#), [getHeight](#),  
[getInheritsPopupMenu](#), [getInputMap](#), [getInputMap](#), [getInputVerifier](#), [getInsets](#),  
[getInsets](#), [getListeners](#), [getLocation](#), [getMaximumSize](#), [getMinimumSize](#),  
[getNextFocusableComponent](#), [getPopupLocation](#), [getPreferredSize](#),  
[getRegisteredKeyStrokes](#), [getRootPane](#), [getSize](#), [getToolTipLocation](#), [getToolTipText](#),  
[getToolTipText](#), [getTopLevelAncestor](#), [getTransferHandler](#), [getUIClassID](#),  
[getVerifyInputWhenFocusTarget](#), [getVetoableChangeListeners](#), [getVisibleRect](#), [getWidth](#),  
[getX](#), [getY](#), [grabFocus](#), [isDoubleBuffered](#), [isLightweightComponent](#), [isManagingFocus](#),  
[isOpaque](#), [isOptimizedDrawingEnabled](#), [isPaintingTile](#), [isRequestFocusEnabled](#),  
[isValidRoot](#), [paint](#), [paintImmediately](#), [paintImmediately](#), [print](#), [printAll](#),  
[putClientProperty](#), [registerKeyboardAction](#), [registerKeyboardAction](#),  
[removeAncestorListener](#), [removeNotify](#), [removeVetoableChangeListener](#), [repaint](#), [repaint](#),  
[requestDefaultFocus](#), [requestFocus](#), [requestFocus](#), [requestFocusInWindow](#),  
[resetKeyboardActions](#), [reshape](#), [revalidate](#), [scrollRectToVisible](#), [setActionMap](#),  
[setAlignmentX](#), [setAlignmentY](#), [setAutoscrolls](#), [setBackground](#), [setBorder](#),  
[setComponentPopupMenu](#), [setDebugGraphicsOptions](#), [setDefaultLocale](#), [setDoubleBuffered](#),  
[setEnabled](#), [setFocusTraversalKeys](#), [setFont](#), [setForeground](#), [setInheritsPopupMenu](#),  
[setInputMap](#), [setInputVerifier](#), [setMaximumSize](#), [setMinimumSize](#),  
[setNextFocusableComponent](#), [setOpaque](#), [setPreferredSize](#), [setRequestFocusEnabled](#),  
[setToolTipText](#), [setTransferHandler](#), [setVerifyInputWhenFocusTarget](#), [setVisible](#),  
[unregisterKeyboardAction](#), [update](#), [updateUI](#)

#### Methods inherited from class java.awt.Container

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyChangeListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, set, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

(continued from last page)

## DrawnViewComponent

```
public DrawnViewComponent(AbstractComponent c)
```

Creates a new instance of DrawnComponentView

## Methods

### store

```
public com.apr.xdr.PibBlock store(int viewNum)
```

### loadDrawnComponent

```
public void loadDrawnComponent(com.apr.xdr.PibBlock block)
```

### InitDrawing

```
public void InitDrawing()
```

Creates a draw view component and puts it in the drawing vectors.

### isRepresenting

```
public boolean isRepresenting(int compid)
```

### getConnectingPt

```
public ConnectingPt getConnectingPt(int compIdent,  
    ConnectionData data)
```

### findEmbedded

```
public DrawnEmbeddedConnection findEmbedded(DrawnEmbeddedConnection[] array,  
    int comp,  
    ConnectionData data)
```

### hasSubComponents

```
public boolean hasSubComponents()
```

### getSubComponentAt

```
public DrawnSubComponent getSubComponentAt(int x,  
    int y)
```

(continued from last page)

## getCustomPopupItems

```
public Vector getCustomPopupItems()
```

---

## draw

```
public void draw(java.awt.Graphics2D g,  
                boolean selected)
```

Draw a simple X'd box if there is no image.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

---

## getMinimumSize

```
public java.awt.Dimension getMinimumSize()
```

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent e)
```

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

### Parameters:

state - A hash table containing modified parameters.

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

### Parameters:

state - A hash table containing modified parameters.

---

## com.cafean.client.ui Interface FullScreenDrawing

### All Known Implementing Classes:

LineAnnotation, DrawnConnection

### public interface FullScreenDrawing

An interface describing a visual DrawnComponent or Annotation extension who's size is always that of the BeanBox that contains it.

### See Also:

DrawnConnection, LineAnnotation

## Method Summary

|                    |  |
|--------------------|--|
| java.awt.Rectangle | getUsedBounds()<br>Retrieves the bounds actually used by this FullScreenDrawing. |
| void               | translate(int dx,int dy)<br>Translates this drawing by the given x and y deltas. |

## Methods

### translate

```
public void translate(int dx,  
                     int dy)
```

Translates this drawing by the given x and y deltas.

### getUsedBounds

```
public java.awt.Rectangle getUsedBounds()
```

Retrieves the bounds actually used by this FullScreenDrawing.

## com.cafean.client.ui

### Class GlassPanel

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- javax.swing.JPanel
                          |
                          +- com.cafean.client.ui.GlassPanel

```

#### All Implemented Interfaces:

java.awt.event.MouseMotionListener, java.awt.event.MouseListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

```

public class GlassPanel
extends JPanel
implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver,
java.awt.MenuContainer, java.io.Serializable, java.awt.event.MouseListener,
java.awt.event.MouseMotionListener

```

A glass panel that rests on top of the {[@link BeanBox bean box](#)} in a view This panel handles the rendering of the move, resize and rubberband selection boxes. Using a glass panel allows lines to draw without requiring the components they pass over to redraw. This greatly speeds up drawing rubberband lines and boxes. This class should only be used by DrawnViews.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |                                       |
|--------|---------------------------------------|
| public | <code>GlassPanel()</code>             |
|        | The constructor for a new GlassPanel. |

## Method Summary

|      |   |
|------|---|
| void | <code>clearRubberBoxes()</code>                       |
|      | Clears all of the current rubberband selection boxes. |
| void | <code>deleteRubberBox()</code>                        |
|      | Erases the existing rubberband selection box.         |

|                                   |   |
|-----------------------------------|---|
| void                              | deleteRubberBoxes()<br>Erase the current rubberband selection boxes.  |
| void                              | deleteRubberLine(java.awt.Point begin, java.awt.Point end)<br>Erases an existing line between two points.   |
| void                              | deleteRubberLines(Vector points, java.awt.Point last)<br>Erase the current rubberband connection lines.   |
| void                              | drawRubberBox(java.awt.Rectangle box)<br>Draws a rubberband box on the GlassPane that corresponds to the given Rectangle coordinates and size.      |
| void                              | drawRubberBoxes(java.awt.Rectangle[] boxes)<br>Draws all the rubberband selection boxes.  |
| void                              | drawRubberLine(java.awt.Point begin, java.awt.Point end, java.awt.Point oldPoint)<br>Draws the rubberband connection line between two given points. |
| void                              | drawRubberLines(Vector points, java.awt.Point last, java.awt.Point oldPoint)<br>Draw the rubberband connection lines                                |
| void                              | forwardMouseEvent(java.awt.event.MouseEvent e)<br>Forward a mouse event to the current mouse target, setting it if necessary.                       |
| <a href="#">AbstractComponent</a> | getConnectionSource()<br>Gets the AbstractComponent that started the connection.  |
| java.awt.Dimension                | getMaximumSize()<br>Return the maximum size of this panel, which is one pixel larger then the BeanBox below it.                                     |
| String                            | getToolTipText(java.awt.event.MouseEvent event)<br>Returns the string to be used as the tooltip for event.  |
| java.awt.Component                | locateComponentAt(int xpos, int ypos)<br>Locate a component at a specific position in the display.  |
| java.awt.Component                | locateComponentAt(int xpos, int ypos, boolean returnChild)<br>Locate a component at a specific position in the display.                             |
| void                              | mouseClicked(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler  |
| void                              | mouseDragged(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler  |
| void                              | mouseEntered(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler  |
| void                              | mouseExited(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler   |



|         |  |
|---------|--|
| void    | mouseMoved(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler   |
| void    | mousePressed(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler   |
| void    | mouseReleased(java.awt.event.MouseEvent orig_evt)<br>Proxies the given MouseEvent to the current MouseHandler  |
| void    | paintComponent(java.awt.Graphics g)<br>Repaint the glasspane.  |
| void    | removeNotify()<br>{ @inheritDoc } Overriden here to unregister from the tooltip manager  |
| void    | repaint(long tm,int x,int y,int width,int height)<br>When repainting a region, add a buffer of 10 pixels all the way region to make sure that all connection bands repaint as well |
| boolean | requiresTarget()<br>Determines if a target ConnectingPt is required for the current Connection.  |

**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Page A-282

(continued from last page)

## GlassPanel

```
public GlassPanel()
```

The constructor for a new GlassPanel. This sets all of the default values on the GlassPanel.

## Methods

### removeNotify

```
public void removeNotify()
```

Overriden here to unregister from the tooltip manager

### getToolTipText

```
public String getToolTipText(java.awt.event.MouseEvent event)
```

Returns the string to be used as the tooltip for event. If the mouse is hovering over a DrawnComponent, that DrawnComponent's name is used.

**Parameters:**

event - the MouseEvent generated by the ToolTipManager.

**Returns:**

the name of this DrawnComponent's Component.

### getMaximumSize

```
public java.awt.Dimension getMaximumSize()
```

Return the maximum size of this panel, which is one pixel larger then the BeanBox below it.

**Returns:**

the Dimension containing the maximum size of the panel.

### drawRubberLines

```
public void drawRubberLines(Vector points,  
    java.awt.Point last,  
    java.awt.Point oldPoint)
```

Draw the rubberband connection lines

**Parameters:**

points - the Vector of points for the line.

last - the Point added before the current position.

oldPoint - the previous location of the mouse, needed for removing the previous line.

### deleteRubberLines

```
public void deleteRubberLines(Vector points,  
    java.awt.Point last)
```

Erase the current rubberband connection lines.

**Parameters:**

points - the Vector of points in the line.

last - the last fixed anchor point.

(continued from last page)

---

## drawRubberLine

```
public void drawRubberLine(java.awt.Point begin,  
    java.awt.Point end,  
    java.awt.Point oldPoint)
```

Draws the rubberband connection line between two given points. Before the new line is drawn, the line that already exists to the old point is drawn again, with the background color, erasing it.

### Parameters:

`begin` - the Point where the line begins.  
`end` - the Point where the line ends.  
`oldPoint` - the previous endpoint.

---

## deleteRubberLine

```
public void deleteRubberLine(java.awt.Point begin,  
    java.awt.Point end)
```

Erases an existing line between two points.

### Parameters:

`begin` - the Point where the line begins.  
`end` - the Point where the line ends.

---

## drawRubberBox

```
public void drawRubberBox(java.awt.Rectangle box)
```

Draws a rubberband box on the GlassPane that corresponds to the given Rectangle coordinates and size.

### Parameters:

`box` - the Rectangle for the rubberband box.

---

## deleteRubberBox

```
public void deleteRubberBox()
```

Erases the existing rubberband selection box.

---

## drawRubberBoxes

```
public void drawRubberBoxes(java.awt.Rectangle[] boxes)
```

Draws all the rubberband selection boxes. This is used to draw a red selection box around all of the currently selected components.

### Parameters:

`boxes` - the Rectangle[].

---

## clearRubberBoxes

```
public void clearRubberBoxes()
```

Clears all of the current rubberband selection boxes. This does not actually erase the rubberbands, this just removes the reference to the previously drawn boxes.

---

## deleteRubberBoxes

```
public void deleteRubberBoxes()
```

Erase the current rubberband selection boxes.

---

## mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

orig\_evt - the MouseEvent being proxied to the MouseHandler

---

## mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

orig\_evt - the MouseEvent being proxied to the MouseHandler

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

orig\_evt - the MouseEvent being proxied to the MouseHandler

---

## mouseExited

```
public void mouseExited(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

orig\_evt - the MouseEvent being proxied to the MouseHandler

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

orig\_evt - the MouseEvent being proxied to the MouseHandler

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

orig\_evt - the MouseEvent being proxied to the MouseHandler

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

Proxies the given MouseEvent to the current MouseHandler

**Parameters:**

(continued from last page)

`orig_evt` - the `MouseEvent` being proxied to the `MouseHandler`

---

## forwardMouseEvent

```
public void forwardMouseEvent(java.awt.event.MouseEvent e)
```

Forward a mouse event to the current mouse target, setting it if necessary.

### Parameters:

`e` - The mouse event to be forwarded.

---

## repaint

```
public void repaint(long tm,  
    int x,  
    int y,  
    int width,  
    int height)
```

When repainting a region, add a buffer of 10 pixels all the way region to make sure that all connection bands repaint as well

### Parameters:

`tm` - this parameter is not used  
`x` - the x value of the dirty region  
`y` - the y value of the dirty region  
`width` - the width of the dirty region  
`height` - the height of the dirty region

### See Also:

`java.awt.Component.isShowing()`  
`RepaintManager.addDirtyRegion()`

---

## paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

Repaint the glasspane. Draw the selection and drag rectangles around each selected component.

### Parameters:

`g` - the `Graphics` object.

---

## locateComponentAt

```
public java.awt.Component locateComponentAt(int xpos,  
    int ypos)
```

Locate a component at a specific position in the display.

### Parameters:

`xpos` - The x coordinate.  
`ypos` - The y coordinate.

### Returns:

The component located at the given coordinate, or null if none found.

---

## locateComponentAt

```
public java.awt.Component locateComponentAt(int xpos,  
    int ypos,  
    boolean returnChild)
```

Locate a component at a specific position in the display.

---

(continued from last page)

**Parameters:**

xpos - The x coordinate.

ypos - The y coordinate.

returnChild - if true, this will return the child component at the give coordinates.

**Returns:**

The component located at the given coordinate, or null if none found.

---

**requiresTarget**

```
public boolean requiresTarget()
```

Determines if a target ConnectingPt is required for the current Connection.

---

**getConnectionSource**

```
public AbstractComponent getConnectionSource()
```

Gets the AbstractComponent that started the connection.

## com.cafean.client.ui Class LocalSubmitDialog

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── java.awt.Window
│   │   │   ├── java.awt.Dialog
│   │   │   │   ├── javax.swing.JDialog
│   │   │   │   └-- com.cafean.client.ui.LocalSubmitDialog
```

public class **LocalSubmitDialog**  
extends JDialog

This dialog allows the user to submit a job to a local calculation server. Since the local calculation server is not associated with a database, the user cannot input any kind of identifier for the runs.

Plugin and context specific extensions of this class should take note of #updateRunOptions for use in sending parameters, options or files to the the analysis code.

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |   |
|--------|---|
| public | <code>LocalSubmitDialog(JFrame parent,org.omg.CORBA.ORB theOrb,String cType,AbstractModel theModel)</code><br>Creates a new LocalSubmitDialog for use in submitting an input deck and MED format model to a Calculation Server. |
|--------|---|

## Method Summary

|      |  |
|------|--|
| void | <code>setInitialValues(int projectId,int modelId,int restartId)</code><br>Sets the initial project, model and restart id's.                          |
| void | <code>setRunOptions(com.cafean.CalcServer.command.RunOptions[] options)</code><br>Sets the Run Options passed to LaunchCalc when a job is submitted. |
| void | <code>setVisible(boolean visible)</code>   |

### Methods inherited from class javax.swing.JDialog



getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setJMenuBar, setLayeredPane, setLayout, update

**Methods inherited from class java.awt.Dialog**

addNotify, getAccessibleContext, getTitle, hide, isModal, isResizable, isUndecorated, setModal, setResizable, setTitle, setUndecorated, show

**Methods inherited from class java.awt.Window**

addNotify, addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getAccessibleContext, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, show, toBack, toFront

**Methods inherited from class java.awt.Container**

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

**Methods inherited from class java.awt.Component**

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### LocalSubmitDialog

```

public LocalSubmitDialog(JFrame parent,
                        org.omg.CORBA.ORB theOrb,
                        String cType,
                        AbstractModel theModel)

```

Creates a new LocalSubmitDialog for use in submitting an input deck and MED format model to a Calculation Server.

#### Parameters:

- parent - the JFrame parent used for modality.
- theOrb - the ORB (CORBA Object Request Broker) used by the application
- cType - a String containing the calculation type, must match a registered plugin's ID.
- theModel - the AbstractModel being submitted for execution.

(continued from last page)

## Methods

### setVisible

```
public void setVisible(boolean visible)
```

---

### setRunOptions

```
public void setRunOptions(com.cafean.CalcServer.command.RunOptions[] options)
```

Sets the Run Options passed to LaunchCalc when a job is submitted.

---

### setInitialValues

```
public void setInitialValues(int projectId,  
    int modelId,  
    int restartId)
```

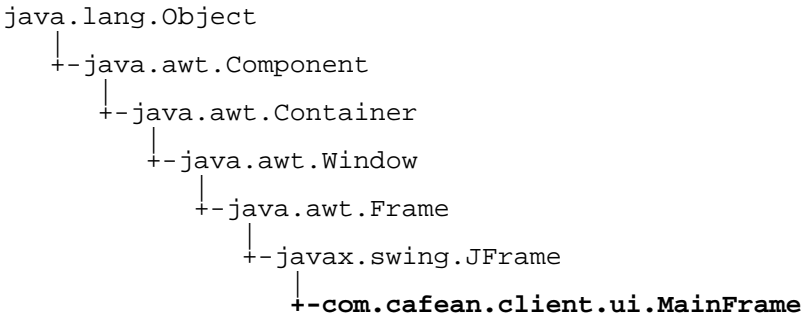
Sets the initial project, model and restart id's. NOTE: Does nothing when called with an invalid restart id.

#### Parameters:

projectId - the project dbid of the default selected model  
modelId - the dbid of the default selected model  
restartId - the dbid of the default selected restart run

com.cafean.client.ui

Class MainFrame



All Implemented Interfaces:

com.cafean.utils.SDITopFrame, java.awt.datatransfer.ClipboardOwner, com.cafean.utils.BatchProcessor, com.cafean.utils.SplashListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, java.awt.MenuContainer, RootPaneContainer, javax.accessibility.Accessible, WindowConstants

public class **MainFrame**  
extends JFrame  
implements WindowConstants, javax.accessibility.Accessible, RootPaneContainer,  
java.awt.MenuContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver,  
java.awt.MenuContainer, java.io.Serializable, com.cafean.utils.SplashListener,  
com.cafean.utils.BatchProcessor, java.awt.datatransfer.ClipboardOwner, com.cafean.utils.SDITopFrame

The MainFrame is the central main class for the ModelEditor.

| Field Summary    |  |
|------------------|--|
| static boolean   | debug<br>Display all debug information to the standard output                      |
| static MainFrame | instance<br>This is a static reference to the current instance of the ModelEditor. |
| static final int | MODE_CLEAR_CONSTANTS<br>THis is clears constant references in clone<br>Value: 3    |
| static final int | MODE_EXPORT_ASCII<br>This is exporting an ASCII file mode<br>Value: 2              |
| static final int | MODE_IMPORT<br>This clips all values less then -1e30 to -1e29<br>Value: 4          |

|                                |  |
|--------------------------------|--|
| static final int               | MODE_NORMAL<br><br>This is normal operational mode<br>Value: 0   |
| static final int               | MODE_SAVE_MED<br><br>This is saving a MED file mode<br>Value: 1  |
| static final int               | SELECTOR_OPEN<br><br>A file selector mode for openFileSelector that indicates an Open type<br>Value: 1 |
| static final int               | SELECTOR_SAVE<br><br>A file selector mode for openFileSelector that indicates a Save type<br>Value: 2  |
| com.cafean.utils.Snap<br>Corba | snapCorba<br><br>The CORBA Interface to the SNAP database.   |

**Fields inherited from class** javax.swing.JFrame

EXIT\_ON\_CLOSE

**Fields inherited from class** java.awt.Frame

CROSSHAIR\_CURSOR, DEFAULT\_CURSOR, E\_RESIZE\_CURSOR, HAND\_CURSOR, ICONIFIED, MAXIMIZED\_BOTH, MAXIMIZED\_HORIZ, MAXIMIZED\_VERT, MOVE\_CURSOR, N\_RESIZE\_CURSOR, NE\_RESIZE\_CURSOR, NORMAL, NW\_RESIZE\_CURSOR, S\_RESIZE\_CURSOR, SE\_RESIZE\_CURSOR, SW\_RESIZE\_CURSOR, TEXT\_CURSOR, W\_RESIZE\_CURSOR, WAIT\_CURSOR

**Fields inherited from class** java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |   |
|--------|---|
| public | MainFrame(boolean showSplash)<br><br>This constructs a new instance of the ModelEditor. |
|--------|---|

## Method Summary

|      |  |
|------|--|
| void | activateRedoButton(String name,boolean set)<br><br>This activates the Undo button, which will allow the user to redo a change that has been undone inside the ModelEditor. |
| void | activateUndoButton(String name,boolean set)<br><br>This activates the Undo button, which will allow the user to undo a change inside the ModelEditor.                      |

|             |  |
|-------------|--|
| static void | addAbstractModel(AbstractModel model)<br>Adds a new AbstractModel to the ModelEditor.  |
| static void | addAbstractModel(AbstractModel model,String label)<br>Adds a new AbstractModel to the model editor.  |
| static void | addContextHelp(java.awt.Component comp,String label)<br>Adds a component to the context sensitive help, with the given label.  |
| void        | addDrawnViewTab(DrawnView view)<br>Adds the given DrawnView to the tabbed pane holding all DrawnViews in Single Window mode.   |
| static void | addExportItem(JMenuItem item)<br>This inserts a new export option into the export menu.  |
| static void | addImportItem(JMenuItem item)<br>This inserts a new import option into the import menu.  |
| static void | addMenuItem(JMenuItem item,String name)<br>This is used to add a new menu item to the model editor main frame menu bar.  |
| static int  | addMessage(String text)<br>Adds a new message to the MessageWindow.  |
| static void | addMessage(String text,AbstractComponent comp)<br>Adds the given text to the message window as a notice associated with the given component.   |
| static void | addMessage(String text,AbstractComponent comp,int severityCode)<br>Adds the given text as a message to the message window associated with the given component  |
| static void | addMessage(String text,GenericObject comp,int severityCode)<br>Adds the given text as a message to the message window associated with the given GenericObject.   |
| static int  | addMessage(String text,int severityCode)<br>Adds the given text as a message to the message window with the given severity code chosen from those in MessageWindow.  |
| static void | addMessage(String text,Vector comps,int severityCode)<br>Adds the given text as a multi-line message to the message window associated with the given components.   |
| static void | addMessage(String text,ViewComponent view,JComponent uiComp,int severityCode)<br>Adds the given text as a message to the Message Window associated with the given ViewComponent for the given GUI Component. |
| void        | addPlaybackPanel(com.cafean.client.anim.PlaybackButtonPanel panel)<br>Adds the given playback panel to the list of panels updated in #updatePlaybackButtons  |

|                                      |   |
|--------------------------------------|---|
| static void                          | addRegisteredDialog(java.awt.Dialog dialog, AbstractModel model)<br>Adds the given dialog to the MainFrame's list of registered child dialogs.  |
| static void                          | addSound(int soundType)<br>This plays a sound, depending on the enumerated sound type.  |
| static void                          | clearCheckErrors()<br>Sets the Message Window's error count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent getErrorCount or getWarningCount calls.     |
| static void                          | clearCheckWarnings()<br>Sets the Message Window's warning count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent getErrorCount or getWarningCount calls. |
| void                                 | closeAbstractModel(AbstractModel model)<br>Closes the given AbstractModel.  |
| void                                 | closeMessageWindow()<br>Closes the MessageWindow from an external source.   |
| void                                 | createModel(MECodePlugin plugin)<br>Creates a new AbstractModel for the given plugin.   |
| static String                        | dateStamp()<br>Gets a string that contains the current date.  |
| static void                          | deactivateRedoButtons()<br>disables the undoButton and redoButton   |
| static void                          | disableMainFrame()<br>Disables all mouse events for this MainFrame.   |
| static void                          | enableMainFrame()<br>Deactivates this MainFrame's glass pane thus allowing mouse events to pass through to the underlying components.   |
| static ClientCodePlugin              | findClientPlugin(String id)<br>Finds a client plugin by its ID.   |
| static <a href="#">AbstractModel</a> | findModelByIdent(int ident)<br>This finds the AbstractModel with the given ident.   |
| static <a href="#">MEPlugin</a>      | findPlugin(String id)<br>Finds a plugin by its ID.  |
| static <a href="#">AbstractModel</a> | getAbstractModelAt(int index)<br>Returns the AbstractModel at a specified index.  |
| static int                           | getAbstractModelCount()<br>Returns the number of models currently inside the calculation project.   |

|   |   |
|---|---|
| static<br>ClientCodePlugin                    | getClientPluginAt(int index)<br>Returns the plugin at the specified index into the client plugins vector.   |
| static int                                    | getClientPluginCount()<br>Returns the number of client plugins currently loaded.  |
| static<br>java.awt.datatransfer<br>.Clipboard | getClipboard()  |
| static <a href="#">MECodePlugin[]</a>         | getCodePlugins()<br>This returns all of the plugins currently loaded that extend MECodePlugin.  |
| static <a href="#">AbstractModel</a>          | getCurrentModel()<br>This returns the AbstractModel that, of the list of currently open models, is currently being worked on.   |
| static<br><a href="#">MEFeaturePlugin[]</a>   | getFeaturePlugins()<br>Returns an array of all the MEFeaturePlugins that are currently loaded.  |
| static<br><a href="#">MEFeaturePlugin[]</a>   | getFeaturePlugins(AbstractModel model)<br>Finds all of the { @link MEFeaturePlugin feature plugins } that have data that needs to be stored along with the given model. |
| static<br>com.cafean.utils.File<br>Chooser    | getFileChooser()<br>Gives access to a FileChooser for selecting a file.   |
| static int                                    | getMode()<br>Getter for property mode.  |
| static <a href="#">AbstractModel</a>          | getModel(String label)<br>This finds the AbstractModel from the list of open models that has been stored under a given label.   |
| static int                                    | getNumberOfErrorsFound()<br>Retrieves the number of error messages in the message window.   |
| static int                                    | getNumberOfWarningsFound()<br>Retrieves the number of warning messages in the message window.   |
| org.omg.CORBA.ORB                             | getOrb()<br>Gets the current CORBA orb.   |
| com.cafean.utils.Refe<br>renceDocs.PDFViewer  | getPdfViewer()<br>Getter for property pdfViewer.  |
| static <a href="#">MEPlugin</a>               | getPluginAt(int index)<br>Returns the plugin at the specified index into the plugins vector.  |
| static int                                    | getPluginCount()<br>Returnsn the number of plugins currently loaded.  |
| SnapPreferences                               | getPrefs()<br>Getter for the SnapPreferences object.  |



|                                      |  |
|--------------------------------------|--|
| static <a href="#">AbstractModel</a> | getReferenceModel(AbstractModel model)<br>Retrieves the reference model for the given model if one has been loaded.  |
| static Iterator                      | getRegisteredDialogs()<br>Returns an unmodifiable List of the registered dialogs.  |
| static Iterator                      | getRegisteredDialogs(AbstractModel model)<br>Returns an unmodifiable List of the registered dialogs for a given model  |
| static java.io.File                  | getSnapHomeDirectory()<br>Returns the installation directory for CAFEAN.   |
| int                                  | getSplashStatus()<br>  |
| String[]                             | getSplashTitle()<br>   |
| SnapUndoManager                      | getUndoManager()<br>Accessor for the SnapUndoManager that controls the undo-stack for the ModelEditor.   |
| UndoableEditSupport                  | getUndoSupport()<br>Accessor for the UndoableEditSupport for the ModelEditor.  |
| String                               | getUserName()<br>Gets the current user id.   |
| static String                        | getValidLabel(String label)<br>Returns a valid key string for the user defined key passed in.  |
| com.cafean.utils.Version             | getVersion()<br>This gets the Version information for this instance of the ModelEditor.  |
| void                                 | init()<br>Reads in the Preferences from the file.  |
| static void                          | loadReferenceModel(AbstractModel model, java.io.File file)<br>Loads the reference model for the given model from the given file.   |
| void                                 | lostOwnership(java.awt.datatransfer.Clipboard clipboard, java.awt.datatransfer.Transferable contents)<br>The ClipboardOwner interface routine.   |
| static void                          | main(String[] args)<br>This is the main function for the ModelEditor.  |
| static void                          | offsetWindowLocation(java.awt.Window childWindow)<br>Offsets the given childWindow from other windows of the same type so that they do not overlap each other perfectly, hiding one another.                             |
| java.io.File                         | openFileSelector(String title, int selectorType)<br>Opens a JFileChooser of the appropriate type configured with the given title and an appropriate button name and returns either the selected file or null on failure. |

|             |   |
|-------------|---|
| static void | openMessageArea()<br>Opens the MessageWindow, and updates the checkbox menu item for the MessageWindow to show that it is open.                                     |
| boolean     | processCommand(String command)  |
| static void | refreshSteamTable()<br>Refreshes the loaded steam tables.   |
| static void | removeAbstractModel(AbstractModel model)<br>Removes a AbstractModel from the model editor.  |
| void        | removeDrawnViewTab(DrawnView view)<br>Removes the given DrawnView from the tabbed pane holding all DrawnViews in Single Window mode.                                |
| void        | removePlaybackPanel(com.cafean.client.anim.PlaybackButtonPanel panel)<br>Removes the given playback panel from the list of panels updated in #updatePlaybackButtons |
| static void | removeRegisteredDialog(java.awt.Dialog dialog, AbstractModel model)<br>Removes the given dialog from the MainFrame's list of registered child dialogs.              |
| boolean     | requestSaveFileName(AbstractModel model)<br>Opens a file selection dialog for specifying the target location of a MED file.   |
| static void | resetAllUnits(int units)<br>Resets all the renderers for units inside the model editor.   |
| static void | resetCursor()<br>Sets the cursor to the default cursor when the process requiring the waiting is complete.  |
| void        | resetWindowingMode()<br>Updates and resets all Window Arrangement based on the current status.  |
| void        | setChildLocation(java.awt.Window childWindow)<br>Sets the given window's location to be that of the center of this main frame's current screen.                     |
| void        | setCurrentModel(AbstractModel m)<br>This sets the AbstractModel from the list of open models that currently is being worked on.                                     |
| static void | setFileChooserLocation(JFileChooser dlg)<br>Sets the given JFileChooser's location to be the center of the main frame's current screen.                             |
| static void | setMode(int mode_)<br>Setter for property mode.   |
| static void | setPlaybackTime(String mess)<br>Sets the currently display status message   |
| void        | setSplashStatus(int stat)   |

|             |  |
|-------------|--|
| void        | setUsername(String name)<br>Sets the current user id.  |
| void        | setVisible(boolean visible)<br>Shows or hides this MainFrame.  |
| static void | setWaitCursor()<br>Sets the cursor to a cursor indicating the user should wait for a process before continuing.                                      |
| static void | setWindowLocation(java.awt.Window window)<br>Sets the given window's location to be that of the center of this main frame's current screen.          |
| static void | showCreateDialog()<br>Attempts to create and add a new model by prompting the user with the list of available plugins.                               |
| static void | showOpenDialog()<br>This event is called when the user wants to open a locally stored SAM file.  |
| void        | toFrontDrawnViewTab(DrawnView view)<br>Sets the given view as the selected tab in tabbed pane holding all DrawnViews in single window mode.          |
| void        | updatePlaybackButtons()<br>Enables or disables the pause, play and time buttons based on the current state of the Source Manger and it's master run. |

#### Methods inherited from class javax.swing.JFrame

getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, update

#### Methods inherited from class java.awt.Frame

addNotify, getAccessibleContext, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setIconImage, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated

#### Methods inherited from class java.awt.Window

addNotify, addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getAccessibleContext, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, show, toBack, toFront

**Methods inherited from class java.awt.Container**

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

**Methods inherited from class java.awt.Component**

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

**Methods inherited from class java.lang.Object**

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

## Fields

### instance

```
public static com.cafean.client.ui.MainFrame instance
```

This is a static reference to the current instance of the ModelEditor.

### debug

```
public static boolean debug
```

Display all debug information to the standard output

### SELECTOR\_OPEN

```
public static final int SELECTOR_OPEN
```

A file selector mode for openFileSelector that indicates an Open type

### SELECTOR\_SAVE

```
public static final int SELECTOR_SAVE
```

A file selector mode for openFileSelector that indicates a Save type

### snapCorba

```
public transient com.cafean.utils.SnapCorba snapCorba
```

The CORBA Interface to the SNAP database.

### MODE\_NORMAL

```
public static final int MODE_NORMAL
```

This is normal operational mode

### MODE\_SAVE\_MED

```
public static final int MODE_SAVE_MED
```

This is saving a MED file mode

### MODE\_EXPORT\_ASCII

```
public static final int MODE_EXPORT_ASCII
```

This is exporting an ASCII file mode

### MODE\_CLEAR\_CONSTANTS

```
public static final int MODE_CLEAR_CONSTANTS
```

This is clears constant references in clone

### MODE\_IMPORT

```
public static final int MODE_IMPORT
```

This clips all values less then -1e30 to -1e29

(continued from last page)

## Constructors

### MainFrame

```
public MainFrame(boolean showSplash)
```

This constructs a new instance of the ModelEditor. The new instance can be accessed from the public static reference `#instance`. As the main controller class for the ModelEditor, this reads in all the plugins, loads all user preferences, and returns the ModelEditor to the visual state of the last time the user closed the ModelEditor.

**Parameters:**

`showSplash` - the boolean that is TRUE if the splashscreen should be shown.

## Methods

### getClipboard

```
public static java.awt.datatransfer.Clipboard getClipboard()
```

---

### resetWindowingMode

```
public void resetWindowingMode()
```

Updates and resets all Window Arrangement based on the current status.

---

### addDrawnViewTab

```
public void addDrawnViewTab(DrawnView view)
```

Adds the given DrawnView to the tabbed pane holding all DrawnViews in Single Window mode. Note: Does nothing in multiple window mode

**Parameters:**

`view` - the DrawnView to include in the tab pane

---

### removeDrawnViewTab

```
public void removeDrawnViewTab(DrawnView view)
```

Removes the given DrawnView from the tabbed pane holding all DrawnViews in Single Window mode. Note: Does nothing in multiple window mode

**Parameters:**

`view` - the DrawnView to remove from the tab pane

---

### toFrontDrawnViewTab

```
public void toFrontDrawnViewTab(DrawnView view)
```

Sets the given view as the selected tab in tabbed pane holding all DrawnViews in single window mode. Note: Does nothing in multiple window mode

**Parameters:**

`view` - the DrawnView to select

---

### setSplashStatus

```
public void setSplashStatus(int stat)
```

---

## getSplashStatus

```
public int getSplashStatus()
```

---

## getSplashTitle

```
public String[] getSplashTitle()
```

---

## getVersion

```
public com.cafean.utils.Version getVersion()
```

This gets the Version information for this instance of the ModelEditor.

**Returns:**

the Version that contains the current release information.

---

## closeMessageWindow

```
public void closeMessageWindow()
```

Closes the MessageWindow from an external source. This ensures that when the MessageWindow is closed, the menu item is updated to show that the MessageWindow is not currently visible.

**See Also:**

MessageWindow

---

## closeAbstractModel

```
public void closeAbstractModel(AbstractModel model)
```

Closes the given AbstractModel. Closes all the dialog windows for that model, removes it from the Navigator, and frees all the data for garbage collection.

**Parameters:**

model - the AbstractModel to be closed.

---

## getCurrentModel

```
public static AbstractModel getCurrentModel()
```

This returns the AbstractModel that, of the list of currently open models, is currently being worked on.

**Returns:**

the AbstractModel that is currently being edited.

---

## setCurrentModel

```
public void setCurrentModel(AbstractModel m)
```

This sets the AbstractModel from the list of open models that currently is being worked on.

**Parameters:**

m - the AbstractModel.

---

(continued from last page)

---

## getModel

```
public static AbstractModel getModel(String label)
```

This finds the AbstractModel from the list of open models that has been stored under a given label.

**Parameters:**

label - the String label given by the user.

**Returns:**

the AbstractModel stored with the given label.

---

## getUndoManager

```
public SnapUndoManager getUndoManager()
```

Accessor for the SnapUndoManager that controls the undo-stack for the ModelEditor.

**Returns:**

the SnapUndoManager for this instance of the ModelEditor.

---

## getUndoSupport

```
public UndoableEditSupport getUndoSupport()
```

Accessor for the UndoableEditSupport for the ModelEditor.

**Returns:**

the UndoableEditSupport for this instance of the ModelEditor.

---

## resetAllUnits

```
public static void resetAllUnits(int units)
```

Resets all the renderers for units inside the model editor.

---

## showCreateDialog

```
public static void showCreateDialog()
```

Attempts to create and add a new model by prompting the user with the list of available plugins.

---

## createModel

```
public void createModel(MECodePlugin plugin)
```

Creates a new AbstractModel for the given plugin. This also creates a new ViewComponent for the model, and opens it.

---

## processCommand

```
public boolean processCommand(String command)
```

---

## showOpenDialog

```
public static void showOpenDialog()
```

This event is called when the user wants to open a locally stored SAM file.

**Parameters:**

event - the ActionEvent that fired this action.

---



---

## requestSaveFileName

```
public boolean requestSaveFileName(AbstractModel model)
```

Opens a file selection dialog for specifying the target location of a MED file. This is used when saving a model for the first time, or when using the Save As option.

**Parameters:**

model - the AbstractModel that is being saved.

**Returns:**

TRUE unless the user cancels the file selection.

---

## activateUndoButton

```
public void activateUndoButton(String name,  
                               boolean set)
```

This activates the Undo button, which will allow the user to undo a change inside the ModelEditor. This is a global undo stack, so all changes in the ModelEditor are stored in one undo stack.

**Parameters:**

name - the String name for the action that will be undone.

set - the boolean for enabling or disabling the UndoButton.

---

## activateRedoButton

```
public void activateRedoButton(String name,  
                               boolean set)
```

This activates the Undo button, which will allow the user to redo a change that has been undone inside the ModelEditor. This is a global undo stack, so all changes in the ModelEditor are stored in one undo stack.

**Parameters:**

name - the String name for the action that will be redone.

set - the boolean for enabling or disabling the re-do button.

---

## addSound

```
public static void addSound(int soundType)
```

This plays a sound, depending on the enumerated sound type.

**Parameters:**

soundType - The enumerated type of the sound to play. \* Valid types are: \* 0 - Chime / AlertSound\* 1 - Boing / InfoSound\* 2 - Cowbell / WarningSound\* 3 - UhOh / UserErrSound\* 4 - Gong / IntErrSound\* 5 - Swish / ConnectSound\* 6 - Cork / DisconnectSound\* \* NOTE: Works for JDK 1.3 and above. \* @param soundType the sound to play.

---

## getNumberOfErrorsFound

```
public static int getNumberOfErrorsFound()
```

Retrieves the number of error messages in the message window.

**Returns:**

the int number of errors from the most recent error check.

---

## getNumberOfWarningsFound

```
public static int getNumberOfWarningsFound()
```

---

(continued from last page)

Retrieves the number of warning messages in the message window.

**Returns:**

the int number of warnings from the most recent error check.

---

## addMessage

```
public static int addMessage(String text)
```

Adds a new message to the MessageWindow.

**Parameters:**

text - the String to be added to the MessageWindow.

**Returns:**

the return flag from adding a message to the MessageWindow.

**See Also:**

MessageWindow

---

## addMessage

```
public static int addMessage(String text,  
                             int severityCode)
```

Adds the given text as a message to the message window with the given severity code chosen from those in MessageWindow.

**Parameters:**

text - the String to be added to the MessageWindow

severityCode - the MessageWindow based enumeration for the type of message.

**Returns:**

the int return flag from the MessageWindow.

**See Also:**

MessageWindow

---

## addMessage

```
public static void addMessage(String text,  
                              AbstractComponent comp)
```

Adds the given text to the message window as a notice associated with the given component.

**Parameters:**

text - the String to be added to the MessageWindow

comp - the AbstractComponent.

**See Also:**

MessageWindow

---

## addMessage

```
public static void addMessage(String text,  
                              AbstractComponent comp,  
                              int severityCode)
```

Adds the given text as a message to the message window associated with the given component

**Parameters:**

text - the String to be added to the MessageWindow

---

(continued from last page)

comp - the AbstractComponent.  
severityCode - the severity code chosen from those in MessageWindow.

**See Also:**

MessageWindow

---

## addMessage

```
public static void addMessage(String text,  
    ViewComponent view,  
    JComponent uiComp,  
    int severityCode)
```

Adds the given text as a message to the Message Window associated with the given ViewComponent for the given GUI Component.

**Parameters:**

text - the String to be added to the MessageWindow  
view - the ViewComponent  
uiComp - the JComponent that the given message refers to  
severityCode - the severity code chosen from those in MessageWindow.

**See Also:**

MessageWindow

---

## addMessage

```
public static void addMessage(String text,  
    GenericObject comp,  
    int severityCode)
```

Adds the given text as a message to the message window associated with the given GenericObject.

**Parameters:**

text - the String to be added to the MessageWindow  
comp - the GenericObject.  
severityCode - the severity code chosen from those in MessageWindow.

**See Also:**

MessageWindow

---

## addMessage

```
public static void addMessage(String text,  
    Vector comps,  
    int severityCode)
```

Adds the given text as a multi-line message to the message window associated with the given components.

**Parameters:**

text - the String to be added to the MessageWindow  
comps - the Vector of components.  
severityCode - the severity code chosen from those in MessageWindow.

**See Also:**

MessageWindow

---

## openMessageArea

```
public static void openMessageArea()
```

Opens the MessageWindow, and updates the checkbox menu item for the MessageWindow to show that it is open.

---

## clearCheckErrors

```
public static void clearCheckErrors()
```

Sets the Message Window's error count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent `getErrorCount` or `getWarningCount` calls.

---

## clearCheckWarnings

```
public static void clearCheckWarnings()
```

Sets the Message Window's warning count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent `getErrorCount` or `getWarningCount` calls.

---

## getAbstractModelAt

```
public static AbstractModel getAbstractModelAt(int index)
```

Returns the `AbstractModel` at a specified index.

**Parameters:**

`index` - the index into the models vector.

**Returns:**

the `AbstractModel` at the specified index.

---

## addAbstractModel

```
public static void addAbstractModel(AbstractModel model)
```

Adds a new `AbstractModel` to the `ModelEditor`. **Note:**The model will be added to the Navigator on the Swing event thread.

**Parameters:**

`model` - the `AbstractModel`.

**See Also:**

```
.addAbstractModel(AbstractModel,String)()
```

---

## addAbstractModel

```
public static void addAbstractModel(AbstractModel model,  
String label)
```

Adds a new `AbstractModel` to the model editor. Sets this model as the currently selected model, and refreshes the Navigator. **Note:**The model will be added to the Navigator on the Swing event thread.

**Parameters:**

`label` - the String label for this model, if NULL the next available is assigned.

`model` - the `AbstractModel` to add

---

## removeAbstractModel

```
public static void removeAbstractModel(AbstractModel model)
```

Removes a `AbstractModel` from the model editor. If this was the currently selected model, and there is another model, the first of the other models is set selected. Otherwise, it is left null

**Parameters:**

`model` - the `AbstractModel`.

---

(continued from last page)

---

## findModelByIdent

```
public static AbstractModel findModelByIdent(int ident)
```

This finds the [AbstractModel](#) with the given ident. A Model's ident is given to it when it is opened inside the [ModelEditor](#). This allows the [ModelEditor](#) to keep the open models separate.

**Parameters:**

ident - the ident of the model.

**Returns:**

the [AbstractModel](#) with the given ident, or NULL if not found.

---

## getReferenceModel

```
public static AbstractModel getReferenceModel(AbstractModel model)
```

Retrieves the reference model for the given model if one has been loaded.

**Parameters:**

model - the [AbstractModel](#) to retrieve a reference model for.

**Returns:**

an [AbstractModel](#) reference to the given model's reference model.

---

## loadReferenceModel

```
public static void loadReferenceModel(AbstractModel model,  
    java.io.File file)
```

Loads the reference model for the given model from the given file.

**Parameters:**

model - the [AbstractModel](#) to load a reference model for.

file - the [File](#) to load the reference model from.

**Returns:**

an [AbstractModel](#) reference to the given model's reference model.

---

## getAbstractModelCount

```
public static int getAbstractModelCount()
```

Returns the number of models currently inside the calculation project.

**Returns:**

the size of the models vector.

---

## dateStamp

```
public static String dateStamp()
```

Gets a string that contains the current date.

**Returns:**

the String containing the current date and time.

---

## setWaitCursor

```
public static void setWaitCursor()
```

Sets the cursor to a `CURSOR_WAIT` indicating the user should wait for a process before continuing.

---

(continued from last page)

**See Also:**

Cursor

---

## resetCursor

```
public static void resetCursor()
```

Sets the cursor to the default cursor when the process requiring the waiting is complete.

**See Also:**

Cursor

---

## disableMainFrame

```
public static void disableMainFrame()
```

Disables all mouse events for this MainFrame. Activates the glass pane within this MainFrame and consumes all mouse events before they can be delivered to the underlying components.

---

## enableMainFrame

```
public static void enableMainFrame()
```

Deactivates this MainFrame's glass pane thus allowing mouse events to pass through to the underlying components.

---

## refreshSteamTable

```
public static void refreshSteamTable()
```

Refreshes the loaded steam tables.

---

## deactivateRedoButtons

```
public static void deactivateRedoButtons()
```

disables the undoButton and redoButton

---

## setVisible

```
public void setVisible(boolean visible)
```

Shows or hides this MainFrame.

**Parameters:**

`visible` - if true, a thread will be started to load the steam tables.

---

## openFileSelector

```
public java.io.File openFileSelector(String title,  
int selectorType)
```

Opens a JFileChooser of the appropriate type configured with the given title and an appropriate button name and returns either the selected file or null on failure.

**Parameters:**

`title` - a String containing the desired dialog title.

`selectorType` - one of the `SELECTOR_*` enumerated integers. Used to determine which 'last' property to track and how to configure the dialog.

**Returns:**

the File selected by the user, or NULL if it was cancelled.

## setWindowLocation

```
public static void setWindowLocation(java.awt.Window window)
```

Sets the given window's location to be that of the center of this main frame's current screen.

**Parameters:**

window - the Window to set location for.

---

## setChildLocation

```
public void setChildLocation(java.awt.Window childWindow)
```

Sets the given window's location to be that of the center of this main frame's current screen.

**Parameters:**

childWindow - the Window to set location for.

---

## offsetWindowLocation

```
public static void offsetWindowLocation(java.awt.Window childWindow)
```

Offsets the given childWindow from other windows of the same type so that they do not overlap each other perfectly, hiding one another. Note: The given window must be registered first with `addRegisteredDialog`.

---

## setFileChooserLocation

```
public static void setFileChooserLocation(JFileChooser dlg)
```

Sets the given JFileChooser's location to be the center of the main frame's current screen. This is primarily used by the Plugins to open FileChoosers in the center of the screen.

**Parameters:**

dlg - the JFileChooser that has just been opened.

---

## lostOwnership

```
public void lostOwnership(java.awt.datatransfer.Clipboard clipboard,  
    java.awt.datatransfer.Transferable contents)
```

The ClipboardOwner interface routine. Implemented as a noop here.

**Parameters:**

clipboard - the Clipboard.

contents - the Transferable contents on the clipboard.

---

## init

```
public void init()
```

Reads in the Preferences from the file.

---

## getPrefs

```
public SnapPreferences getPrefs()
```

Getter for the SnapPreferences object.

**Returns:**

prefs.

---

(continued from last page)

---

## setUserName

```
public void setUserName(String name)
```

Sets the current user id.

**Parameters:**

name - the String user id.

---

## getUserName

```
public String getUserName()
```

Gets the current user id.

**Returns:**

the String user id.

---

## main

```
public static void main(String[] args)
```

This is the main function for the ModelEditor. It spawns a new instance of the MainFrame, opens all the Corba objects, and handles all of the commandline arguments.

**Parameters:**

args - the String[] command line arguments. Acceptable arguments are: --version : prints out the current version of the ModelEditor.--usage & --help : prints out the command line help.--debug : Turns on debugging.--nosplash : Turns off the splash screen.--userid : Sets the current userid to the next argument.--batch : Opens the batchfile contained in the next argument.

---

## getOrb

```
public org.omg.CORBA.ORB getOrb()
```

Gets the current CORBA orb. If that ORB does not exist, it creates it. This is needed for submitting jobs.

**Returns:**

the org.omg.CORBA.ORB object.

**See Also:**

org.omg.CORBA.ORB

---

## addExportItem

```
public static void addExportItem(JMenuItem item)
```

This inserts a new export option into the export menu. This allows MEPlugins to add plugin specific export actions to the MainFrame's menu.

**Parameters:**

item - the JMenuItem being added.

---

## addImportItem

```
public static void addImportItem(JMenuItem item)
```

This inserts a new import option into the import menu. This allows MEPlugins to add plugin specific import actions to the MainFrame's menu.

**Parameters:**

item - the JMenuItem being added.

---



## addMenuItem

```
public static void addMenuItem(JMenuItem item,  
    String name)
```

This is used to add a new menu item to the model editor main frame menu bar.

**Parameters:**

`item` - This JMenuItem is about to be inserted into a menu.  
`name` - the Name of the menu where the item is to be inserted. Allowed Names are:  
FILEEDITVIEWTOOLSWINDOWHELP

---

## findClientPlugin

```
public static ClientCodePlugin findClientPlugin(String id)
```

Finds a client plugin by its ID.

**Parameters:**

`id` - the String id of the plugin being looked for.

**Returns:**

the ClientCodePlugin with the given id; or null if not found

---

## getClientPluginAt

```
public static ClientCodePlugin getClientPluginAt(int index)
```

Returns the plugin at the specified index into the client plugins vector.

**Parameters:**

`index` - the index of the desired client plugin

**Returns:**

the ClientCodePlugin at the specified index.

---

## getClientPluginCount

```
public static int getClientPluginCount()
```

Returns the number of client plugins currently loaded.

**Returns:**

the number of client plugins available

---

## findPlugin

```
public static MEPlugin findPlugin(String id)
```

Finds a plugin by its ID. If the plugin is a MECodePlugin, the plugin ID should be the same as the { @link AbstractModel#getPluginId code name } of an AbstractModel built by that plugin.

**Parameters:**

`id` - the String id of the plugin being looked for.

**Returns:**

the MEPlugin with id for its PluginId, or null if there isn't one.

---

(continued from last page)

---

## getPluginAt

```
public static MEPlugin getPluginAt(int index)
```

Returns the plugin at the specified index into the plugins vector.

**Parameters:**

index - the index into the plugin vector.

**Returns:**

the MEPlugin at the specified index.

---

## getPluginCount

```
public static int getPluginCount()
```

Returns the number of plugins currently loaded.

**Returns:**

the size of the plugins vector.

---

## getCodePlugins

```
public static MECodePlugin\[\] getCodePlugins()
```

This returns all of the plugins currently loaded that extend MECodePlugin.

**Returns:**

A MECodePlugin[] containing all the currently loaded code plugins.

---

## getValidLabel

```
public static String getValidLabel(String label)
```

Returns a valid key string for the user defined key passed in. If NULL is passed in, this will return the next available key. If the key is invalid, this will return NULL.

**Parameters:**

label - See Above

**Returns:**

See Above

---

## getFeaturePlugins

```
public static MEFeaturePlugin\[\] getFeaturePlugins(AbstractModel model)
```

Finds all of the {@link MEFeaturePlugin feature plugins} that have data that needs to be stored along with the given model. This makes use of the MEFeaturePlugin#isAssociated function on all of the current MEFeaturePlugins.

**Parameters:**

model - the AbstractModel that is about to be stored.

**Returns:**

the MEFeaturePlugin[] of plugins associated with the model.

---

## getFeaturePlugins

```
public static MEFeaturePlugin\[\] getFeaturePlugins()
```

Returns an array of all the MEFeaturePlugins that are currently loaded.

---

(continued from last page)

**Returns:**

the current MEFeaturePlugins.

---

## getSnapHomeDirectory

```
public static java.io.File getSnapHomeDirectory()
```

Returns the installation directory for CAFEAN. This is either passed to the VM by a -DCAFEAN\_HOME="dir" or its the current working directory.

**Returns:**

the File object containing the absolute path to the current working component.

---

## getFileChooser

```
public static com.cafean.utils.FileChooser getFileChooser()
```

Gives access to a FileChooser for selecting a file. This should be used by any plugin that attempts to load or store information in a file. The dialog created by this FileChooser always appears at the center of the screen.

**Returns:**

the FileChooser created by the ModelEditor.

---

## addRegisteredDialog

```
public static void addRegisteredDialog(java.awt.Dialog dialog,  
    AbstractModel model)
```

Adds the given dialog to the MainFrame's list of registered child dialogs. Registered child dialogs appear in the Windows menu and have their location offset from other registered windows by #setWindowLocation.

**Parameters:**

dialog - the Dialog that will be added to the window list.  
model - the AbstractModel that the given dialog is related to

---

## getRegisteredDialogs

```
public static Iterator getRegisteredDialogs()
```

Returns an unmodifiable List of the registered dialogs.

---

## getRegisteredDialogs

```
public static Iterator getRegisteredDialogs(AbstractModel model)
```

Returns an unmodifiable List of the registered dialogs for a given model

---

## removeRegisteredDialog

```
public static void removeRegisteredDialog(java.awt.Dialog dialog,  
    AbstractModel model)
```

Removes the given dialog from the MainFrame's list of registered child dialogs.

**Parameters:**

dialog - the Dialog that will be removed from the window list.

**See Also:**

.addRegisteredDialog()

(continued from last page)

---

## getMode

```
public static int getMode()
```

Getter for property mode.

**Returns:**

Value of property mode.

---

## setMode

```
public static void setMode(int mode_)
```

Setter for property mode.

**Parameters:**

mode - New value of property mode.

---

## getPdfViewer

```
public com.cafean.utils.ReferenceDocs.PDFViewer getPdfViewer()
```

Getter for property pdfViewer.

**Returns:**

Value of property pdfViewer.

---

## updatePlaybackButtons

```
public void updatePlaybackButtons()
```

Enables or disables the pause, play and time buttons based on the current state of the Source Manger and it's master run.

---

## setPlaybackTime

```
public static void setPlaybackTime(String mess)
```

Sets the currently display status message

**Parameters:**

mess - A String containing the new message

---

## addContextHelp

```
public static void addContextHelp(java.awt.Component comp,  
    String label)
```

Adds a component to the context sensitive help, with the given label.

---

## addPlaybackPanel

```
public void addPlaybackPanel(com.cafean.client.anim.PlaybackButtonPanel panel)
```

Adds the given playback panel to the list of panels updated in #updatePlaybackButtons

---

## removePlaybackPanel

```
public void removePlaybackPanel(com.cafean.client.anim.PlaybackButtonPanel panel)
```

Removes the given playback panel from the list of panels updated in #updatePlaybackButtons

---

## com.cafean.client.ui

# Class MessageWindow

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JPanel
          |-- com.cafean.client.ui.MessageWindow

```

public class **MessageWindow**  
 extends JPanel

A panel implementation of a centralized message window.

### Field Summary

|                  |   |
|------------------|---|
| static final int | AlertMsg<br>Message type for non-error messages that the user should be alerted to.<br>Value: 0 |
| static final int | AlertSound<br>Sound type for an alert event<br>Value: 0   |
| static final int | ConnectSound<br>Sound type for a connection event<br>Value: 5                                   |
| static final int | DisconnectSound<br>Sound type for a disconnection event<br>Value: 6                             |
| static final int | InfoMsg<br>Message type for normal messages.<br>Value: 1  |
| static final int | InfoSound<br>Sound type for an information event<br>Value: 1                                    |
| static final int | InternalErrMsg<br>Message type for errors that are not user created or caused.<br>Value: 4      |
| static final int | InternalErrSound<br>Sound type for an internal error event<br>Value: 4                          |

|                               |  |
|-------------------------------|--|
| <code>static final int</code> | <b>NoSound</b><br>Sound type for the default case<br>Value: <b>-1</b>  |
| <code>static final int</code> | <b>UserErrorMsg</b><br>Message type for errors that are considered user created or caused<br>Value: <b>3</b> |
| <code>static final int</code> | <b>UserErrorSound</b><br>Sound type for a user error event<br>Value: <b>3</b>                                |
| <code>static final int</code> | <b>WarningMsg</b><br>Message type for non-fatal errors or warnings<br>Value: <b>2</b>                        |
| <code>static final int</code> | <b>WarningSound</b><br>Sound type for a warning event<br>Value: <b>2</b>                                     |

**Fields inherited from class** `javax.swing.JComponent`

`TOOL_TIP_TEXT_KEY`, `UNDEFINED_CONDITION`, `WHEN_ANCESTOR_OF_FOCUSED_COMPONENT`, `WHEN_FOCUSED`, `WHEN_IN_FOCUSED_WINDOW`

**Fields inherited from class** `java.awt.Component`

`BOTTOM_ALIGNMENT`, `CENTER_ALIGNMENT`, `LEFT_ALIGNMENT`, `RIGHT_ALIGNMENT`, `TOP_ALIGNMENT`

## Constructor Summary

|                     |   |
|---------------------|---|
| <code>public</code> | <b>MessageWindow()</b><br>Creates new form <code>MessageDialog</code> . |
|---------------------|---|

## Method Summary

|                   |   |
|-------------------|---|
| <code>int</code>  | <b>addMessage(String message)</b><br>Adds the given message as type <code>InfoMsg</code> .  |
| <code>void</code> | <b>addMessage(String text, AbstractComponent comp)</b><br>Adds the given message as type <code>InfoMsg</code> with the attached component.                        |
| <code>void</code> | <b>addMessage(String text, AbstractComponent comp, int severityCode)</b><br>Adds the given message as type <code>severityCode</code> with the attached component. |
| <code>void</code> | <b>addMessage(String text, GenericObject object, int severityCode)</b><br>Adds the given message as type <code>severityCode</code> with the attached component.   |

|                     |   |
|---------------------|---|
| int                 | addMessage(String text,int severityCode)<br>Adds the given message with the given severity code   |
| void                | addMessage(String text,Vector comps,int severityCode)<br>Adds the given message as type severityCode with the attached components.  |
| void                | addMessage(String text,ViewComponent comp,int severityCode,JComponent uiComp)<br>Adds the given message as type severityCode in the given View for the given UI Component.  |
| void                | addSound(int soundType)<br>Attempts to play a sound of the given type. Valid types are: 0 - Chime / AlertSound1 - Boing / InfoSound2 - Cowbell / WarningSound3 - UhOh / UserErrSound4 - Gong / IntErrSound5 - Swish / ConnectSound6 - Cork / DisconnectSound NOTE: Works for JDK 1.3 and above. |
| void                | clearCheckErrors()<br>Sets the error count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent getErrorCount or getWarningCount calls.  |
| void                | clearWarningCount()<br>Sets the error count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent getErrorCount or getWarningCount calls.   |
| String              | dateStamp()<br>Creates a formatted date stamp of the current time/date.   |
| int                 | getNumberOfErrorsFound()<br>Retrieves the number of error messages since the last clearWarningCount or clearCheckErrors call.   |
| java.io.PrintStream | getOutputStream()<br>Retrieves the PrintStream that this MessageWindow sends it's alternate output stream to.   |
| int                 | getWarningCount()<br>Retrieves the number of warning messages since the last clearWarningCount or clearCheckErrors call.  |
| void                | selectComp(AbstractComponent comp,boolean clear)<br><b>Deprecated.</b>  |
| void                | setOutputStream(java.io.PrintStream out)<br>Sets the PrintStream that this MessageWindow sends it's alternate output stream to.   |
| void                | setupContextHelp()<br>Sets up the context sensitive help on the Message Window  |
| void                | toFront()   |

**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

**Methods inherited from class java.awt.Component**



action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### AlertMsg

public static final int **AlertMsg**

Message type for non-error messages that the user should be alerted to.

### InfoMsg

public static final int **InfoMsg**

Message type for normal messages.

### WarningMsg

public static final int **WarningMsg**

(continued from last page)

Message type for non-fatal errors or warnings

---

## UserErrorMsg

public static final int **UserErrorMsg**

Message type for errors that are considered user created or caused

---

## InternalErrMsg

public static final int **InternalErrMsg**

Message type for errors that are not user created or caused.

---

## NoSound

public static final int **NoSound**

Sound type for the default case

---

## AlertSound

public static final int **AlertSound**

Sound type for an alert event

---

## InfoSound

public static final int **InfoSound**

Sound type for an information event

---

## WarningSound

public static final int **WarningSound**

Sound type for a warning event

---

## UserErrorSound

public static final int **UserErrorSound**

Sound type for a user error event

---

## InternalErrSound

public static final int **InternalErrSound**

Sound type for an internal error event

---

## ConnectSound

public static final int **ConnectSound**

Sound type for a connection event

---

## DisconnectSound

public static final int **DisconnectSound**

Sound type for a disconnection event

---

## Constructors

(continued from last page)

## MessageWindow

```
public MessageWindow( )
```

Creates new form MessageDialog. This should only be done by the MainFrame, when a new instance of the ModelEditor is started. The MessageWindow is never Modal, and always uses the current instance of the MainFrame as a parent.

## Methods

### setupContextHelp

```
public void setupContextHelp( )
```

Sets up the context sensitive help on the Message Window

### setOutputStream

```
public void setOutputStream( java.io.PrintStream out )
```

Sets the PrintStream that this MessageWindow sends it's alternate output stream to. The alternate output stream can be turned off by using setOutputStream(null) and defaults to System.err.

**Parameters:**

out - the PrintStream to print alternate output to.

### getOutputStream

```
public java.io.PrintStream getOutputStream( )
```

Retrieves the PrintStream that this MessageWindow sends it's alternate output stream to. The alternate output stream can be turned off by using setOutputStream(null) and defaults to System.err.

### clearWarningCount

```
public void clearWarningCount( )
```

Sets the error count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent getErrorCount or getWarningCount calls.

### clearCheckErrors

```
public void clearCheckErrors( )
```

Sets the error count beginning index to the number of messages currently in the table so that they will not be counted as errors or warnings in subsequent getErrorCount or getWarningCount calls.

### addMessage

```
public int addMessage(String message)
```

Adds the given message as type InfoMsg. If the message begins with the string Error or Warning then it is added without that prefix.

**Parameters:**

message - a String containing the message to be displayed.

### addMessage

```
public int addMessage(String text,  
int severityCode)
```

Adds the given message with the given severity code

(continued from last page)

**Parameters:**

`text` - a String containing the message to be displayed.

**Returns:**

the index of the Message created for use with `replaceMessage`.

---

**addMessage**

```
public void addMessage(String text,  
    AbstractComponent comp)
```

Adds the given message as type `InfoMsg` with the attached component.

**Parameters:**

`text` - a String containing the message to be displayed.

**Returns:**

the index of the Message created for use with `replaceMessage`.

---

**addMessage**

```
public void addMessage(String text,  
    Vector comps,  
    int severityCode)
```

Adds the given message as type `severityCode` with the attached components.

**Parameters:**

`text` - a String containing the message to be displayed.

**Returns:**

the index of the Message created for use with `replaceMessage`.

---

**addMessage**

```
public void addMessage(String text,  
    AbstractComponent comp,  
    int severityCode)
```

Adds the given message as type `severityCode` with the attached component.

**Parameters:**

`text` - a String containing the message to be displayed.

**Returns:**

the index of the Message created for use with `replaceMessage`.

---

**addMessage**

```
public void addMessage(String text,  
    ViewComponent comp,  
    int severityCode,  
    JComponent uiComp)
```

Adds the given message as type `severityCode` in the given View for the given UI Component.

**Parameters:**

`text` - a String containing the message to be displayed.  
`uiComp` - the JComponent target of this message.

**Returns:**

the index of the Message created for use with `replaceMessage`.

---

## addMessage

```
public void addMessage(String text,  
    GenericObject object,  
    int severityCode)
```

Adds the given message as type severityCode with the attached component.

**Parameters:**

text - a String containing the message to be displayed.

**Returns:**

the index of the Message created for use with replaceMessage.

---

## getWarningCount

```
public int getWarningCount()
```

Retrieves the number of warning messages since the last clearWarningCount or clearCheckErrors call.

---

## getNumberOfErrorsFound

```
public int getNumberOfErrorsFound()
```

Retrieves the number of error messages since the last clearWarningCount or clearCheckErrors call.

---

## selectComp

```
public void selectComp(AbstractComponent comp,  
    boolean clear)
```

**Deprecated.**

adds a component to the selected list and centers the ModelView on the given component.

**Parameters:**

comp - the AbstractComponent object to be added to list of selected components and centered on.

clear - if true, clear list of selected components before selecting

---

## toFront

```
public void toFront()
```

---

## addSound

```
public void addSound(int soundType)
```

Attempts to play a sound of the given type. Valid types are: 0 - Chime / AlertSound1 - Boing / InfoSound2 - Cowbell / WarningSound3 - UhOh / UserErrSound4 - Gong / IntErrSound5 - Swish / ConnectSound6 - Cork / DisconnectSound NOTE: Works for JDK 1.3 and above.

**Parameters:**

soundType - the sound to play.

---

## dateStamp

```
public String dateStamp()
```

Creates a formatted date stamp of the current time/date.

---

## com.cafean.client.ui Class NamedValueSelector

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- java.awt.Window
                    |
                    +- java.awt.Dialog
                          |
                          +- javax.swing.JDialog
                                |
                                +- com.cafean.client.ui.NamedValueSelector
  
```

public class **NamedValueSelector**  
extends **JDialog**

The **NamedValueSelector** provides the user with list of values with a name associated with them. The **NamedValueSelector** is given two parallel arrays or **Vectors** with the value and the names in each.

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | NamedValueSelector(java.awt.Frame parent, Object[] values, Object[] names)<br>Creates a new <b>NamedValueSelector</b> that is modal over the given parent. |
| public | NamedValueSelector(JDialog parent, Object[] values, Object[] names)<br>Creates a new <b>NamedValueSelector</b> that is modal over the given parent.        |
| public | NamedValueSelector(java.awt.Frame parent, Vector values, Vector names)<br>Creates a new <b>NamedValueSelector</b> that is modal over the given parent.     |
| public | NamedValueSelector(JDialog parent, Vector values, Vector names)<br>Creates a new <b>NamedValueSelector</b> that is modal over the given parent.            |

## Method Summary

|                 |  |
|-----------------|--|
| int             | getSelectedIndex()<br>Returns the index of the object selected by the user.  |
| TableCellEditor | getTableCellEditor()<br>Provides a table cell editor of a property editor panel and the popup named value selection dialog |

|         |  |
|---------|--|
| boolean | isCancelled()<br><br>This is used to determine how the user closed the NamedValueSelector. |
| void    | setValueLabel(String text)   |

**Methods inherited from class** javax.swing.JDialog

getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setJMenuBar, setLayeredPane, setLayout, update

**Methods inherited from class** java.awt.Dialog

addNotify, getAccessibleContext, getTitle, hide, isModal, isResizable, isUndecorated, setModal, setResizable, setTitle, setUndecorated, show

**Methods inherited from class** java.awt.Window

addNotify, addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getAccessibleContext, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, show, toBack, toFront

**Methods inherited from class** java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

**Methods inherited from class** java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### NamedValueSelector

```

public NamedValueSelector(java.awt.Frame parent,
                           Object[] values,
                           Object[] names)

```

Creates a new NamedValueSelector that is modal over the given parent.

#### Parameters:

- parent - the Frame object that spawned this selector.
- values - the Object[] of values.
- names - the Object[] of names for the values.



(continued from last page)

## NamedValueSelector

```
public NamedValueSelector(JDialog parent,  
                           Object[] values,  
                           Object[] names)
```

Creates a new NamedValueSelector that is modal over the given parent.

### Parameters:

parent - the JDialog object that spawned this selector.  
values - the Object[] of values.  
names - the Object[] of names for the values.

---

## NamedValueSelector

```
public NamedValueSelector(java.awt.Frame parent,  
                           Vector values,  
                           Vector names)
```

Creates a new NamedValueSelector that is modal over the given parent.

### Parameters:

parent - the Frame object that spawned this selector.  
values - the Vector of values.  
names - the Vector of names for the values.

---

## NamedValueSelector

```
public NamedValueSelector(JDialog parent,  
                           Vector values,  
                           Vector names)
```

Creates a new NamedValueSelector that is modal over the given parent.

### Parameters:

parent - the Dialog object that spawned this selector.  
values - the Vector of values.  
names - the Vector of names for the values.

## Methods

### getSelectedIndex

```
public int getSelectedIndex()
```

Returns the index of the object selected by the user.

### Returns:

the index of the Object selected by the user.

---

### isCancelled

```
public boolean isCancelled()
```

This is used to determine how the user closed the NamedValueSelector. If the user exits the dialog by any means other than by pressing the OK button, this is set to TRUE.

### Returns:

FALSE if the user pressed the OK button.

(continued from last page)

## **setValueLabel**

```
public void setValueLabel(String text)
```

---

## **getTableCellEditor**

```
public TableCellEditor getTableCellEditor()
```

Provides a table cell editor of a property editor panel and the popup named value selection dialog

### **Returns:**

the TableCellEditor to be used for this dialog

## com.cafean.client.ui

# Class RealArrayDialog

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- java.awt.Window
                    |-- java.awt.Dialog
                          |-- javax.swing.JDialog
                                |-- com.cafean.client.ui.RealArrayDialog

```

public class **RealArrayDialog**  
 extends JDialog

This dialog allows the user to change the values stored inside an array of Reals. Optionally this allows for the number of elements inside the array to change as well. This option defaults to true.

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | RealArrayDialog(java.awt.Frame parent,Real[] array)<br>Creates new form RealArrayDialog  |
| public | RealArrayDialog(java.awt.Frame parent,Real[] array,boolean fixedDimension)<br>Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.                     |
| public | RealArrayDialog(java.awt.Frame parent,Real[] array,boolean fixedDimension,AbstractModel model)<br>Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified. |
| public | RealArrayDialog(JDialog parent,Real[] array)<br>Creates new form RealArrayDialog   |
| public | RealArrayDialog(JDialog parent,Real[] array,boolean fixedDimension)<br>Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.                            |
| public | RealArrayDialog(JDialog parent,Real[] array,boolean fixedDimension,AbstractModel model)<br>Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.        |

## Method Summary

|                        |  |
|------------------------|--|
| <a href="#">Real[]</a> | <code>getArray()</code><br>Gets the array that has been modified by this dialog. |
| <code>boolean</code>   | <code>isCancelled()</code><br>Determines how the dialog was exited.              |

### Methods inherited from class `javax.swing.JDialog`

`getAccessibleContext`, `getContentPane`, `getDefaultCloseOperation`, `getGlassPane`, `getJMenuBar`, `getLayeredPane`, `getRootPane`, `isDefaultLookAndFeelDecorated`, `remove`, `setContentPane`, `setDefaultCloseOperation`, `setDefaultLookAndFeelDecorated`, `setGlassPane`, `setJMenuBar`, `setLayeredPane`, `setLayout`, `update`

### Methods inherited from class `java.awt.Dialog`

`addNotify`, `getAccessibleContext`, `getTitle`, `hide`, `isModal`, `isResizable`, `isUndecorated`, `setModal`, `setResizable`, `setTitle`, `setUndecorated`, `show`

### Methods inherited from class `java.awt.Window`

`addNotify`, `addPropertyChangeListener`, `addPropertyChangeListener`, `addWindowFocusListener`, `addWindowListener`, `addWindowStateListener`, `applyResourceBundle`, `applyResourceBundle`, `createBufferStrategy`, `createBufferStrategy`, `dispose`, `getAccessibleContext`, `getBufferStrategy`, `getFocusableWindowState`, `getFocusCycleRootAncestor`, `getFocusOwner`, `getFocusTraversalKeys`, `getGraphicsConfiguration`, `getInputContext`, `getListeners`, `getLocale`, `getMostRecentFocusOwner`, `getOwnedWindows`, `getOwner`, `getToolkit`, `getWarningString`, `getWindowFocusListeners`, `getWindowListeners`, `getWindowStateListeners`, `hide`, `isActive`, `isAlwaysOnTop`, `isFocusableWindow`, `isFocusCycleRoot`, `isFocused`, `isLocationByPlatform`, `isShowing`, `pack`, `postEvent`, `removeWindowFocusListener`, `removeWindowListener`, `removeWindowStateListener`, `setAlwaysOnTop`, `setBounds`, `setCursor`, `setFocusableWindowState`, `setFocusCycleRoot`, `setLocationByPlatform`, `setLocationRelativeTo`, `show`, `toBack`, `ToFront`

### Methods inherited from class `java.awt.Container`

`add`, `add`, `add`, `add`, `add`, `addContainerListener`, `addNotify`, `addPropertyChangeListener`, `addPropertyChangeListener`, `applyComponentOrientation`, `areFocusTraversalKeysSet`, `countComponents`, `deliverEvent`, `doLayout`, `findComponentAt`, `findComponentAt`, `getAlignmentX`, `getAlignmentY`, `getComponent`, `getComponentAt`, `getComponentAt`, `getComponentCount`, `getComponents`, `getComponentZOrder`, `getContainerListeners`, `getFocusTraversalKeys`, `getFocusTraversalPolicy`, `getInsets`, `getLayout`, `getListeners`, `getMaximumSize`, `getMinimumSize`, `getMousePosition`, `getPreferredSize`, `insets`, `invalidate`, `isAncestorOf`, `isFocusCycleRoot`, `isFocusCycleRoot`, `isFocusTraversalPolicyProvider`, `isFocusTraversalPolicySet`, `layout`, `list`, `list`, `locate`, `minimumSize`, `paint`, `paintComponents`, `preferredSize`, `print`, `printComponents`, `remove`, `remove`, `removeAll`, `removeContainerListener`, `removeNotify`, `setComponentZOrder`, `setFocusCycleRoot`, `setFocusTraversalKeys`, `setFocusTraversalPolicy`, `setFocusTraversalPolicyProvider`, `setFont`, `setLayout`, `transferFocusBackward`, `transferFocusDownCycle`, `update`, `validate`

### Methods inherited from class `java.awt.Component`

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## RealArrayDialog

### Creates new form RealArrayDialog

## RealArrayDialog

Creates a new `RealArrayDialog` for editing the given array, setting whether the array's length can be modified.

(continued from last page)

**Parameters:**

parent - the Frame that created this editor.  
array - the Real[] that is being modified.  
fixedDimension - whether the user can modify the number of elements.

---

**RealArrayDialog**

```
public RealArrayDialog(java.awt.Frame parent,  
                        Real[] array,  
                        boolean fixedDimension,  
                        AbstractModel model)
```

Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.

**Parameters:**

parent - the Frame that created this editor.  
array - the Real[] that is being modified.  
fixedDimension - whether the user can modify the number of elements.

---

**RealArrayDialog**

```
public RealArrayDialog(JDialog parent,  
                        Real[] array)
```

Creates new form RealArrayDialog

---

**RealArrayDialog**

```
public RealArrayDialog(JDialog parent,  
                        Real[] array,  
                        boolean fixedDimension)
```

Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.

**Parameters:**

parent - the JDialog that created this editor.  
array - the Real[] that is being modified.  
fixedDimension - whether the user can modify the number of elements.

---

**RealArrayDialog**

```
public RealArrayDialog(JDialog parent,  
                        Real[] array,  
                        boolean fixedDimension,  
                        AbstractModel model)
```

Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.

**Parameters:**

parent - the JDialog that created this editor.  
array - the Real[] that is being modified.  
fixedDimension - whether the user can modify the number of elements.

---

## Methods

**isCancelled**

```
public boolean isCancelled()
```

Determines how the dialog was exited.

**Returns:**

(continued from last page)

true unless the Ok button was pressed.

---

## **getArray**

```
public Real\[\] getArray()
```

Gets the array that has been modified by this dialog.

## com.cafean.client.ui

### Class RealEditor

```

java.lang.Object
  |
  +- javax.swing.AbstractCellEditor
      |
      +- javax.swing.DefaultCellEditor
          |
          +- com.cafean.client.ui.RealEditor
  
```

#### All Implemented Interfaces:

[ModelDependent](#), java.io.Serializable, CellEditor, TreeCellEditor, TableCellEditor

#### public class **RealEditor**

extends DefaultCellEditor

implements TableCellEditor, TreeCellEditor, CellEditor, java.io.Serializable, [ModelDependent](#)

RealEditor is a TableCellEditor that edits Real values stored inside a JTable. RealEditor is actually a wrapper for a RealTextField that actually handles editing the Real value. RealEditor overrides getCellEditorValue and returns the value from the RealTextField.

#### See Also:

Real, RealTextField

### Constructor Summary

|        |  |
|--------|--|
| public | RealEditor()<br>This constructor for a new RealEditor will create its own RealTextField. |
| public | RealEditor(RealTextField field)<br>This constructor for a new RealEditor                 |

### Method Summary

|                               |  |
|-------------------------------|--|
| boolean                       | doMultiEdit()<br>  |
| Object                        | getCellEditorValue()<br>This gets the value from the RealTextField.  |
| <a href="#">AbstractModel</a> | getModel()<br>   |
| java.awt.Component            | getTableCellEditorComponent(JTable table, Object value, boolean isSelected, int row, int column)<br>delegates to DefaultCellEditor and sets the returned component's font to the table's font. |
| void                          | setModel(AbstractModel model)<br>  |



|             |   |
|-------------|---|
| static void | <pre>setUpEditor(JTable table,int fromCol,int toCol,String[] line1,String[] line2,MultiLineHeaderRenderer headerRenderer)</pre> <p>This sets up a table with RealEditors between the specified columns.</p> |
| static void | <pre>setUpRealEditor(JTable table)</pre> <p>This fills all the columns in a JTable with RealEditors.</p>  |
| static void | <pre>setUpRealEditor(JTable table,AbstractModel model)</pre> <p>This fills all the columns in a JTable with RealEditors.</p>  |

#### Methods inherited from class javax.swing.DefaultCellEditor

cancelCellEditing, getCellEditorValue, getClickCountToStart, getComponent, getTableCellEditorComponent, getTreeCellEditorComponent, isCellEditable, setClickCountToStart, shouldSelectCell, stopCellEditing

#### Methods inherited from class javax.swing.AbstractCellEditor

addCellEditorListener, cancelCellEditing, getCellEditorListeners, isCellEditable, removeCellEditorListener, shouldSelectCell, stopCellEditing

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### RealEditor

```
public RealEditor()
```

This constructor for a new RealEditor will create its own RealTextField. This is for use inside Tables where the RealTextField doesn't need a parent.

### RealEditor

```
public RealEditor(RealTextField field)
```

This constructor for a new RealEditor

#### Parameters:

textField - the RealTextField that will actually edit the value.

## Methods

### getCellEditorValue

```
public Object getCellEditorValue()
```

This gets the value from the RealTextField. If the text returned is not a valid floating point number, the Real returned is Unknown.

#### Returns:

the Real value form the RealTextField, or a Real set unknown.

## setUpRealEditor

```
public static void setUpRealEditor(JTable table)
```

This fills all the columns in a JTable with RealEditors.

### Parameters:

table - the JTable that is being filled.

---

## setUpRealEditor

```
public static void setUpRealEditor(JTable table,  
    AbstractModel model)
```

This fills all the columns in a JTable with RealEditors.

### Parameters:

table - the JTable that is being filled.

model - the AbstractModel that contains the current units.

---

## setUpEditor

```
public static void setUpEditor(JTable table,  
    int fromCol,  
    int toCol,  
    String[] line1,  
    String[] line2,  
    MultiLineHeaderRenderer headerRenderer)
```

This sets up a table with RealEditors between the specified columns.

### Parameters:

table - the JTable that is being setup.

fromCol - the first column that should have RealEditors.

toCol - the last column that should have RealEditors.

line1 - the String[] that contains the first line for all of the column headers.

line2 - the String[] that contains the second line for all of the column headers.

headerRenderer - a MultiLineHeaderRenderer for the table.

---

## doMultiEdit

```
public boolean doMultiEdit()
```

---

## getTableCellEditorComponent

```
public java.awt.Component getTableCellEditorComponent(JTable table,  
    Object value,  
    boolean isSelected,  
    int row,  
    int column)
```

delegates to DefaultCellEditor and sets the returned component's font to the table's font.

---

## getModel

```
public AbstractModel getModel()
```

---

(continued from last page)

## **setModel**

```
public void setModel(AbstractModel model)
```

## com.cafean.client.ui Class RealTextField

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.text.JTextComponent
          |-- javax.swing.JTextField
            |-- com.cafean.client.ui.RealTextField

```

### All Implemented Interfaces:

[ModelDependent](#), java.awt.datatransfer.ClipboardOwner, java.io.Serializable, InsertErrorListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, Scrollable, SwingConstants

### public class **RealTextField**

extends JTextField

implements SwingConstants, Scrollable, javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, InsertErrorListener, java.io.Serializable, java.awt.datatransfer.ClipboardOwner, [ModelDependent](#)

The RealTextField is a JTextField that is specialized for working with Real values and extensions from the Number package. A RealTextField is created like a regular TextField. However, getValue() returns a Real Object instead of a String Object. The default Real is dimensionless, so storing the entered value into a value with units should involve a convert( double value )call on the target to preserve the entered value.

#### Fields inherited from class javax.swing.JTextField

notifyAction

#### Fields inherited from class javax.swing.text.JTextComponent

DEFAULT\_KEYMAP, FOCUS\_ACCELERATOR\_KEY

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |   |
|--------|---|
| public | <code>RealTextField()</code><br>Creates a default textfield with a null string, 0 columns, and a null <code>DecimalFormat</code> .  |
| public | <code>RealTextField(int alignment)</code><br>Creates a default textfield with a null string, 0 columns, and a null <code>DecimalFormat</code> .   |
| public | <code>RealTextField(String text,int columns,java.text.DecimalFormat format)</code>  |
| public | <code>RealTextField(String text,int columns,java.text.DecimalFormat format,boolean addListener)</code><br>This constructor allows for the <code>MouseListener</code> that opens the right click menu for the text field to be disabled. |
| public | <code>RealTextField(int columns,java.text.DecimalFormat format)</code><br>Creates a textfield with an initial null <code>String</code> as text, and the specified number of columns and format.   |
| public | <code>RealTextField(String text)</code><br>Creates a textfield with the specified text, 0 columns and a null <code>DecimalFormat</code>   |
| public | <code>RealTextField(String text,int columns)</code><br>Creates a textfield with the specified text, the specified number of columns, and a null <code>DecimalFormat</code>  |

## Method Summary

|                                      |  |
|--------------------------------------|--|
| void                                 | <code>formatChanged()</code><br>This function is called when the document changes format   |
| Double                               | <code>getDoubleValue()</code><br>This attempts to return the value inside this <code>RealTextField</code> as a <code>Double</code> .                                     |
| <code>java.text.DecimalFormat</code> | <code>getFormat()</code><br>Getter for the <code>DecimalFormat</code> of this <code>RealTextField</code> .   |
| Long                                 | <code>getLongValue()</code><br>This attempts to return the value inside this <code>RealTextField</code> as a <code>Long</code> .   |
| <a href="#">AbstractModel</a>        | <code>getModel()</code>  |
| Number                               | <code>getNumberValue()</code><br>This attempts to return the value inside this <code>RealTextField</code> as a <code>Number</code> .                                     |
| <a href="#">Real</a>                 | <code>getValue()</code><br>This attempts to return the value inside this <code>RealTextField</code> as a <code>Real</code> .   |
| void                                 | <code>insertFailed(NumericPlainDocument doc,int offset,String str,AttributeSet a)</code><br>This function is overridden to handle an error that occurs during insertion. |
| boolean                              | <code>isUncertain()</code><br>Determines if the current value in the <code>RealTextField</code> is uncertain.  |

|         |   |
|---------|---|
| boolean | isUnknown()<br>This method is used to determine whether the value inside this RealTextField is a known value.                   |
| void    | lostOwnership(java.awt.datatransfer.Clipboard clipboard, java.awt.datatransfer.Transferable contents)                           |
| void    | normalize()<br>Formats the String in the RealTextField to the specified DecimalFormat.  |
| void    | setFormat(java.text.DecimalFormat format)<br>Sets the DecimalFormat on this RealTextField.                                      |
| void    | setModel(AbstractModel model)   |
| void    | setText(String text)  |
| void    | setValue(double d)<br>Sets the value of the current RealTextField to the specified double                                       |
| void    | setValue(long l)<br>Sets the value of the current RealTextField to the specified long   |
| void    | setValue(Number number)<br>Sets the value of the current RealTextField to the specified Number                                  |
| void    | setValue(Real r)<br>Sets the value of the current RealTextField to the specified Real value.                                    |
| void    | setValueUncertain()<br>If the RealTextField represents multiple sources of data that differ, the value is considered uncertain. |
| void    | setValueUnknown()<br>Sets the value of the current RealTextField to Unknown   |
| String  | toString()<br>Returns the current value of the RealTextField as a String.   |

**Methods inherited from class** javax.swing.JTextField

addActionListener, getAccessibleContext, getAction, getActionListeners, getActions, getColumns, getHorizontalAlignment, getHorizontalVisibility, getPreferredSize, getScrollOffset, getUIClassID, isValidateRoot, postActionEvent, removeActionListener, scrollRectToVisible, setAction, setActionCommand, setColumns, setDocument, setFont, setHorizontalAlignment, setScrollOffset

**Methods inherited from class** javax.swing.text.JTextComponent

addCaretListener, addInputMethodListener, addKeymap, copy, cut, getAccessibleContext, getActions, getCaret, getCaretColor, getCaretListeners, getCaretPosition, getDisabledTextColor, getDocument, getDragEnabled, getFocusAccelerator, getHighlighter, getInputMethodRequests, getKeymap, getKeymap, getMargin, getNavigationFilter, getPreferredScrollableViewportSize, getScrollableBlockIncrement, getScrollableTracksViewportHeight, getScrollableTracksViewportWidth, getScrollableUnitIncrement, getSelectedText, getSelectedTextColor, getSelectionColor, getSelectionEnd, getSelectionStart, getText, getText, getToolTipText, getUI, isEditable, loadKeymap, modelToView, moveCaretPosition, paste, read, removeCaretListener, removeKeymap, removeNotify, replaceSelection, select, selectAll, setCaret, setCaretColor, setCaretPosition, setComponentOrientation, setDisabledTextColor, setDocument, setDragEnabled, setEditable, setFocusAccelerator, setHighlighter, setKeymap, setMargin, setNavigationFilter, setSelectedTextColor, setSelectionColor, setSelectionEnd, setSelectionStart, setText, setUI, updateUI, viewToModel, write

#### Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component





(continued from last page)

## RealTextField

```
public RealTextField(String text,  
                     int columns,  
                     java.text.DecimalFormat format)
```

### Parameters:

text - the initial text to appear in the textfield  
columns - the number of columns to appear in the textfield  
format - the DecimalFormat to display the numbers in the textfield

---

## RealTextField

```
public RealTextField(String text,  
                     int columns,  
                     java.text.DecimalFormat format,  
                     boolean addListener)
```

This constructor allows for the MouseListener that opens the right click menu for the text field to be disabled.

### Parameters:

text - the initial text to appear in the textfield  
columns - the number of columns to appear in the textfield  
format - the DecimalFormat to display the numbers in the textfield  
addListener - boolean flag for disabling the MouseListener

---

## RealTextField

```
public RealTextField(int columns,  
                     java.text.DecimalFormat format)
```

Creates a textfield with an initial null String as text, and the specified number of columns and format.

---

## RealTextField

```
public RealTextField(String text)
```

Creates a textfield with the specified text, 0 columns and a null DecimalFormat

---

## RealTextField

```
public RealTextField(String text,  
                     int columns)
```

Creates a textfield with the specified text, the specified number of columns, and a null DecimalFormat

## Methods

### setFormat

```
public void setFormat(java.text.DecimalFormat format)
```

Sets the DecimalFormat on this RealTextField.

### Parameters:

format - the DecimalFormat required for the RealTextField.

---

### getFormat

```
public java.text.DecimalFormat getFormat()
```

(continued from last page)

Getter for the DecimalFormat of this RealTextField.

**Returns:**

the DecimalFormat.

---

## formatChanged

```
public void formatChanged()
```

This function is called when the document changes format

---

## isUnknown

```
public boolean isUnknown()
```

This method is used to determine whether the value inside this RealTextField is a known value.

**Returns:**

TRUE if the value in the RealTextField is unknown.

---

## setText

```
public void setText(String text)
```

---

## getLongValue

```
public Long getLongValue()  
throws java.text.ParseException
```

This attempts to return the value inside this RealTextField as a Long. If the value cannot be converted it throws a ParseException.

**Returns:**

the value of the RealTextField as a Long

**Throws:**

a - ParseException if the value in the textfield cannot be parsed as a Long

---

## getDoubleValue

```
public Double getDoubleValue()  
throws java.text.ParseException
```

This attempts to return the value inside this RealTextField as a Double. If the value cannot be converted it throws a ParseException.

**Returns:**

the value of the RealTextField as a Double

**Throws:**

a - ParseException if the value in the textfield cannot be parsed as a Double

---

## getNumberValue

```
public Number getNumberValue()  
throws java.text.ParseException
```

This attempts to return the value inside this RealTextField as a Number. If the value cannot be converted it throws a ParseException.

**Returns:**

(continued from last page)

the value of the textfield as a Number

**Throws:**

- a - ParseException if the value in the textfield cannot be parsed as a Number

---

## getValue

```
public Real getValue()  
    throws java.text.ParseException
```

This attempts to return the value inside this RealTextField as a Real. If the value cannot be converted it throws a ParseException.

**Returns:**

the value of the textfield as a Real

**Throws:**

- a - ParseException if the value in the textfield cannot be parsed as a Real

---

## setValueUnknown

```
public void setValueUnknown()  
    Sets the value of the current RealTextField to Unknown
```

---

## setValue

```
public void setValue(Number number)  
    Sets the value of the current RealTextField to the specified Number
```

**Parameters:**

- number - the Number to be set in the RealTextField.

---

## setValue

```
public void setValue(long l)  
    Sets the value of the current RealTextField to the specified long
```

**Parameters:**

- l - the long to be set in the RealTextField.

---

## setValue

```
public void setValue(double d)  
    Sets the value of the current RealTextField to the specified double
```

**Parameters:**

- d - the double to be set in the RealTextField.

---

## setValue

```
public void setValue(Real r)  
    Sets the value of the current RealTextField to the specified Real value.
```

**Parameters:**

- r - the Real to be set in the RealTextField.

(continued from last page)

---

## normalize

```
public void normalize()  
    throws java.text.ParseException
```

Formats the String in the RealTextField to the specified DecimalFormat.

**Throws:**

ParseException - if the current String cannot be formatted

---

## insertFailed

```
public void insertFailed(NumericPlainDocument doc,  
    int offset,  
    String str,  
    AttributeSet a)
```

This function is overridden to hanadle an error that occurs during insertion.

**Parameters:**

doc - the NumericPlainDocument being edited.

offset - the current index in the string where the error occured.

str - the String that is in the TextField.

---

## toString

```
public String toString()
```

Returns the current value of the RealTextField as a String.

**Returns:**

the text String of the RealTextField

---

## setValueUncertain

```
public void setValueUncertain()
```

If the RealTextField represents multiple sources of data that differ, the value is considered uncertain. the Text that is displayed is multiple stars. This sets the value stored to be uncertain.

---

## isUncertain

```
public boolean isUncertain()
```

Determines if the current value in the RealTextField is uncertain.

**Returns:**

TRUE if the value is uncertain.

---

## lostOwnership

```
public void lostOwnership(java.awt.datatransfer.Clipboard clipboard,  
    java.awt.datatransfer.Transferable contents)
```

---

## getModel

```
public AbstractModel getModel()
```

---

(continued from last page)

## **setModel**

```
public void setModel(AbstractModel model)
```

## com.cafean.client.ui Interface RefreshableDialog

### All Known Implementing Classes:

AsciiViewer

### public interface **RefreshableDialog**

An interface describing a Dialog extension whos data and/or structure can be refreshed directly based on a change of the currently displayed units type or a global refresh.

### Method Summary

|      |  |
|------|--|
| void | <code>refresh()</code><br>Refreshes and redisplay this dialog's data   |
| void | <code>unitsChanged()</code><br>Indicates that the currently displayed units type has changed and updates this dialogs display. |

### Methods

#### **unitsChanged**

public void **unitsChanged()**

Indicates that the currently displayed units type has changed and updates this dialogs display.

#### **refresh**

public void **refresh()**

Refreshes and redisplay this dialog's data

## com.cafean.client.ui Class TableSorter

```

java.lang.Object
  |
  +- javax.swing.table.AbstractTableModel
    |
    +- com.cafean.client.ui.TableSorter
  
```

### All Implemented Interfaces:

TableModelListener, java.io.Serializable, TableModel

```

public class TableSorter
extends AbstractTableModel
implements TableModel, java.io.Serializable, TableModelListener
  
```

A sorting wrapper for TableModels to allow sorting by a particular column.

TableSorter does not store or copy the data in the TableModel, instead it maintains an array of integers which it keeps the same size as the number of rows in its model.

When the model changes it notifies the sorter that something has changed eg. "rowsAdded" so that its internal array of integers can be reallocated. As requests are made of the sorter (such as `int`) it redirects them to its model via the mapping array. That way the TableSorter appears to hold another copy of the table with the rows in a different order. The sorting algorithm used is stable which means that it does not move around rows when its comparison function returns 0 to denote that they are equivalent.

To use, simply wrap your table model with TableSorter and use `#getIndexForRow` to convert JTable row indicies to your TableModel's row indicies.

Modified from an example by Philip Milne from [java.sun.com swing tutorial](http://java.sun.com/swing/tutorial)

### See Also:

`#addMouseListenerToHeaderInTable`

## Constructor Summary

|        |  |
|--------|--|
| public | TableSorter()<br>Creates a new instance of Table Sorter                                  |
| public | TableSorter(TableModel model)<br>Creates an instance of table sorter given a table model |

## Method Summary

|      |  |
|------|--|
| void | addMouseListenerToHeaderInTable(JTable table)<br>Adds a mouse listener to the table header.          |
| void | checkModel()<br>Verifies the current model has the same size as the data stored by the table sorter. |
| int  | compare(int row1,int row2)<br>Compares the values stored in two rows of the table.                   |

|            |   |
|------------|---|
| int        | <code>compareRowsByColumn(int row1,int row2,int column)</code><br>Compares the value stored in two rows of a given column.  |
| Class      | <code>getColumnClass(int aColumn)</code><br>Returns the Class of the given column.  |
| int        | <code>getColumnCount()</code><br>Returns the number of columns in the tablemodel.   |
| String     | <code>getColumnName(int aColumn)</code><br>Returns the name of the given column.  |
| int        | <code>getIndexForRow(int aRow)</code><br>Finds the actual index given a specific table row.   |
| TableModel | <code>getModel()</code><br>Accessor for the TableModel of the JTable this is sorting  |
| int        | <code>getRowCount()</code><br>Returns the number of rows in the tablemodel.   |
| Object     | <code>getValueAt(int aRow,int aColumn)</code><br>Passes through the value stored in the sorted locations.   |
| boolean    | <code>isCellEditable(int row,int column)</code><br>Determines if the cell specified by row and column is editable.  |
| void       | <code>setModel(TableModel model)</code><br>Setter for the TableModel of the JTable this is sorting  |
| void       | <code>setValueAt(Object aValue,int aRow,int aColumn)</code><br>Sets the value stored in the sorted locations.   |
| void       | <code>sort(Object sender)</code><br>Arranges the rows of a table so that they are sorted.   |
| void       | <code>sortByColumn(int column)</code><br>Sorts the table by the given column in ascending order.  |
| void       | <code>sortByColumn(int column,boolean ascending)</code><br>Sorts the table by the given column, in either ascending or decending order.   |
| void       | <code>tableChanged(TableModelEvent e)</code><br>This is implemented to support the TableListener interface the table indecies are reallocated whenever a table's values change. |

**Methods inherited from class `javax.swing.table.AbstractTableModel`**

`addTableModelListener`, `findColumn`, `fireTableCellUpdated`, `fireTableChanged`, `fireTableDataChanged`, `fireTableRowsDeleted`, `fireTableRowsInserted`, `fireTableRowsUpdated`, `fireTableStructureChanged`, `getColumnClass`, `getColumnName`, `getListeners`, `getTableModelListeners`, `isCellEditable`, `removeTableModelListener`, `setValueAt`

**Methods inherited from class `java.lang.Object`**



```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Constructors

### TableSorter

```
public TableSorter()
```

Creates a new instance of Table Sorter

### TableSorter

```
public TableSorter(TableModel model)
```

Creates an instance of table sorter given a table model

**Parameters:**

model - the TableModel for the JTable this is sorting.

## Methods

### getModel

```
public TableModel getModel()
```

Accessor for the TableModel of the JTable this is sorting

### setModel

```
public void setModel(TableModel model)
```

Setter for the TableModel of the JTable this is sorting

**Returns:**

the TableModel.

### compareRowsByColumn

```
public int compareRowsByColumn(int row1,  
                                int row2,  
                                int column)
```

Compares the value stored in two rows of a given column.

**Parameters:**

row1 - the index of the first row.

row2 - the index of the second row.

column - the index of the column.

### compare

```
public int compare(int row1,  
                   int row2)
```

Compares the values stored in two rows of the table.

**Parameters:**

row1 - the index of the first row.

(continued from last page)

row2 - the index of the second row.

---

## tableChanged

```
public void tableChanged(TableModelEvent e)
```

This is implemented to support the TableListener interface the table indecies are reallocated whenever a table's values change.

### Parameters:

e - the TableModeEvent that was fired.

---

## checkModel

```
public void checkModel()
```

Verifies the current model has the same size as the data stored by the table sorter.

### Throws:

Exception - if the model has a different number of rows then the local vector.

---

## sort

```
public void sort(Object sender)
```

Arranges the rows of a table so that they are sorted.

---

## getValueAt

```
public Object getValueAt(int aRow,  
                           int aColumn)
```

Passes through the value stored in the sorted locations.

### Parameters:

aRow - the row index.

aColumn - the column index.

### Returns:

the Object stored at the sorted row index for aRow

---

## setValueAt

```
public void setValueAt(Object aValue,  
                        int aRow,  
                        int aColumn)
```

Sets the value stored in the sorted locations.

### Parameters:

aValue - the Object stored at the sorted row index for aRow.

aRow - the row index.

aColumn - the column index.

---

## getRowCount

```
public int getRowCount()
```

Returns the number of rows in the tablemodel.

### Returns:

the number of rows in the TableModel or 0 if there isn't a model.

---

## getColumnCount

```
public int getColumnCount()
```

Returns the number of columns in the tablemodel.

**Returns:**

the number of columns in the TableModel or 0 if there isn't a model.

---

## getColumnName

```
public String getColumnName(int aColumn)
```

Returns the name of the given column.

**Parameters:**

aColumn - the index of the column.

**Returns:**

the String name stored for the column in the table model.

---

## getColumnClass

```
public Class getColumnClass(int aColumn)
```

Returns the Class of the given column.

**Parameters:**

aColumn - the index of the column.

**Returns:**

the Class stored for the column in the table model.

---

## isCellEditable

```
public boolean isCellEditable(int row,  
int column)
```

Determines if the cell specified by row and column is editable. This is passed on to the TableModel.

**Parameters:**

row - the row index.

column - the column index.

**Returns:**

TRUE if the specified cell can be edited.

---

## sortByColumn

```
public void sortByColumn(int column)
```

Sorts the table by the given column in ascending order.

**Parameters:**

column - the column index.

---

## sortByColumn

```
public void sortByColumn(int column,  
boolean ascending)
```

(continued from last page)

Sorts the table by the given column, in either ascending or decending order.

**Parameters:**

column - the column index.

ascending - If this is TRUE, the table is sorted in ascending order.

---

## **getIndexForRow**

```
public int getIndexForRow(int aRow)
```

Finds the actual index given a specific table row.

**Parameters:**

aRow - the row index.

---

## **addMouseListenerToHeaderInTable**

```
public void addMouseListenerToHeaderInTable(JTable table)
```

Adds a mouse litener to the table header. If a column is selected, that column gets sorted.

**Parameters:**

table - the JTable.

## com.cafean.client.ui Class Toolbox

```

java.lang.Object
  |
+- java.awt.Component
    |
    +- java.awt.Container
        |
        +- javax.swing.JComponent
            |
            +- javax.swing.JToolBar
                |
                +- com.cafean.client.ui.Toolbox
  
```

public class **Toolbox**  
extends JToolBar

The central handler class for a DrawnView's Tools. Tools in this care are the `MouseHandler` instances that are available in a particular `DrawnView`. Examples of these handlers are the `Select Tool`, `Connection Tool`, etc.

### Field Summary

|                                 |   |
|---------------------------------|---|
| static final<br>java.awt.Cursor | CURSOR_MANIP_P<br>used to indicate the manipulation of points                   |
| static final Icon               | ICON_ELL_ANNOT  |
| static final Icon               | ICON_IMAGE_ANNOT  |
| static final Icon               | ICON_LINE_ANNOT   |
| static final Icon               | ICON_NEW  |
| static final Icon               | ICON_REC_ANNOT  |
| static final Icon               | ICON_TXT_ANNOT  |
| static final int                | TOOL_CONNECT<br>The enumeration for the Connect tool<br>Value: <b>5</b>         |
| static final int                | TOOL_INSERT<br>The enumeration for the Insert tool<br>Value: <b>4</b>           |
| static final int                | TOOL_INTERACTIVE<br>The enumeration for the Interactive tool<br>Value: <b>6</b> |

|                               |  |
|-------------------------------|--|
| <code>static final int</code> | <b>TOOL_NONE</b><br>No tool has been selected<br>Value: <b>0</b>             |
| <code>static final int</code> | <b>TOOL_PAN</b><br>The enumeration for the Pan tool<br>Value: <b>2</b>       |
| <code>static final int</code> | <b>TOOL_SELECT</b><br>The enumeration for the select Tool<br>Value: <b>1</b> |
| <code>static final int</code> | <b>TOOL_ZOOM</b><br>The enumeration for the Zoom tool<br>Value: <b>3</b>     |

**Fields inherited from class** `javax.swing.JComponent`

`TOOL_TIP_TEXT_KEY`, `UNDEFINED_CONDITION`, `WHEN_ANCESTOR_OF_FOCUSED_COMPONENT`, `WHEN_FOCUSED`, `WHEN_IN_FOCUSED_WINDOW`

**Fields inherited from class** `java.awt.Component`

`BOTTOM_ALIGNMENT`, `CENTER_ALIGNMENT`, `LEFT_ALIGNMENT`, `RIGHT_ALIGNMENT`, `TOP_ALIGNMENT`

## Constructor Summary

|                     |   |
|---------------------|---|
| <code>public</code> | <b>Toolbox</b> ( <code>DrawnView parent</code> )<br>Creates new Toolbox |
|---------------------|---|

## Method Summary

|   |  |
|---|--|
| <code>void</code>                       | <b>addToolChangeListener</b> ( <code>ToolChangeListener listener</code> )<br>Adds the given listener to the the list to notify when the current tool has changed.  |
| <code>void</code>                       | <b>createBeanSelectionMenu</b> ( <code>JMenu menu</code> )<br>Creates a select menu appropriate for selecting <code>AbstractDisplayBean</code> instances within the <code>DrawnView</code> parent of this <code>Toolbox</code> .   |
| <code>JToolBar[]</code>                 | <b>createCategoryToolBars</b> ()<br>Creates a set of <code>JToolBar</code> instances appropriate for allowing quick selection of types for the Insertion Tool.   |
| <code>static<br/>java.awt.Cursor</code> | <b>createCursor</b> ( <code>String resource</code> )<br>Creates a <code>Cursor</code> from the image referred to by the given resource name.   |
| <code>JComponent[]</code>               | <b>createDisplayBeans</b> ( <code>int pixelsPerMeter</code> , <code>double widthScaleFactor</code> , <code>DrawnComponent dc</code> )<br>Attempts to create a set of <code>AbstractDisplayBeans</code> from the given <code>DrawnComponent</code> using <code>DrawnComponent#createDisplayBeans</code> . |

|   |  |
|---|--|
| JComponent[]                                      | createDisplayBeans(int pixelsPerMeter, DrawnComponent dc)<br>Attempts to create a set of AbstractDisplayBeans from the given DrawnComponent using DrawnComponent#createDisplayBeans. |
| JComponent  | createSelected()<br>Creates an instance of the currently selected component type in the given model and returns a renderer for the created component.                                |
| static Class                                      | findDisplayBeanClass(String className)<br>Retrieves the Class with the given class name in the list of Display Bean classes available.   |
| <a href="#">Category</a>                          | getComponentCategory()<br>Retrieves the Category of the currently selected component type in the palette of the Toolbox.   |
| java.awt.Cursor                                   | getCurrentCursor()<br>Retrieves the cursor for the views based off of the currently selected tool.   |
| java.awt.Cursor                                   | getCurrentCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for the views based off of the currently selected tool.  |
| int   | getCurrentTool()<br>Retrieves the enumerated type of the current tool.   |
| boolean   | isCompActionSelected()<br>Returns true if the selected toolbox action will create an AbstractComponent.  |
| static com.cafean.client.anim.AbstractDisplayBean | loadDisplayBean(DrawnDisplayBeanRec rec)<br>Loads a display bean from the given display bean rec.  |
| static void                                       | loadDisplayBeanClasses()<br>Loads the display beans from the jar files included in the Components directories in both the current user's home directory and system wide.             |
| boolean   | loadVedaExport(java.io.File exportedFile)<br>Attempts to import a VEDA mask dump file.   |
| void  | removeToolChangeListener(ToolChangeListener listener)<br>Adds the given listener to the the list to notify when the current tool has changed.  |
| void  | setCurrentAction(ToolboxAction action)<br>Sets the current ToolboxAction used for creating new drawings with the Insertion Tool.   |
| void  | setCurrentTool(int tool)<br>switches the toolbox to the given tool.  |
| void  | setLocked(boolean locked)<br>Updates the state of this Toolbox to correspond with the given locked state.  |
| static byte[]                                     | storeDisplayBean(com.cafean.client.anim.AbstractDisplayBean bean)<br>XML Encodes the given bean and returns the byte[] result of the encoding.                                       |

|                |   |
|----------------|---|
| static boolean | verifyBeansAvailable(Vector records)<br><br>Verifies that any AbstractDisplayBeans stored in the given Vector as DrawnDisplayBeanRecs have the proper classes available to be loaded. |
|----------------|---|

#### Methods inherited from class javax.swing.JToolBar

add, addSeparator, addSeparator, getAccessibleContext, getComponentAtIndex, getComponentIndex, getMargin, getOrientation, getUI, getUIClassID, isBorderPainted, isFloatable, isRollover, setBorderPainted, setFloatable, setLayout, setMargin, setOrientation, setRollover, setUI, updateUI

#### Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupMenuLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component



```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate
```

#### Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Fields

### TOOL\_NONE

```
public static final int TOOL_NONE
    No tool has been selected
```

### TOOL\_SELECT

```
public static final int TOOL_SELECT
    The enumeration for the select Tool
```

### TOOL\_PAN

```
public static final int TOOL_PAN
```

(continued from last page)

The enumeration for the Pan tool

---

## **TOOL\_ZOOM**

```
public static final int TOOL_ZOOM
```

The enumeration for the Zoom tool

---

## **TOOL\_INSERT**

```
public static final int TOOL_INSERT
```

The enumeration for the Insert tool

---

## **TOOL\_CONNECT**

```
public static final int TOOL_CONNECT
```

The enumeration for the Connect tool

---

## **TOOL\_INTERACTIVE**

```
public static final int TOOL_INTERACTIVE
```

The enumeration for the Interactive tool

---

## **CURSOR\_MANIP\_P**

```
public static final java.awt.Cursor CURSOR_MANIP_P
```

used to indicate the manipulation of points

---

## **ICON\_NEW**

```
public static final javax.swing.Icon ICON_NEW
```

---

## **ICON\_TXT\_ANNOT**

```
public static final javax.swing.Icon ICON_TXT_ANNOT
```

---

## **ICON\_REC\_ANNOT**

```
public static final javax.swing.Icon ICON_REC_ANNOT
```

---

## **ICON\_ELL\_ANNOT**

```
public static final javax.swing.Icon ICON_ELL_ANNOT
```

---

## **ICON\_LINE\_ANNOT**

```
public static final javax.swing.Icon ICON_LINE_ANNOT
```

---

(continued from last page)

## ICON\_IMAGE\_ANNOT

```
public static final javax.swing.Icon ICON_IMAGE_ANNOT
```

## Constructors

### Toolbox

```
public Toolbox(DrawnView parent)
```

Creates new Toolbox

## Methods

### setLocked

```
public void setLocked(boolean locked)
```

Updates the state of this Toolbox to correspond with the given locked state. When locked, the insert and connect tools are disabled.

### createCategoryToolBars

```
public JToolBar[] createCategoryToolBars()
```

Creates a set of JToolBar instances appropriate for allowing quick selection of types for the Insertion Tool. The buttons included in these toolbars activate the Insertion Tool and set the currently selected insert type.

### createBeanSelectionMenu

```
public void createBeanSelectionMenu(JMenu menu)
```

Creates a select menu appropriate for selecting AbstractDisplayBean instances within the DrawnView parent of this Toolbox.

### addToolChangeListener

```
public void addToolChangeListener(ToolChangeListener listener)
```

Adds the given listener to the the list to notify when the current tool has changed.

**Parameters:**

`listener` - the ToolChangeListener being added to the toolChangeListeners list.

### removeToolChangeListener

```
public void removeToolChangeListener(ToolChangeListener listener)
```

Adds the given listener to the the list to notify when the current tool has changed.

**Parameters:**

`listener` - the ToolChangeListener being removed from the toolChangeListeners list.

### getCurrentCursor

```
public java.awt.Cursor getCurrentCursor()
```

Retrieves the cursor for the views based off of the currently selected tool.

(continued from last page)

---

## setCurrentTool

```
public void setCurrentTool(int tool)
```

switches the toolbox to the given tool.

---

## getCurrentTool

```
public int getCurrentTool()
```

Retrieves the enumerated type of the current tool.

**Returns:**  
the Current tool.

---

## getCurrentCursor

```
public java.awt.Cursor getCurrentCursor( java.awt.event.MouseEvent evt )
```

Retrieves the cursor for the views based off of the currently selected tool.

---

## getComponentCategory

```
public Category getComponentCategory()
```

Retrieves the Category of the currently selected component type in the palette of the Toolbox.

**Parameters:**  
model - the AbstractModel that the retrieved Category is for.

**Returns:**  
the Category selected or null if none is selected.

---

## isCompActionSelected

```
public boolean isCompActionSelected()
```

Returns true if the selected toolbox action will create an AbstractComponent. This method is needed to allow insert handlers to determine if a complete() type method may be called from createSelected().

---

## createSelected

```
public JComponent createSelected()
```

Creates an instance of the currently selected component type in the given model and returns a renderer for the created component.

**Returns:**  
the JComponent being used to render the created component.

**Throws:**  
`IllegalArgumentException` - if the selected component type is not supported by the given model.

---

## setCurrentAction

```
public void setCurrentAction(ToolboxAction action)
```

Sets the current ToolboxAction used for creating new drawings with the Insertion Tool. This method should only be called from ToolboxAction derivatives.

---

(continued from last page)

---

## createCursor

```
public static java.awt.Cursor createCursor(String resource)
```

Creates a Cursor from the image referred to by the given resource name.

**Returns:**

a java.awt.Cursor created with the given resource.

---

## storeDisplayBean

```
public static byte[] storeDisplayBean(com.cafean.client.anim.AbstractDisplayBean bean)
```

XML Encodes the given bean and returns the byte[] result of the encoding.

---

## createDisplayBeans

```
public JComponent[] createDisplayBeans(int pixelsPerMeter,  
    DrawnComponent dc)
```

Attempts to create a set of AbstractDisplayBeans from the given DrawnComponent using DrawnComponent#createDisplayBeans.

---

## createDisplayBeans

```
public JComponent[] createDisplayBeans(int pixelsPerMeter,  
    double widthScaleFactor,  
    DrawnComponent dc)
```

Attempts to create a set of AbstractDisplayBeans from the given DrawnComponent using DrawnComponent#createDisplayBeans.

---

## loadVedaExport

```
public boolean loadVedaExport(java.io.File exportedFile)
```

Attempts to import a VEDA mask dump file. This is an experimental method that is only available during debug mode.

---

## verifyBeansAvailable

```
public static boolean verifyBeansAvailable(Vector records)
```

Verifies that any AbstractDisplayBeans stored in the given Vector as DrawnDisplayBeanRecs have the proper classes available to be loaded.

**Returns:**

true if the display beans in the given Vector can be loaded

---

## loadDisplayBean

```
public static com.cafean.client.anim.AbstractDisplayBean loadDisplayBean(DrawnDisplayBeanRec  
rec)
```

Loads a display bean from the given display bean rec.

---

## findDisplayBeanClass

```
public static Class findDisplayBeanClass(String className)
```

Retrieves the Class with the given class name in the list of Display Bean classes available.

**Returns:**

the Class referred to by className or null.

---

---

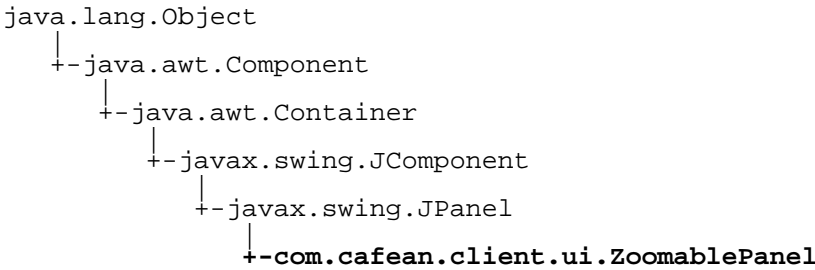
## loadDisplayBeanClasses

```
public static void loadDisplayBeanClasses()
```

Loads the display beans from the jar files included in the Components directories in both the current user's home directory and system wide.

com.cafean.client.ui

# Class ZoomablePanel



All Implemented Interfaces:  
[ToolChangeListener](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
java.io.Serializable, javax.accessibility.Accessible

public class **ZoomablePanel**  
extends JPanel  
implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver,  
java.awt.MenuContainer, java.io.Serializable, [ToolChangeListener](#)

The Zoomable panel is the panel in the DrawnView that contains both the BeanBox for components, and the GlassPane that covers the view. It controls the MouseHandlers based on the ToolDialog's currently selected tool, and handles zooming the view based off of a scaleFactor.

| Fields inherited from class javax.swing.JComponent   |
|--|
| TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |

| Fields inherited from class java.awt.Component                                     |
|--|
| BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT |

| Constructor Summary |   |
|---------------------|---|
| public              | ZoomablePanel(AbstractModel model)<br>Constructs a new Zoomable Panel in a DrawnView. |

| Method Summary |  |
|----------------|--|
| AbstractButton | addMouseHandler(MouseHandler handler)<br>Adds the given handler to the list of zoomable panel handlers and Toolbox handlers. |
| void           | addNotify()<br>Initializes this panel in response to its addition to another component.                                      |

|                              |  |
|------------------------------|--|
| <a href="#">BeanBox</a>      | getBeanBox()<br>The BeanBox contains all the components that are visible inside the DrawnView.                                       |
| <a href="#">DrawnView</a>    | getDrawnView()<br>Each Zoomable Panel appears inside a scroll panel in a DrawnView.  |
| <a href="#">GlassPanel</a>   | getGlassPane()<br>Access routine for the glass pane.   |
| java.awt.Dimension           | getMaximumSize()<br>Return the maximum size of this component.   |
| java.awt.Dimension           | getMinimumSize()<br>Return the minimum size of this component.   |
| <a href="#">MouseHandler</a> | getMouseHandler()<br>Gets the currently active MouseHandler based on the current tool.   |
| java.awt.Dimension           | getPanelSize()<br>The PanelSize is the size of the canvas below this zoomable panel.   |
| java.awt.Dimension           | getPreferredSize()<br>Return the preferred size of this component.   |
| double                       | getScale()<br>The scale factor is the factor by which the current view is multiplied when zooming in or out.                         |
| JScrollPane                  | getScrollPane()<br>Each Zoomable Panel appears inside a scroll panel in a DrawnView.   |
| java.awt.Point               | getViewportCenter()<br>Calculates the center of the currently visible position of the scrollpane containing this ZoomablePanel.      |
| java.awt.Dimension           | getZoomSize()<br>Return the size of scaled components.   |
| java.awt.Point               | inverseTransformPoint(java.awt.Point p1)<br>The reverse transform of a point backwards throught the scaling factor.                  |
| void                         | refresh()<br>Forces a repaint of all children.   |
| void                         | refreshCompBounds(java.awt.Rectangle componentBounds)<br>Repaints the given component-space bounds.                                  |
| void                         | removeMouseHandler(MouseHandler handler)<br>Removes the given handler from the list of zoomable panel handlers and Toolbox handlers. |
| void                         | revalidate()   |
| void                         | setPanelSize(java.awt.Dimension size)<br>Sets the size of the canvas in the view.  |



|                |   |
|----------------|---|
| void           | setScale(double scale)<br>Set the scaling factor property.  |
| void           | setViewCenter(java.awt.Point center)<br>This sets the viewports center to the given Point.  |
| void           | toolChanged(int oldTool,int newTool)<br>Responds to a notification from the Toolbox that the current tool has changed from oldTool to newTool by activating the new MouseHandler, deactivating the old MouseHandler, and updating the cursor. |
| java.awt.Point | transformPoint(java.awt.Point p1)<br>Translate the coordinates of a point to account for the scaling factor.  |
| void           | zoomToFit(java.awt.Rectangle rect)<br>This zooms the current view to fit the given rectangle.   |

**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListeners, addHierarchyChangeListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

(continued from last page)

## ZoomablePanel

```
public ZoomablePanel(AbstractModel model)
```

Constructs a new Zoomable Panel in a DrawnView.

**Parameters:**

model - the AbstractModel that contains the View that owns this panel.

## Methods

### addNotify

```
public void addNotify()
```

Initializes this panel in response to its addition to another component.

### getBeanBox

```
public BeanBox getBeanBox()
```

The BeanBox contains all the components that are visible inside the DrawnView. All routines that manipulate those components, or detail what components appear in the view are ultimately controlled through the BeanBox.

**Returns:**

the BeanBox that contains all the components inside this zoomable panel.

### getGlassPane

```
public GlassPanel getGlassPane()
```

Access routine for the glass pane.

**Returns:**

the GlassPanel that covers this zoomable panel.

### getViewportCenter

```
public java.awt.Point getViewportCenter()
```

Calculates the center of the currently visible position of the scrollpane containing this ZoomablePanel.

**Returns:**

a Point containing the scaled viewport center coordinates.

### toolChanged

```
public void toolChanged(int oldTool,  
                        int newTool)
```

Responds to a notification from the Toolbox that the current tool has changed from oldTool to newTool by activating the new MouseHandler, deactivating the old MouseHandler, and updating the cursor. Tools are one of the TOOL\_\* enumerations in Toolbox.

**Parameters:**

oldTool - the int type of the old tool.

newTool - the int type of the new tool.

**See Also:**

`Toolbox.getCurrentTool()`

---

## getMouseHandler

```
public MouseHandler getMouseHandler()
```

Gets the currently active MouseHandler based on the current tool.

**Returns:**

the MouseHandler used by the current tool.

---

## addMouseHandler

```
public AbstractButton addMouseHandler(MouseHandler handler)
```

Adds the given handler to the list of zoomable panel handlers and Toolbox handlers. This also assigns a proper handler ID to the given handler. Note that this method also creates a toolbar button for the handler and adds the button to the same toolbar as the select and pan tools. The created button is then returned. To move this button to a different toolbar simply remove the button from its parent with `button.getParent().remove(button)`.

**Parameters:**

handler - the MouseHandler to add and assign an ID to.

**Returns:**

the AbstractButton created for the new handler in the toolbox.

**Throws:**

IllegalArgumentException - if the given mouse handler has a handler ID that is already in use.

---

## removeMouseHandler

```
public void removeMouseHandler(MouseHandler handler)
```

Removes the given handler from the list of zoomable panel handlers and Toolbox handlers.

**Parameters:**

handler - the MouseHandler to remove

---

## setPanelSize

```
public void setPanelSize(java.awt.Dimension size)
```

Sets the size of the canvas in the view.

**Parameters:**

size - the Dimension that will be the size of the canvas.

---

## revalidate

```
public void revalidate()
```

---

## refresh

```
public void refresh()
```

Forces a repaint of all children.

---

## refreshCompBounds

```
public void refreshCompBounds(java.awt.Rectangle componentBounds)
```

---

(continued from last page)

Repaints the given component-space bounds. The given bounds are scaled appropriately for this panel's scale factor.  
WARNING: the given rectangle will be modified

---

## getPanelSize

```
public java.awt.Dimension getPanelSize()
```

The PanelSize is the size of the canvas below this zoomable panel. This defines the size of the white canvas inside a view.

**Returns:**

the Dimension of the panel.

---

## getZoomSize

```
public java.awt.Dimension getZoomSize()
```

Return the size of scaled components. This is the panel size scaled by the current scale factor.

**Returns:**

A dimension object indicating this component's preferred size.

---

## getPreferredSize

```
public java.awt.Dimension getPreferredSize()
```

Return the preferred size of this component. This is the panel size scaled by the current scale factor.

**Returns:**

A dimension object indicating this component's preferred size.

---

## getMaximumSize

```
public java.awt.Dimension getMaximumSize()
```

Return the maximum size of this component.

**Returns:**

A dimension object indicating this component's maximum size.

**See Also:**

`.getPreferredSize()`

---

## getMinimumSize

```
public java.awt.Dimension getMinimumSize()
```

Return the minimum size of this component.

**Returns:**

A dimension object indicating this component's minimum size.

**See Also:**

`.getPreferredSize()`

---

## setScale

```
public void setScale(double scale)
```

Set the scaling factor property.

**Parameters:**

`scale` - the new scaling factor.

---

## getScale

```
public double getScale()
```

The scale factor is the factor by which the current view is multiplied when zooming in or out.

**Returns:**

the double that is the current scale factor.

---

## inverseTransformPoint

```
public java.awt.Point inverseTransformPoint(java.awt.Point p1)
```

The reverse transform of a point backwards through the scaling factor.

**Parameters:**

p1 - the Point to be inverse transformed by the scaling factor.

**Returns:**

A Point that represents p1 inverse transformed by the scaling factor.

---

## transformPoint

```
public java.awt.Point transformPoint(java.awt.Point p1)
```

Translate the coordinates of a point to account for the scaling factor.

**Parameters:**

p1 - the Point to be transformed by the scaling factor.

**Returns:**

A Point that represents p1 transformed by the scaling factor.

---

## getDrawnView

```
public DrawnView getDrawnView()
```

Each Zoomable Panel appears inside a scroll panel in a DrawnView. This accessor finds the closest ancestor of this panel that is a DrawnView, and returns it.

**Returns:**

the DrawnView parent of this zoomable panel.

---

## getScrollPane

```
public JScrollPane getScrollPane()
```

Each Zoomable Panel appears inside a scroll panel in a DrawnView. This accessor finds the closest ancestor of this panel that is a JScrollPane, and returns it.

**Returns:**

the JScrollPane parent of this zoomable panel.

---

## setViewCenter

```
public void setViewCenter(java.awt.Point center)
```

This sets the viewports center to the given Point. This adds the setViewPosition to the event stack, which makes sure that all other current events are finished before the view's position gets set. This is to ensure that during a Zoom operation the viewport center gets set last.

**Parameters:**

(continued from last page)

`center` - the Point in current zoomed coordinates that should be the center.

---

## **zoomToFit**

```
public void zoomToFit(java.awt.Rectangle rect)
```

This zooms the current view to fit the given rectangle. The viewport in the JScrollPane will be set to the center of the rectangle after the view is zoomed.

### **Parameters:**

`rect` - the Rectangle in current zoomed coordinates.

---

## Package

# **com.cafean.client.ui.annotation**

This package contains the ModelEditor provided Annotation elements for views. Documentation for these classes is provided to allow for their use in generating Animation plug-in displays from plug-in specific DrawnComponent instances.



## com.cafean.client.ui.annotation Class Annotation

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JPanel
          |-- com.cafean.client.ui.annotation.Annotation

```

### All Implemented Interfaces:

java.awt.event.MouseListener, java.io.Serializable, StateEditable, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

### Direct Known Subclasses:

[ImageAnnotation](#), [TextAnnotation](#), [LineAnnotation](#), [RectangularAnnotation](#), [EllipticalAnnotation](#)

public abstract class **Annotation**

extends JPanel

implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, StateEditable, java.io.Serializable, java.awt.event.MouseListener

This Abstract class should be extended by any object that is used only to impart data to an analyst looking at a [com.cafean.client.ui.DrawnView](#).

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | Annotation()<br>Creates a new instance of Annotation |
|--------|--|

## Method Summary

|                           |   |
|---------------------------|---|
| void                      | addPopupMenuItems(JPopupMenu menu, java.awt.event.MouseEvent evt)   |
| <a href="#">DrawnView</a> | getDrawnView()<br>Retrieves the DrawnView that contains this DrawnComponent or null if this DrawnComponent exists outside of a DrawnView. |

|                    |   |
|--------------------|---|
| void               | mouseClicked( java.awt.event.MouseEvent e )   |
| void               | mouseEntered( java.awt.event.MouseEvent e )   |
| void               | mouseExited( java.awt.event.MouseEvent e )  |
| void               | mousePressed( java.awt.event.MouseEvent e )   |
| void               | mouseReleased( java.awt.event.MouseEvent e )  |
| void               | moveTo( double x, double y, boolean last )<br>Move the Annotation so the center is the specified position |
| abstract void      | popupEditor()<br>This is used for Annotations to generate their editing dialogs.                          |
| void               | refresh( java.awt.Rectangle r )<br>This refreshes a given Rectangle that indicates a dirty region.        |
| abstract void      | resetSize()<br>This resets the current size of the annotation.  |
| void               | restoreState( Hashtable state )<br>Restore the state of the bean from an earlier edit.                    |
| void               | setBorder( Border border )  |
| void               | setBounds( java.awt.Rectangle bounds )  |
| void               | setHeight( int height )   |
| void               | setWidth( int width )   |
| DrawnAnnotationRec | store()   |
| void               | storeState( Hashtable state )<br>Store the state of the bean to permit undo.                              |
| String             | toString()  |

**Methods inherited from class** javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class** javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### Annotation

```
public Annotation()
```

Creates a new instance of Annotation

## Methods

### getDrawnView

```
public DrawnView getDrawnView()
```

Retrieves the DrawnView that contains this DrawnComponent or null if this DrawnComponent exists outside of a DrawnView.

(continued from last page)

**Returns:**

the DrawnView at the top of the ancestor list.

---

**setBorder**

```
public void setBorder(Border border)
```

---

**setHeight**

```
public void setHeight(int height)
```

---

**setWidth**

```
public void setWidth(int width)
```

---

**setBounds**

```
public void setBounds(java.awt.Rectangle bounds)
```

---

**store**

```
public DrawnAnnotationRec store()
```

---

**storeState**

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo. NOTE: If the component storing its state needs a deep copy that its clone() method does not provide, it must override storeState to find that functionality elsewhere.

**Parameters:**

state - A hash table containing modified parameters.

---

**restoreState**

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

**Parameters:**

state - A hash table containing modified parameters.

---

**popupEditor**

```
public abstract void popupEditor()
```

This is used for Annotations to generate their editing dialogs. Since these are not components, they cannot use the Component View editor. This should handle creating the undo object and saving the current state before implementing changes

(continued from last page)

---

## moveTo

```
public void moveTo(double x,  
                  double y,  
                  boolean last)
```

Move the Annotation so the center is the specified position

### Parameters:

- x - the new center x position
- y - the new center y position
- last - false while moving, true on the final move

---

## resetSize

```
public abstract void resetSize()
```

This resets the current size of the annotation. This ensures that all annotations do not lose any data when the [scale](#) changes.

---

## addPopupMenuItems

```
public void addPopupMenuItems(JPopupMenu menu,  
                               java.awt.event.MouseEvent evt)
```

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

---

## mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent e)
```

---

## mouseExited

```
public void mouseExited(java.awt.event.MouseEvent e)
```

---

## mousePressed

```
public void mousePressed(java.awt.event.MouseEvent e)
```

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent e)
```

---

## refresh

```
public void refresh(java.awt.Rectangle r)
```

This refreshes a given Rectangle that indicates a dirty region.

### Parameters:

(continued from last page)

`r` - the Rectangle indicating the dirty region to be refreshed.

**See Also:**

`GlassPanel.repaint()`

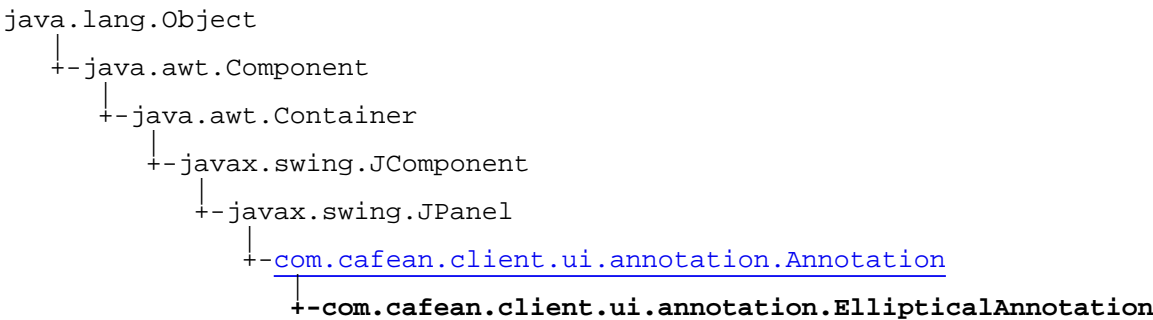
---

## **toString**

```
public String toString()
```

# com.cafean.client.ui.annotation

## Class EllipticalAnnotation



All Implemented Interfaces:  
[Insertable](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, java.awt.event.MouseListener, java.io.Serializable, StateEditable

public class **EllipticalAnnotation**  
extends [Annotation](#)  
implements StateEditable, java.io.Serializable, java.awt.event.MouseListener,  
javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver,  
java.awt.MenuContainer, java.io.Serializable, [Insertable](#)

A configurable drawn ellipse.

|  |
|--|
| Fields inherited from class javax.swing.JComponent   |
| TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |

|  |
|--|
| Fields inherited from class java.awt.Component                                     |
| BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT |

| Constructor Summary |                        |   |
|---------------------|------------------------|---|
| public              | EllipticalAnnotation() | Creates a new instance of RectangularAnnotation |

| Method Summary                        |   |                                    |
|---------------------------------------|---|------------------------------------|
| int                                   | getLineThickness()                        | Getter for property lineThickness. |
| <a href="#">AbstractInsertHandler</a> | getNewInsertHandler(ZoomablePanel parent) |                                    |
| boolean                               | isFilled()                                |                                    |



|      |   |
|------|---|
| void | paintComponent( java.awt.Graphics g)<br>{ @inheritDoc}  |
| void | popupEditor()<br>{ @inheritDoc}   |
| void | resetSize()<br>{ @inheritDoc}   |
| void | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.              |
| void | setEqualTo(EllipticalAnnotation annotation)<br>Sets this annotation equal to the given annotation |
| void | setFilled(boolean filled)   |
| void | setLineThickness(int lineThickness)<br>Setter for property lineThickness.                         |
| void | setOpaque(boolean filled)   |
| void | storeState(Hashtable state)   |

**Methods inherited from class [com.cafean.client.ui.annotation.Annotation](#)**

[addPopupMenuItems](#), [getDrawnView](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mousePressed](#), [mouseReleased](#), [moveTo](#), [popupEditor](#), [refresh](#), [resetSize](#), [restoreState](#), [setBorder](#), [setBounds](#), [setHeight](#), [setWidth](#), [store](#), [storeState](#), [toString](#)

**Methods inherited from class javax.swing.JPanel**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### EllipticalAnnotation

```
public EllipticalAnnotation()
```

Creates a new instance of RectangularAnnotation

## Methods

### setOpaque

```
public void setOpaque(boolean filled)
```

---

## setFilled

```
public void setFilled(boolean filled)
```

---

## isFilled

```
public boolean isFilled()
```

---

---

## paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

---

---

## popupEditor

```
public void popupEditor()
```

This is used for Annotations to generate their editing dialogs. Since these are not components, they cannot use the Component View editor. This should handle creating the undo object and saving the current state before implementing changes

---

## setEqualTo

```
public void setEqualTo(EllipticalAnnotation annotation)
```

Sets this annotation equal to the given annotation

---

## storeState

```
public void storeState(Hashtable state)
```

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

### Parameters:

state - A hash table containing modified parameters.

---

## resetSize

```
public void resetSize()
```

This resets the current size of the annotation. This ensures that all annotations do not lose any data when the [scale](#) changes.

---

## getLineThickness

```
public int getLineThickness()
```

Getter for property lineThickness.

### Returns:

Value of property lineThickness.

---

---

## setLineThickness

```
public void setLineThickness(int lineThickness)
```

Setter for property lineThickness.

### Parameters:

lineThickness - New value of property lineThickness.

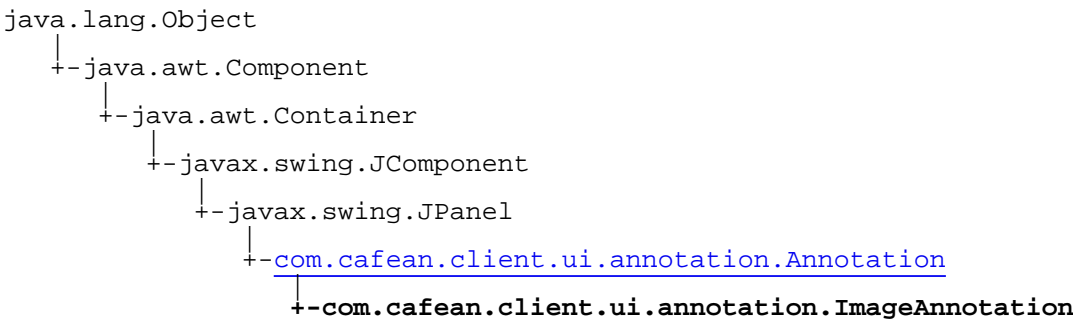
---

## getNewInsertHandler

```
public AbstractInsertHandler getNewInsertHandler(ZoomablePanel parent)
```

# com.cafean.client.ui.annotation

## Class ImageAnnotation



**All Implemented Interfaces:**  
java.awt.event.ComponentListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, java.awt.event.MouseListener, java.io.Serializable, StateEditable

public class **ImageAnnotation**  
extends [Annotation](#)  
implements StateEditable, java.io.Serializable, java.awt.event.MouseListener,  
javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver,  
java.awt.MenuContainer, java.io.Serializable, java.awt.event.ComponentListener

| Fields inherited from class javax.swing.JComponent   |
|--|
| TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |

| Fields inherited from class java.awt.Component                                     |
|--|
| BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT |

| Constructor Summary |                   |  |
|---------------------|-------------------|--|
| public              | ImageAnnotation() | Creates a new instance of DrawnComponentView |

| Method Summary |   |  |
|----------------|---|--|
| void           | addPopupMenuItems(JPopupMenu menu, java.awt.event.MouseEvent evt) |  |
| void           | componentHidden(java.awt.event.ComponentEvent e)                  |  |
| void           | componentMoved(java.awt.event.ComponentEvent e)                   |  |

|                         |  |
|-------------------------|--|
| void                    | componentResized( java.awt.event.ComponentEvent e )  |
| void                    | componentShown( java.awt.event.ComponentEvent e )  |
| DrawnImageAnnotationRec | createPibBlock()   |
| byte[]                  | getImageData()<br>Retrieves this annotation's image data.  |
| java.awt.Dimension      | getMinimumSize()   |
| void                    | paintComponent( java.awt.Graphics g )<br>Draw a simple X'd box if there is no image.   |
| void                    | popupEditor()  |
| void                    | reset()<br>Creates a draw view component and puts it in the drawing vectors.   |
| void                    | resetSize()  |
| void                    | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.   |
| void                    | setImageData(byte[] imageData)<br>Sets this annotation's image data and resizes to the size of the image in imageData.                           |
| void                    | setImageData(byte[] imageData,boolean resize)<br>Sets this annotation's image data and optionally resizes to the size of the image in imageData. |
| void                    | storeState(Hashtable state)<br>Store the state of the bean to permit undo.   |

**Methods inherited from class [com.cafean.client.ui.annotation.Annotation](#)**

[addPopupMenuItems](#), [getDrawnView](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mousePressed](#), [mouseReleased](#), [moveTo](#), [popupEditor](#), [refresh](#), [resetSize](#), [restoreState](#), [setBorder](#), [setBounds](#), [setHeight](#), [setWidth](#), [store](#), [storeState](#), [toString](#)

**Methods inherited from class javax.swing.JPanel**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component



```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### ImageAnnotation

```
public ImageAnnotation()
```

Creates a new instance of DrawnComponentView

## Methods

### addPopupMenuItems

```
public void addPopupMenuItems(JPopupMenu menu,
    java.awt.event.MouseEvent evt)
```

---

## getMinimumSize

```
public java.awt.Dimension getMinimumSize()
```

---

## reset

```
public void reset()
```

Creates a draw view component and puts it in the drawing vectors.

---

## paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

Draw a simple X'd box if there is no image.

---

## componentResized

```
public void componentResized(java.awt.event.ComponentEvent e)
```

---

## componentHidden

```
public void componentHidden(java.awt.event.ComponentEvent e)
```

---

## componentMoved

```
public void componentMoved(java.awt.event.ComponentEvent e)
```

---

## componentShown

```
public void componentShown(java.awt.event.ComponentEvent e)
```

---

## popupEditor

```
public void popupEditor()
```

---

## resetSize

```
public void resetSize()
```

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

### Parameters:

state - A hash table containing modified parameters.

---

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

### Parameters:

state - A hash table containing modified parameters.

---

## createPibBlock

```
public DrawnImageAnnotationRec createPibBlock()
```

---

## setImageData

```
public void setImageData(byte[] imageData)
```

Sets this annotation's image data and resizes to the size of the image in imageData.

### See Also:

```
.setImageData(byte[],boolean)()
```

---

## setImageData

```
public void setImageData(byte[] imageData,  
    boolean resize)
```

Sets this annotation's image data and optionally resizes to the size of the image in imageData.

### Parameters:

imageData - a byte[] containing a Java-Readable image in appropriate for loading with new ImageIcon(imageData);  
resize - if true this annotation will adjust its size to that of the given image.

---

## getImageData

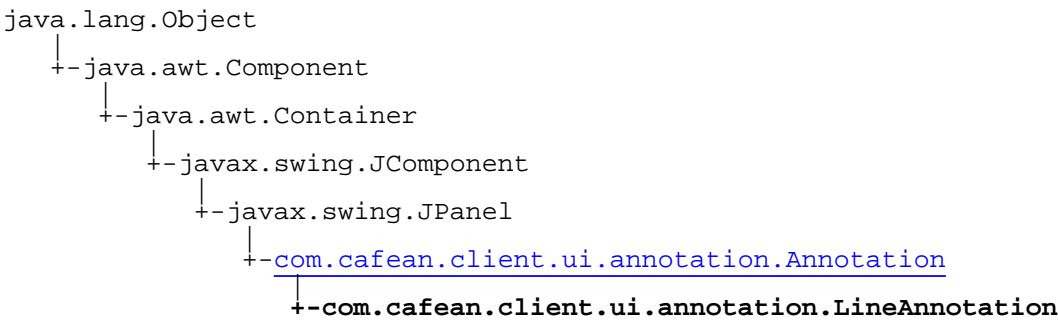
```
public byte[] getImageData()
```

Retrieves this annotation's image data. Image data is the byte[] data read from file to create this annotation's image.

---

# com.cafean.client.ui.annotation

## Class LineAnnotation



All Implemented Interfaces:  
[Insertable](#), [FullScreenDrawing](#), java.awt.event.MouseMotionListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, java.awt.event.MouseListener, java.io.Serializable, StateEditable

public class **LineAnnotation**  
extends [Annotation](#)  
implements StateEditable, java.io.Serializable, java.awt.event.MouseListener, javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.awt.event.MouseMotionListener, [FullScreenDrawing](#), [Insertable](#)

A configurable drawn line.

| Field Summary         |                          |
|-----------------------|--------------------------|
| static final int      | ARROW_FILLED<br>Value: 1 |
| static final int      | ARROW_HOLLOW<br>Value: 2 |
| static final int      | ARROW_LINE<br>Value: 4   |
| static final String[] | ARROW_NAMES              |
| static final int      | ARROW_NONE<br>Value: 0   |
| static final int[]    | ARROW_TYPES              |

| Fields inherited from class javax.swing.JComponent   |
|--|
| TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |

**Fields inherited from class** `java.awt.Component`

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|                     |  |
|---------------------|--|
| <code>public</code> | <code>LineAnnotation()</code><br>Creates a new instance of <code>LineAnnotation</code> |
|---------------------|--|

## Method Summary

|  |  |
|--|--|
| <code>boolean</code>                               | <code>addPoint(java.awt.Point p)</code><br>Adds a path point at the given point, if the point falls on an existing line segment. |
| <code>void</code>                                  | <code>addPopupMenuItems(JPopupMenu menu, java.awt.event.MouseEvent evt)</code>   |
| <code>boolean</code>                               | <code>canAddPoint(java.awt.Point p)</code><br>Return true if a point can be added to the path.                                   |
| <code>boolean</code>                               | <code>canRemovePoint(java.awt.Point p)</code><br>Return true if a point can be removed from the path.                            |
| <code>boolean</code>                               | <code>contains(double x, double y)</code>  |
| <code>boolean</code>                               | <code>contains(int x, int y)</code>  |
| <code>boolean</code>                               | <code>contains(java.awt.Point p)</code>  |
| <code>int[]</code>                                 | <code>getArrows()</code><br>Getter for property <code>arrows</code> .  |
| <code>float</code>                                 | <code>getArrowSize()</code><br>Getter for property <code>arrowSize</code> .  |
| <code>static String</code>                         | <code>getArrowTypeName(int type)</code>  |
| <code>int</code>                                   | <code>getHead1()</code>  |
| <code>int</code>                                   | <code>getHead2()</code>  |
| <code>int</code>                                   | <code>getLineThickness()</code><br>Getter for property <code>lineThickness</code> .  |
| <a href="#"><code>AbstractInsertHandler</code></a> | <code>getNewInsertHandler(ZoomablePanel parent)</code>   |
| <code>java.awt.Point[]</code>                      | <code>getPath()</code><br>Retrieves a copy of the plotted path points that this <code>DrawnConnection</code> is painting.        |

|                    |   |
|--------------------|---|
| String             | getToolTipText()<br>{ @inheritDoc }   |
| java.awt.Rectangle | getUsedBounds()   |
| boolean            | isDashed()<br>Getter for property dashed.   |
| boolean            | isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)<br>{ @inheritDoc }  |
| void               | mouseClicked(java.awt.event.MouseEvent e)   |
| void               | mouseDragged(java.awt.event.MouseEvent e)<br>Handle mouseDragged events for segment and point manipulation            |
| void               | mouseEntered(java.awt.event.MouseEvent e)   |
| void               | mouseExited(java.awt.event.MouseEvent e)  |
| void               | mouseMoved(java.awt.event.MouseEvent e)<br>{ @inheritDoc }  |
| void               | mousePressed(java.awt.event.MouseEvent e)<br>Handle mousePressed events to support segment and point manipulation     |
| void               | mouseReleased(java.awt.event.MouseEvent e)<br>Handle mouseReleased events to support segment and point manipulation   |
| void               | moveTo(double x,double y,boolean last)<br>Move the Annotation so the center is the specified position                 |
| void               | paintComponent(java.awt.Graphics g)   |
| void               | popupEditor()   |
| void               | removeClosestPoint(java.awt.Point p)<br>Attempts to remove the closest point in this DrawnConnection's set of points. |
| boolean            | removePoint(java.awt.Point p)<br>Attempts to remove the path point at the given location.                             |
| void               | repaint()   |
| void               | resetSize()   |
| void               | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.                                  |
| void               | revalidate()  |

|      |   |
|------|---|
| void | setArrows(int[] arrows)<br>Setter for property arrows.  |
| void | setArrowSize(float arrowSize)<br>Setter for property arrowSize.   |
| void | setDashed(boolean dashed)<br>Setter for property dashed.  |
| void | setEqualTo(LineAnnotation annotation)<br>Sets this annotation equal to the given annotation               |
| void | setHead1(int head1)   |
| void | setHead2(int head2)   |
| void | setLineThickness(int lineThickness)<br>Setter for property lineThickness.                                 |
| void | setPath(java.awt.Point[] path)<br>Sets the path points that this DrawnConnection will use for connecting. |
| void | storeState(Hashtable state)<br>Store the state of the bean to permit undo.                                |
| void | translate(int dx,int dy)  |
| void | validate()  |

**Methods inherited from class [com.cafean.client.ui.annotation.Annotation](#)**

[addPopupMenuItems](#), [getDrawnView](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mousePressed](#), [mouseReleased](#), [moveTo](#), [popupEditor](#), [refresh](#), [resetSize](#), [restoreState](#), [setBorder](#), [setBounds](#), [setHeight](#), [setWidth](#), [store](#), [storeState](#), [toString](#)

**Methods inherited from class [javax.swing.JPanel](#)**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class [javax.swing.JComponent](#)**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component



```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

**ARROW\_NONE**

```
public static final int ARROW_NONE
```

**ARROW FILLED**

```
public static final int ARROW FILLED
```

(continued from last page)

---

## ARROW\_HOLLOW

```
public static final int ARROW_HOLLOW
```

---

## ARROW\_LINE

```
public static final int ARROW_LINE
```

---

## ARROW\_NAMES

```
public static final java.lang.String ARROW_NAMES
```

---

## ARROW\_TYPES

```
public static final int ARROW_TYPES
```

---

## Constructors

### LineAnnotation

```
public LineAnnotation()
```

Creates a new instance of LineAnnotation

---

## Methods

### getArrowTypeName

```
public static String getArrowTypeName(int type)
```

---

### paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

---

### moveTo

```
public void moveTo(double x,  
    double y,  
    boolean last)
```

Move the Annotation so the center is the specified position

#### Parameters:

- x - the new center x position
  - y - the new center y position
  - last - false while moving, true on the final move
-

(continued from last page)

## contains

```
public boolean contains(double x,  
                        double y)
```

---

## contains

```
public boolean contains(java.awt.Point p)
```

---

## contains

```
public boolean contains(int x,  
                        int y)
```

---

## getPath

```
public java.awt.Point[] getPath()
```

Retrieves a copy of the plotted path points that this DrawnConnection is painting.

**Returns:**

a Point[] containing the coordinates of each point.

---

## setPath

```
public void setPath(java.awt.Point[] path)
```

Sets the path points that this DrawnConnection will use for connecting.

---

## popupEditor

```
public void popupEditor()
```

---

## setEqualTo

```
public void setEqualTo(LineAnnotation annotation)
```

Sets this annotation equal to the given annotation

---

## mousePressed

```
public void mousePressed(java.awt.event.MouseEvent e)
```

Handle mousePressed events to support segment and point manipulation

**See Also:**

`.mouseReleased()`

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

---

(continued from last page)

---

## removeClosestPoint

```
public void removeClosestPoint(java.awt.Point p)
```

Attempts to remove the closest point in this DrawnConnection's set of points.

**Parameters:**

p - the Point to remove the closest path point.

---

## removePoint

```
public boolean removePoint(java.awt.Point p)
```

Attempts to remove the path point at the given location.

**Parameters:**

p - the Point at which to delete a path point.

**Returns:**

true if the point was deleted.

---

## canRemovePoint

```
public boolean canRemovePoint(java.awt.Point p)
```

Return true if a point can be removed from the path.

---

## canAddPoint

```
public boolean canAddPoint(java.awt.Point p)
```

Return true if a point can be added to the path.

---

## addPoint

```
public boolean addPoint(java.awt.Point p)
```

Adds a path point at the given point, if the point falls on an existing line segment.

**Parameters:**

p - the Point at which to add a path point.

**Returns:**

true if the point is actually added.

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent e)
```

Handle mouseReleased events to support segment and point manipulation

**See Also:**

.mousePressed()

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent e)
```

Handle mouseDragged events for segment and point manipulation

**See Also:**

.mousePressed()

---

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent e)
    { @inheritDoc }
```

---

## mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent e)
```

---

## mouseExited

```
public void mouseExited(java.awt.event.MouseEvent e)
```

---

## getToolTipText

```
public String getToolTipText()
```

---

## isObjectInsideBounds

```
public boolean isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)
    { @inheritDoc }
```

---

## repaint

```
public void repaint()
```

---

## translate

```
public void translate(int dx,
    int dy)
```

---

## getUsedBounds

```
public java.awt.Rectangle getUsedBounds()
```

---

## resetSize

```
public void resetSize()
```

---

## getLineThickness

```
public int getLineThickness()
    Getter for property lineThickness.
```

**Returns:**

Value of property lineThickness.

---

---

## setLineThickness

```
public void setLineThickness(int lineThickness)
```

Setter for property lineThickness.

### Parameters:

lineThickness - New value of property lineThickness.

---

## addPopupMenuItems

```
public void addPopupMenuItems(JPopupMenu menu,  
    java.awt.event.MouseEvent evt)
```

---

## revalidate

```
public void revalidate()
```

---

## validate

```
public void validate()
```

---

## storeState

```
public void storeState(Hashtable state)
```

Store the state of the bean to permit undo.

### Parameters:

state - A hash table containing modified parameters.

---

## restoreState

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

### Parameters:

state - A hash table containing modified parameters.

---

## getArrows

```
public int[] getArrows()
```

Getter for property arrows.

### Returns:

Value of property arrows.

---

## setArrows

```
public void setArrows(int[] arrows)
```

Setter for property arrows.

### Parameters:

---

(continued from last page)

arrows - New value of property arrows.

---

## getArrowSize

```
public float getArrowSize()
```

Getter for property arrowSize.

### Returns:

Value of property arrowSize.

---

## setArrowSize

```
public void setArrowSize(float arrowSize)
```

Setter for property arrowSize.

### Parameters:

arrowSize - New value of property arrowSize.

---

## isDashed

```
public boolean isDashed()
```

Getter for property dashed.

### Returns:

Value of property dashed.

---

## setDashed

```
public void setDashed(boolean dashed)
```

Setter for property dashed.

### Parameters:

dashed - New value of property dashed.

---

## getHead1

```
public int getHead1()
```

---

## setHead1

```
public void setHead1(int head1)
```

---

## getHead2

```
public int getHead2()
```

---

## setHead2

```
public void setHead2(int head2)
```

---

(continued from last page)

## **getNewInsertHandler**

```
public AbstractInsertHandler getNewInsertHandler(ZoomablePanel parent)
```



# com.cafean.client.ui.annotation

## Class RectangularAnnotation



**All Implemented Interfaces:**  
[Insertable](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, java.awt.event.MouseListener, java.io.Serializable, StateEditable

public class **RectangularAnnotation**  
extends [Annotation](#)  
implements StateEditable, java.io.Serializable, java.awt.event.MouseListener,  
javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver,  
java.awt.MenuContainer, java.io.Serializable, [Insertable](#)

A configurable drawn rectangle with border, thickness and optional rounded corners.

|  |  |
|--|--|
| Fields inherited from class javax.swing.JComponent   |  |
| TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |  |
| Fields inherited from class java.awt.Component   |  |
| BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT                               |  |

| Constructor Summary |                         |   |
|---------------------|-------------------------|---|
| public              | RectangularAnnotation() | Creates a new instance of RectangularAnnotation |

| Method Summary |                |                                |
|----------------|----------------|--------------------------------|
| double         | getArcHeight() | Getter for property arcHeight. |
| double         | getArcWidth()  | Getter for property arcWidth.  |

|                                       |  |
|---------------------------------------|--|
| int                                   | getLineThickness()<br>Getter for property lineThickness.   |
| <a href="#">AbstractInsertHandler</a> | getNewInsertHandler(ZoomablePanel parent)  |
| boolean                               | isFilled()   |
| boolean                               | isOpaque()   |
| boolean                               | isRounded()<br>Returns true if this annotation has rounded corners.                                |
| void                                  | paintComponent(java.awt.Graphics g)  |
| void                                  | popupEditor()  |
| void                                  | resetSize()<br>{ @inheritDoc }   |
| void                                  | setArcHeight(double arcHeight)<br>Setter for property arcHeight.                                   |
| void                                  | setArcWidth(double arcWidth)<br>Setter for property arcWidth.                                      |
| void                                  | setEqualTo(RectangularAnnotation annotation)<br>Sets this annotation equal to the given annotation |
| void                                  | setFilled(boolean filled)  |
| void                                  | setLineThickness(int lineThickness)<br>Setter for property lineThickness.                          |
| void                                  | setRounded(boolean rounded)<br>Sets this annotations rounded corner property.                      |

**Methods inherited from class [com.cafean.client.ui.annotation.Annotation](#)**

[addPopupMenuItems](#), [getDrawnView](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mousePressed](#), [mouseReleased](#), [moveTo](#), [popupEditor](#), [refresh](#), [resetSize](#), [restoreState](#), [setBorder](#), [setBounds](#), [setHeight](#), [setWidth](#), [store](#), [storeState](#), [toString](#)

**Methods inherited from class javax.swing.JPanel**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### RectangularAnnotation

```
public RectangularAnnotation()
```

Creates a new instance of RectangularAnnotation

## Methods

### paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

## setFilled

```
public void setFilled(boolean filled)
```

---

## isFilled

```
public boolean isFilled()
```

---

## isOpaque

```
public boolean isOpaque()
```

---

## popupEditor

```
public void popupEditor()
```

---

## setEqualTo

```
public void setEqualTo(RectangularAnnotation annotation)
```

Sets this annotation equal to the given annotation

---

## resetSize

```
public void resetSize()
```

This resets the current size of the annotation. This ensures that all annotations do not lose any data when the [scale](#) changes.

---

## isRounded

```
public boolean isRounded()
```

Returns true if this annotation has rounded corners.

---

## setRounded

```
public void setRounded(boolean rounded)
```

Sets this annotations rounded corner property.

---

## getArcWidth

```
public double getArcWidth()
```

Getter for property arcWidth.

### Returns:

Value of property arcWidth.

---

## setArcWidth

```
public void setArcWidth(double arcWidth)
```

Setter for property arcWidth.

---

(continued from last page)

**Parameters:**

arcWidth - New value of property arcWidth.

---

**getArcHeight**

```
public double getArcHeight()
```

Getter for property arcHeight.

**Returns:**

Value of property arcHeight.

---

**setArcHeight**

```
public void setArcHeight(double arcHeight)
```

Setter for property arcHeight.

**Parameters:**

arcHeight - New value of property arcHeight.

---

**getLineThickness**

```
public int getLineThickness()
```

Getter for property lineThickness.

**Returns:**

Value of property lineThickness.

---

**setLineThickness**

```
public void setLineThickness(int lineThickness)
```

Setter for property lineThickness.

**Parameters:**

lineThickness - New value of property lineThickness.

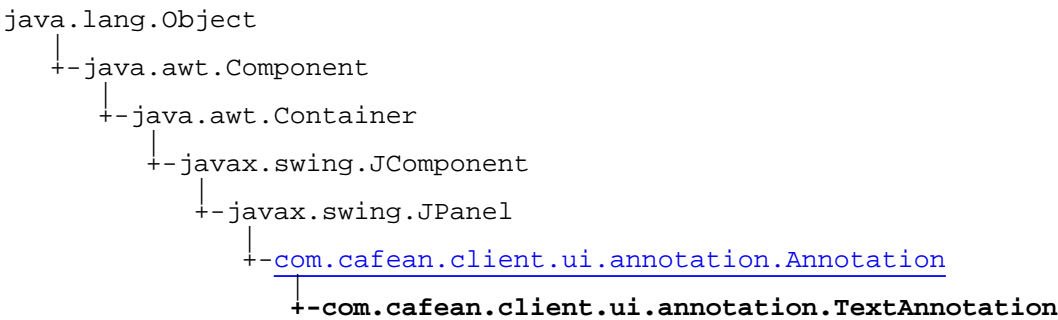
---

**getNewInsertHandler**

```
public AbstractInsertHandler getNewInsertHandler(ZoomablePanel parent)
```

com.cafean.client.ui.annotation

Class TextAnnotation



public class **TextAnnotation**  
extends [Annotation](#)

|  |
|--|
| <b>Fields inherited from class</b> javax.swing.JComponent  |
| TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |

|  |
|--|
| <b>Fields inherited from class</b> java.awt.Component                              |
| BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT |

| Constructor Summary |  |
|---------------------|--|
| public              | TextAnnotation()<br>Creates a new instance of TextAnnotation |

| Method Summary |   |
|----------------|---|
| void           | ensureVersion21()<br>Updates the font, text and color values of this TextAnnotation to compensate for an older version which used only the JLabel properties. |
| java.awt.Color | getBackground()   |
| java.awt.Font  | getFont()   |
| java.awt.Color | getForeground()   |
| JLabel         | getLabel()<br>Getter for property label.  |

|        |   |
|--------|---|
| String | getText()   |
| void   | popupEditor()   |
| void   | resetSize()<br>Resets the size of the label and the text annotation to be slightly larger then the preferred size of the label. |
| void   | restoreState(Hashtable state)<br>Restore the state of the bean from an earlier edit.  |
| void   | setBackground(java.awt.Color bg)  |
| void   | setFont(java.awt.Font font)   |
| void   | setForeground(java.awt.Color fg)  |
| void   | setLabel(JLabel label)<br>Setter for property label.  |
| void   | setSize(java.awt.Dimension size)  |
| void   | setText(String text)  |
| void   | storeState(Hashtable state)<br>Store the state of the bean to permit undo.  |

**Methods inherited from class [com.cafean.client.ui.annotation.Annotation](#)**

[addPopupMenuItems](#), [getDrawnView](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mousePressed](#), [mouseReleased](#), [moveTo](#), [popupEditor](#), [refresh](#), [resetSize](#), [restoreState](#), [setBorder](#), [setBounds](#), [setHeight](#), [setWidth](#), [store](#), [storeState](#), [toString](#)

**Methods inherited from class javax.swing.JPanel**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class javax.swing.JComponent**



addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### TextAnnotation

public **TextAnnotation**()

Creates a new instance of TextAnnotation

## Methods

### storeState

public void **storeState**(Hashtable state)

Store the state of the bean to permit undo. NOTE: If the component storing its state needs a deep copy that it's clone() method does not provide, it must override storeState to find that functionality elsewhere.

(continued from last page)

**Parameters:**

state - A hash table containing modified parameters.

---

**restoreState**

```
public void restoreState(Hashtable state)
```

Restore the state of the bean from an earlier edit.

**Parameters:**

state - A hash table containing modified parameters.

---

**popupEditor**

```
public void popupEditor()
```

---

**getText**

```
public String getText()
```

---

**setText**

```
public void setText(String text)
```

---

**setFont**

```
public void setFont(java.awt.Font font)
```

---

**getFont**

```
public java.awt.Font getFont()
```

---

**getForeground**

```
public java.awt.Color getForeground()
```

---

**getBackground**

```
public java.awt.Color getBackground()
```

---

**setForeground**

```
public void setForeground(java.awt.Color fg)
```

---

(continued from last page)

## setBackground

```
public void setBackground(java.awt.Color bg)
```

---

## getLabel

```
public JLabel getLabel()
```

Getter for property label.

**Returns:**

Value of property label.

---

## setLabel

```
public void setLabel(JLabel label)
```

Setter for property label.

**Parameters:**

label - New value of property label.

---

## resetSize

```
public void resetSize()
```

Resets the size of the label and the text annotation to be slightly larger then the preferred size of the label.

---

## setSize

```
public void setSize(java.awt.Dimension size)
```

---

## ensureVersion21

```
public void ensureVersion21()
```

Updates the font, text and color values of this TextAnnotation to compensate for an older version which used only the JLabel properties.

**Since:**

0.21.0

---

---

## Package

# com.cafean.client.ui.beans

This package contains the bean-based editors provided for use in plug-ins.

Plug-in writers should take note of the following classes before creating editors for particular types.

- [BooleanEditor](#)
- [BorderStyleEditor](#)
- [ComponentSelectionEditor](#)
- [IntrospectingEditor](#)
- [NamedIntEditor](#)
- [PropertyController](#)
- [RealArrayEditor](#)
- [RealBeanEditor](#)

## com.cafean.client.ui.beans

### Class BooleanEditor

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JPanel
                          |-- com.cafean.client.ui.beans.BooleanEditor

```

#### All Implemented Interfaces:

java.beans.PropertyEditor, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

#### Direct Known Subclasses:

[NamelistBooleanEditor](#)

public class **BooleanEditor**

extends JPanel

implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.beans.PropertyEditor

This is the radio-button form of a Boolean value editor used in the PropertySetPanel and its associated components.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | BooleanEditor()<br>Creates a new instance of CellComponentSelector |
|--------|--|

## Method Summary

|                    |  |
|--------------------|--|
| String             | getAsText()  |
| java.awt.Component | getCustomEditor()  |
| String             | getJavaInitializationString()<br>Does nothing, and returns null. |

|          |   |
|----------|---|
| String[] | getTags()<br>Does nothing, and returns null.              |
| Object   | getValue()  |
| boolean  | isPaintable()   |
| void     | paintValue(java.awt.Graphics gfx, java.awt.Rectangle box) |
| void     | setAsText(String text)                                    |
| void     | setValue(Object value)                                    |
| boolean  | supportsCustomEditor()                                    |

**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

```

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt,
getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners,
getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets,
invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove,
remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy,
setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward,
transferFocusDownCycle, update, validate

```

#### Methods inherited from class java.awt.Component

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors



(continued from last page)

## BooleanEditor

```
public BooleanEditor()
```

Creates a new instance of CellComponentSelector

## Methods

### getAsText

```
public String getAsText()
```

### getCustomEditor

```
public java.awt.Component getCustomEditor()
```

### getJavaInitializationString

```
public String getJavaInitializationString()
```

Does nothing, and returns null.

### getTags

```
public String[] getTags()
```

Does nothing, and returns null.

### getValue

```
public Object getValue()
```

### isPaintable

```
public boolean isPaintable()
```

### paintValue

```
public void paintValue(java.awt.Graphics gfx,  
    java.awt.Rectangle box)
```

### setAsText

```
public void setAsText(String text)
```

### setValue

```
public void setValue(Object value)
```

---

## **supportsCustomEditor**

```
public boolean supportsCustomEditor()
```

## com.cafean.client.ui.beans

### Class BorderStyleEditor

```

java.lang.Object
  |
  +- java.beans.PropertyEditorSupport
      |
      +- com.cafean.client.ui.beans.BorderStyleEditor
  
```

#### All Implemented Interfaces:

java.awt.event.ItemListener, java.beans.PropertyEditor

public class **BorderStyleEditor**  
 extends java.beans.PropertyEditorSupport  
 implements java.beans.PropertyEditor, java.awt.event.ItemListener

Custom property editor to modify the component border properties.

### Constructor Summary

|        |                                     |
|--------|-------------------------------------|
| public | BorderStyleEditor()<br>Constructor. |
|--------|-------------------------------------|

### Method Summary

|                    |  |
|--------------------|--|
| void               | editorChangeValue(Border newBorder)<br>Update the user interface to reflect the edited border properties.                            |
| String             | getAsText()<br>The property value as a human editable string.  |
| int                | getBorderType(Border bord)<br>Determine the border type for a given border.  |
| java.awt.Component | getCustomEditor()<br>A component that will allow direct editing of the current property value.                                       |
| String             | getJavaInitializationString()<br>This method is intended for use when generating Java code to set the value of the property.         |
| String[]           | getTags()<br>If the property value must be one of a set of known tagged values, then this method should return an array of the tags. |
| Object             | getValue()<br>Access routine for the channel name.   |
| boolean            | isPaintable()<br>True if the class will honor the paintValue method.   |

|         |  |
|---------|--|
| void    | <code>itemStateChanged( java.awt.event.ItemEvent evt )</code><br>Invoked when the combobox's state has been changed.                                     |
| void    | <code>paintValue( java.awt.Graphics gfx, java.awt.Rectangle box )</code><br>Paint a representation of the value into a given area of screen real estate. |
| void    | <code>setAsText( String s )</code><br>Set the property value by parsing a given String.  |
| void    | <code>setValue( Object o )</code><br>Set (or change) the object that is to be edited.  |
| boolean | <code>supportsCustomEditor( )</code><br>True if the <code>propertyEditor</code> can provide a custom editor.   |

#### Methods inherited from class `java.beans.PropertyEditorSupport`

`addPropertyChangeListener, firePropertyChange, getAsText, getCustomEditor, getJavaInitializationString, getSource, getTags, getValue, isPaintable, paintValue, removePropertyChangeListener, setAsText, setSource, setValue, supportsCustomEditor`

#### Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructors

### BorderStyleEditor

`public BorderStyleEditor( )`

Constructor.

## Methods

### itemStateChanged

`public void itemStateChanged( java.awt.event.ItemEvent evt )`

Invoked when the combobox's state has been changed.

#### Parameters:

`evt` - The event object.

### setValue

`public void setValue( Object o )`

Set (or change) the object that is to be edited.

#### Parameters:

`o` - The object to edit.

(continued from last page)

---

## editorChangeValue

```
public void editorChangeValue(Border newBorder)
```

Update the user interface to reflect the edited border properties.

**Parameters:**

`newBorder` - The border to be used to update the user interface.

---

## getBorderType

```
public int getBorderType(Border bord)
```

Determine the border type for a given border.

**Parameters:**

`bord` - The border to evaluate.

**Returns:**

The border type. 0=none, 1=Raised Bevel. 2=lowered bevel, 3=etched, 4=line, 5=empty.

---

## getValue

```
public Object getValue()
```

Access routine for the channel name.

---

## setAsText

```
public void setAsText(String s)  
throws IllegalArgumentException
```

Set the property value by parsing a given String.

**Parameters:**

`s` - The string to be parsed.

---

## getJavaInitializationString

```
public String getJavaInitializationString()
```

This method is intended for use when generating Java code to set the value of the property.

**Returns:**

A fragment of Java code that can be used to initialize a variable with the current property value.

---

## isPaintable

```
public boolean isPaintable()
```

True if the class will honor the `paintValue` method.

**Returns:**

false

---

## paintValue

```
public void paintValue(java.awt.Graphics gfx,  
    java.awt.Rectangle box)
```

Paint a representation of the value into a given area of screen real estate. Not used here.

---

(continued from last page)

## getAsText

```
public String getAsText()
```

The property value as a human editable string.

---

## getTags

```
public String[] getTags()
```

If the property value must be one of a set of known tagged values, then this method should return an array of the tags. Not used here.

---

## getCustomEditor

```
public java.awt.Component getCustomEditor()
```

A component that will allow direct editing of the current property value.

**Returns:**

this.

---

## supportsCustomEditor

```
public boolean supportsCustomEditor()
```

True if the propertyEditor can provide a custom editor.

**Returns:**

true.

---

## com.cafean.client.ui.beans

# Class ComponentSelectionEditor

```

java.lang.Object
  |
  +- java.beans.PropertyEditorSupport
        |
        +- com.cafean.client.ui.beans.ComponentSelectionEditor
  
```

### All Implemented Interfaces:

[ModelDependent](#), java.beans.PropertyEditor

public class **ComponentSelectionEditor**  
 extends java.beans.PropertyEditorSupport  
 implements java.beans.PropertyEditor, [ModelDependent](#)

This bean editor contains a JPanel with a label and a button. This allows the user to select a component from the current model. Plugin specific extensions can be created that initialize the category value, which will limit the selection scope by that category.

## Constructor Summary

|        |   |
|--------|---|
| public | ComponentSelectionEditor(Category category,boolean allowEmpty)<br>Creates a new instance of CellComponentSelector |
| public | ComponentSelectionEditor()<br>Creates a new instance of CellComponentSelector                                     |

## Method Summary

|                               |  |
|-------------------------------|--|
| String                        | getAsText()<br>  |
| <a href="#">Category</a>      | getCategory()<br>Gets the Category for this ComponentSelectionEditor.              |
| java.awt.Component            | getCustomEditor()<br>  |
| String                        | getJavaInitializationString()<br>Does nothing, and returns null.                   |
| <a href="#">AbstractModel</a> | getModel()<br>   |
| TableCellEditor               | getTableCellEditor()<br>Method allows this panel to be used in JTable as an editor |
| String[]                      | getTags()<br>Does nothing, and returns null.                                       |
| Object                        | getValue()<br>   |

|         |  |
|---------|--|
| boolean | isCancelled()  |
| boolean | isPaintable()  |
| void    | paintValue(java.awt.Graphics gfx, java.awt.Rectangle box)  |
| void    | setAsText(String text)   |
| void    | setCategory(Category category)<br>Sets the Category for this ComponentSelectionEditor.   |
| void    | setCreator(ComponentCreator creator)<br>This sets the ComponentCreator that can be used to create new instances of the AbstractComponent this selector is supposed to select between |
| void    | setEditable(boolean e)   |
| void    | setFont(java.awt.Font font)<br>Sets the font on the label for this ComponentSelectionEditor.   |
| void    | setModel(AbstractModel model)  |
| void    | setValue(Object value)   |
| boolean | supportsCustomEditor()   |

#### Methods inherited from class java.beans.PropertyEditorSupport

addPropertyChangeListener, firePropertyChange, getAsText, getCustomEditor, getJavaInitializationString, getSource, getTags, getValue, isPaintable, paintValue, removePropertyChangeListener, setAsText, setSource, setValue, supportsCustomEditor

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### ComponentSelectionEditor

```
public ComponentSelectionEditor(Category category,
                               boolean allowEmpty)
```

Creates a new instance of CellComponentSelector

### ComponentSelectionEditor

```
public ComponentSelectionEditor()
```

Creates a new instance of CellComponentSelector



(continued from last page)

## Methods

### **isCancelled**

```
public boolean isCancelled()
```

### **setEditable**

```
public void setEditable(boolean e)
```

### **getAsText**

```
public String getAsText()
```

### **getCustomEditor**

```
public java.awt.Component getCustomEditor()
```

### **getJavaInitializationString**

```
public String getJavaInitializationString()
```

Does nothing, and returns null.

### **getTags**

```
public String[] getTags()
```

Does nothing, and returns null.

### **getValue**

```
public Object getValue()
```

### **isPaintable**

```
public boolean isPaintable()
```

### **setFont**

```
public void setFont(java.awt.Font font)
```

Sets the font on the label for this ComponentSelectionEditor.

### **setCreator**

```
public void setCreator(ComponentCreator creator)
```

This sets the ComponentCreator that can be used to create new instances of the AbstractComponent this selector is supposed to select between

---

## paintValue

```
public void paintValue(java.awt.Graphics gfx,  
                        java.awt.Rectangle box)
```

---

## setAsText

```
public void setAsText(String text)  
    throws IllegalArgumentException
```

---

## setValue

```
public void setValue(Object value)
```

---

## supportsCustomEditor

```
public boolean supportsCustomEditor()
```

---

## getCategory

```
public Category getCategory()  
    Gets the Category for this ComponentSelectionEditor.
```

**Returns:**  
the Category of the component reference.

---

## setCategory

```
public void setCategory(Category category)  
    Sets the Category for this ComponentSelectionEditor.
```

**Parameters:**  
category - the Category of the component reference.

---

## getModel

```
public AbstractModel getModel()
```

---

## setModel

```
public void setModel(AbstractModel model)
```

---

## getTableCellEditor

```
public TableCellEditor getTableCellEditor()  
    Method allows this panel to be used in JTable as an editor
```

**Returns:**

---

(continued from last page)

TableCellEditor value to be used to edit JTable cells;

## com.cafean.client.ui.beans Interface IntrospectingEditor

### All Known Implementing Classes:

RealArrayEditor, RealBeanEditor

public interface **IntrospectingEditor**  
extends java.beans.PropertyEditor

An interface for PropertyEditors that use introspection to initialize themselves and to create new instances of the property they edit.

This is a functional equivalent interface to the SwingEditorSupport base class.

### See Also:

javax.swing.beaninfo.SwingEditorSupport

## Method Summary

|      |  |
|------|--|
| void | <code>init(java.beans.PropertyDescriptor descriptor)</code><br>For property editors that must be initialized with values from the property descriptor. |
|------|--|

### Methods inherited from interface java.beans.PropertyEditor

`addPropertyChangeListener`, `getAsText`, `getCustomEditor`, `getJavaInitializationString`, `getTags`, `getValue`, `isPaintable`, `paintValue`, `removePropertyChangeListener`, `setAsText`, `setValue`, `supportsCustomEditor`

## Methods

### **init**

public void **init**(java.beans.PropertyDescriptor descriptor)

For property editors that must be initialized with values from the property descriptor.

## com.cafean.client.ui.beans Class NamedIntEditor

```
java.lang.Object
|
|-- java.beans.PropertyEditorSupport
|   |-- com.cafean.client.ui.beans.NamedIntEditor
```

### All Implemented Interfaces:

java.awt.event.ItemListener, java.beans.PropertyEditor

### Direct Known Subclasses:

[NamelistNamedIntEditor](#)

### public class **NamedIntEditor**

extends java.beans.PropertyEditorSupport

implements java.beans.PropertyEditor, java.awt.event.ItemListener

This bean editor contains a JComboBox for selecting between named integers.

### Example:

```
public class OnOffSelEditor
    extends NamedIntEditor
{
    // the option values
    private static final int[] optVal = {0, 1};
    // the option descriptions
    private static final String[] optStr = {"Off", "On"};
    // Creates a new instance of OnOffSelEditor
    public OnOffSelEditor() {
        super( optStr, optVal );
        setShowNumbers(false);
    }
}
```

## Field Summary

|                  |                                       |
|------------------|---------------------------------------|
| static final int | DIFF_VAL<br>Value: <b>-2147483647</b> |
| static final int | INACTIVE<br>Value: <b>-2147483648</b> |
| static final int | UNKNOWN<br>Value: <b>2147483647</b>   |

## Constructor Summary

|        |  |
|--------|--|
| public | NamedIntEditor()<br>Creates a new instance of NamedIntEditor |
|--------|--|

|        |  |
|--------|--|
| public | NamedIntEditor(String[] strings,int[] numbers)<br><br>Creates a new instance of NamedIntEditor for use in editing the given set of enumerated integers and their associated string descriptions. |
|--------|--|

## Method Summary

|                    |  |
|--------------------|--|
| String             | getAsText()  |
| java.awt.Component | getCustomEditor()  |
| String             | getJavaInitializationString()<br>{ @inheritDoc } Does nothing, and returns null.   |
| String[]           | getTags()<br>{ @inheritDoc } Does nothing, and returns null.   |
| Object             | getValue()   |
| boolean            | isPaintable()  |
| boolean            | isShowNumbers()<br>Getter for property showNumbers.  |
| void               | itemStateChanged(java.awt.event.ItemEvent e)   |
| void               | paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)   |
| void               | setAsText(String text)   |
| void               | setContext(PropertyController owner,java.beans.PropertyDescriptor descriptor)<br>Sets the owner and the description of the property currently being edited by this editor. |
| void               | setShowNumbers(boolean showNumbers)<br>Setter for property showNumbers.  |
| void               | setValue(Object value)   |
| boolean            | supportsCustomEditor()   |

### Methods inherited from class java.beans.PropertyEditorSupport

addPropertyChangeListener, firePropertyChange, getAsText, getCustomEditor, getJavaInitializationString, getSource, getTags, getValue, isPaintable, paintValue, removePropertyChangeListener, setAsText, setSource, setValue, supportsCustomEditor

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### DIFF\_VAL

```
public static final int DIFF_VAL
```

### UNKNOWN

```
public static final int UNKNOWN
```

### INACTIVE

```
public static final int INACTIVE
```

## Constructors

### NamedIntEditor

```
public NamedIntEditor()  
    Creates a new instance of NamedIntEditor
```

### NamedIntEditor

```
public NamedIntEditor(String[] strings,  
                      int[] numbers)  
    Creates a new instance of NamedIntEditor for use in editing the given set of enumerated integers and their associated string descriptions.
```

## Methods

### setContext

```
public void setContext(PropertyController owner,  
                     java.beans.PropertyDescriptor descriptor)  
    Sets the owner and the description of the property currently being edited by this editor. Extensions of this editor that require the namelist functionality must implement NamelistEditor.
```

#### Parameters:

`owner` - the `PropertyController` that owns the property being edited by this editor. This may be null.  
`descriptor` - a `PropertyDescriptor` describing the property being edited by this editor.

### itemStateChanged

```
public void itemStateChanged(java.awt.event.ItemEvent e)
```

### getAsText

```
public String getAsText()
```

---

## getCustomEditor

```
public java.awt.Component getCustomEditor()
```

---

## getJavaInitializationString

```
public String getJavaInitializationString()
```

Does nothing, and returns null.

---

## getTags

```
public String[] getTags()
```

Does nothing, and returns null.

---

## getValue

```
public Object getValue()
```

---

## isPaintable

```
public boolean isPaintable()
```

---

## paintValue

```
public void paintValue(java.awt.Graphics gfx,  
    java.awt.Rectangle box)
```

---

## setAsText

```
public void setAsText(String text)  
    throws IllegalArgumentException
```

---

## setValue

```
public void setValue(Object value)
```

---

## supportsCustomEditor

```
public boolean supportsCustomEditor()
```

---

## isShowNumbers

```
public boolean isShowNumbers()
```

Getter for property showNumbers.

**Returns:**



(continued from last page)

Value of property showNumbers.

---

## **setShowNumbers**

```
public void setShowNumbers(boolean showNumbers)
```

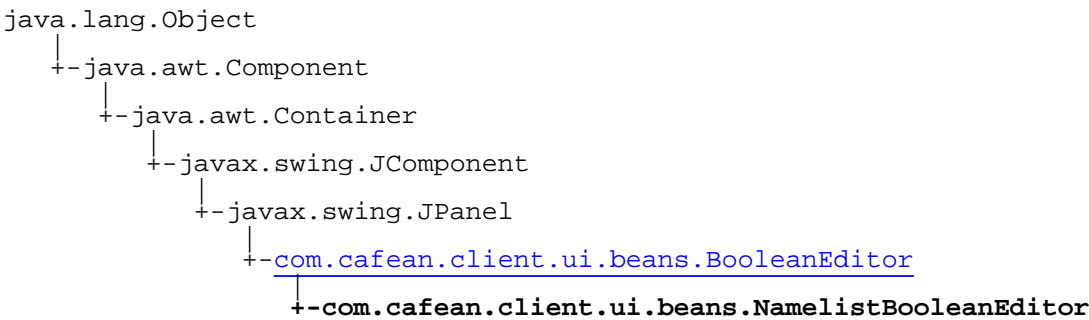
Setter for property showNumbers.

### **Parameters:**

showNumbers - New value of property showNumbers.

# com.cafean.client.ui.beans

## Class NamelistBooleanEditor



**All Implemented Interfaces:**  
java.awt.event.ItemListener, [NamelistEditor](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, [java.io.Serializable](#), javax.accessibility.Accessible, java.beans.PropertyEditor

public class **NamelistBooleanEditor**  
extends [BooleanEditor](#)  
implements java.beans.PropertyEditor, javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, [NamelistEditor](#), java.awt.event.ItemListener

This is an editor for a namelist style boolean value that can be activated and deactivated.

| Fields inherited from class javax.swing.JComponent |   |
|--|---|
| TOOL_TIP_TEXT_KEY                                  | UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW |

| Fields inherited from class java.awt.Component |  |
|--|--|
| BOTTOM_ALIGNMENT                               | CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT |

| Constructor Summary |                         |   |
|---------------------|-------------------------|---|
| public              | NamelistBooleanEditor() | Creates a new instance of NamelistBooleanEditor |

| Method Summary |   |  |
|----------------|---|--|
| void           | firePropertyChange()                          | Report that we have been modified to any interested listeners. |
| void           | itemStateChanged( java.awt.event.ItemEvent e) |  |

|      |   |
|------|---|
| void | setContext(PropertyController owner, java.beans.PropertyDescriptor descriptor)<br><br>Sets the owner and the description of the property currently being edited by this editor. |
| void | setValue(Object value)  |

**Methods inherited from class [com.cafean.client.ui.beans.BooleanEditor](#)**

[getAsText](#), [getCustomEditor](#), [getJavaInitializationString](#), [getTags](#), [getValue](#), [isPaintable](#), [paintValue](#), [setAsText](#), [setValue](#), [supportsCustomEditor](#)

**Methods inherited from class [javax.swing.JPanel](#)**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class [javax.swing.JComponent](#)**

[addAncestorListener](#), [addNotify](#), [addVetoableChangeListener](#), [computeVisibleRect](#), [contains](#), [createToolTip](#), [disable](#), [enable](#), [firePropertyChange](#), [firePropertyChange](#), [firePropertyChange](#), [getAccessibleContext](#), [getActionForKeyStroke](#), [getActionMap](#), [getAlignmentX](#), [getAlignmentY](#), [getAncestorListeners](#), [getAutoscrolls](#), [getBorder](#), [getBounds](#), [getClientProperty](#), [getComponentPopupMenu](#), [getConditionForKeyStroke](#), [getDebugGraphicsOptions](#), [getDefaultLocale](#), [getFontMetrics](#), [getGraphics](#), [getHeight](#), [getInheritsPopupMenu](#), [getInputMap](#), [getInputMap](#), [getInputVerifier](#), [getInsets](#), [getInsets](#), [getListeners](#), [getLocation](#), [getMaximumSize](#), [getMinimumSize](#), [getNextFocusableComponent](#), [getPopupLocation](#), [getPreferredSize](#), [getRegisteredKeyStrokes](#), [getRootPane](#), [getSize](#), [getToolTipLocation](#), [getToolTipText](#), [getToolTipText](#), [getTopLevelAncestor](#), [getTransferHandler](#), [getUIClassID](#), [getVerifyInputWhenFocusTarget](#), [getVetoableChangeListeners](#), [getVisibleRect](#), [getWidth](#), [getX](#), [getY](#), [grabFocus](#), [isDoubleBuffered](#), [isLightweightComponent](#), [isManagingFocus](#), [isOpaque](#), [isOptimizedDrawingEnabled](#), [isPaintingTile](#), [isRequestFocusEnabled](#), [isValidateRoot](#), [paint](#), [paintImmediately](#), [paintImmediately](#), [print](#), [printAll](#), [putClientProperty](#), [registerKeyboardAction](#), [registerKeyboardAction](#), [removeAncestorListener](#), [removeNotify](#), [removeVetoableChangeListener](#), [repaint](#), [repaint](#), [requestDefaultFocus](#), [requestFocus](#), [requestFocus](#), [requestFocusInWindow](#), [resetKeyboardActions](#), [reshape](#), [revalidate](#), [scrollRectToVisible](#), [setActionMap](#), [setAlignmentX](#), [setAlignmentY](#), [setAutoscrolls](#), [setBackground](#), [setBorder](#), [setComponentPopupMenu](#), [setDebugGraphicsOptions](#), [setDefaultLocale](#), [setDoubleBuffered](#), [setEnabled](#), [setFocusTraversalKeys](#), [setFont](#), [setForeground](#), [setInheritsPopupMenu](#), [setInputMap](#), [setInputVerifier](#), [setMaximumSize](#), [setMinimumSize](#), [setNextFocusableComponent](#), [setOpaque](#), [setPreferredSize](#), [setRequestFocusEnabled](#), [setToolTipText](#), [setTransferHandler](#), [setVerifyInputWhenFocusTarget](#), [setVisible](#), [unregisterKeyboardAction](#), [update](#), [updateUI](#)

**Methods inherited from class [java.awt.Container](#)**

[add](#), [add](#), [add](#), [add](#), [add](#), [addContainerListener](#), [addNotify](#), [addPropertyChangeListener](#), [addPropertyChangeListener](#), [applyComponentOrientation](#), [areFocusTraversalKeysSet](#), [countComponents](#), [deliverEvent](#), [doLayout](#), [findComponentAt](#), [findComponentAt](#), [getAlignmentX](#), [getAlignmentY](#), [getComponent](#), [getComponentAt](#), [getComponentAt](#), [getComponentCount](#), [getComponents](#), [getComponentZOrder](#), [getContainerListeners](#), [getFocusTraversalKeys](#), [getFocusTraversalPolicy](#), [getInsets](#), [getLayout](#), [getListeners](#), [getMaximumSize](#), [getMinimumSize](#), [getMousePosition](#), [getPreferredSize](#), [insets](#), [invalidate](#), [isAncestorOf](#), [isFocusCycleRoot](#), [isFocusCycleRoot](#), [isFocusTraversalPolicyProvider](#), [isFocusTraversalPolicySet](#), [layout](#), [list](#), [list](#), [locate](#), [minimumSize](#), [paint](#), [paintComponents](#), [preferredSize](#), [print](#), [printComponents](#), [remove](#), [remove](#), [removeAll](#), [removeContainerListener](#), [removeNotify](#), [setComponentZOrder](#), [setFocusCycleRoot](#), [setFocusTraversalKeys](#), [setFocusTraversalPolicy](#), [setFocusTraversalPolicyProvider](#), [setFont](#), [setLayout](#), [transferFocusBackward](#), [transferFocusDownCycle](#), [update](#), [validate](#)

**Methods inherited from class [java.awt.Component](#)**

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### NamelistBooleanEditor

```
public NamelistBooleanEditor()
```

Creates a new instance of NamelistBooleanEditor

## Methods

### setValue

```
public void setValue(Object value)
```

---

## setContext

```
public void setContext(PropertyController owner,  
    java.beans.PropertyDescriptor descriptor)
```

Sets the owner and the description of the property currently being edited by this editor.

### Parameters:

owner - the PropertyController that owns the property being edited by this editor. This may be null.

descriptor - a PropertyDescriptor describing the property being edited by this editor.

---

## itemStateChanged

```
public void itemStateChanged(java.awt.event.ItemEvent e)
```

---

## firePropertyChange

```
public void firePropertyChange()
```

Report that we have been modified to any interested listeners.

## com.cafean.client.ui.beans Class NamelistDoubleEditor

```

java.lang.Object
  |
  +- java.beans.PropertyEditorSupport
      |
      +- javax.swing.beaninfo.SwingEditorSupport
          |
          +- com.cafean.client.ui.beans.NamelistDoubleEditor
  
```

### All Implemented Interfaces:

java.awt.event.ItemListener, [NamelistEditor](#), java.beans.PropertyEditor

```

public class NamelistDoubleEditor
  extends SwingEditorSupport
  implements java.beans.PropertyEditor, NamelistEditor, java.awt.event.ItemListener
  
```

This is an editor for a namelist style double value that can be activated and deactivated.

### See Also:

NamelistEditor

## Constructor Summary

|        |   |
|--------|---|
| public | NamelistDoubleEditor()<br>Creates a new NamelistIntEditor |
|--------|---|

## Method Summary

|      |   |
|------|---|
| void | itemStateChanged(java.awt.event.ItemEvent e)  |
| void | setContext(PropertyController owner, java.beans.PropertyDescriptor descriptor)<br>Sets the owner and the description of the property currently being edited by this editor. |
| void | setValue(Object value)  |

### Methods inherited from class javax.swing.beaninfo.SwingEditorSupport

getCustomEditor, init, supportsCustomEditor

### Methods inherited from class java.beans.PropertyEditorSupport

addPropertyChangeListener, firePropertyChange, getAsText, getCustomEditor, getJavaInitializationString, getSource, getTags, getValue, isPaintable, paintValue, removePropertyChangeListener, setAsText, setSource, setValue, supportsCustomEditor

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### NamelistDoubleEditor

```
public NamelistDoubleEditor()
```

Creates a new NamelistIntEditor

## Methods

### setValue

```
public void setValue(Object value)
```

---

### setContext

```
public void setContext(PropertyController owner,  
    java.beans.PropertyDescriptor descriptor)
```

Sets the owner and the description of the property currently being edited by this editor.

#### Parameters:

`owner` - the PropertyController that owns the property being edited by this editor. This may be null.  
`descriptor` - a PropertyDescriptor describing the property being edited by this editor.

---

### itemStateChanged

```
public void itemStateChanged(java.awt.event.ItemEvent e)
```

## com.cafean.client.ui.beans Interface NamelistEditor

### All Known Implementing Classes:

NamelistBooleanEditor, NamelistRealEditor, NamelistIntEditor, NamelistDoubleEditor, NamelistNamedIntEditor

### public interface **NamelistEditor**

An interface describing an editor for a property that conforms to the namelist variable concept in which a property is actually a combination of a property and a boolean activation state.

Editors of this type assume that the following method is implemented in the object being edited:

```
void setPropertyActive( String propertyName )
```

### See Also:

PropertyController#isPropertyActive

## Method Summary

|      |   |
|------|---|
| void | setContext(PropertyController owner, java.beans.PropertyDescriptor descriptor)<br>Sets the owner and the description of the property currently being edited by this editor. |
|------|---|

## Methods

### setContext

```
public void setContext(PropertyController owner,  
    java.beans.PropertyDescriptor descriptor)
```

Sets the owner and the description of the property currently being edited by this editor.

#### Parameters:

`owner` - the PropertyController that owns the property being edited by this editor. This may be null.  
`descriptor` - a PropertyDescriptor describing the property being edited by this editor.



## com.cafean.client.ui.beans

### Class NamelistIntEditor

```

java.lang.Object
  |
  +- java.beans.PropertyEditorSupport
        |
        +- javax.swing.beaninfo.SwingEditorSupport
              |
              +- com.cafean.client.ui.beans.NamelistIntEditor
  
```

#### All Implemented Interfaces:

java.awt.event.ItemListener, [NamelistEditor](#), java.beans.PropertyEditor

```

public class NamelistIntEditor
  extends SwingEditorSupport
  implements java.beans.PropertyEditor, NamelistEditor, java.awt.event.ItemListener
  
```

This is an editor for a namelist style integer value that can be activated and deactivated.

#### See Also:

NamelistEditor

## Constructor Summary

|        |  |
|--------|--|
| public | NamelistIntEditor()<br>Creates a new NamelistIntEditor |
|--------|--|

## Method Summary

|      |   |
|------|---|
| void | itemStateChanged(java.awt.event.ItemEvent e)  |
| void | setContext(PropertyController owner, java.beans.PropertyDescriptor descriptor)<br>Sets the owner and the description of the property currently being edited by this editor. |
| void | setValue(Object value)  |

#### Methods inherited from class javax.swing.beaninfo.SwingEditorSupport

getCustomEditor, init, supportsCustomEditor

#### Methods inherited from class java.beans.PropertyEditorSupport

addPropertyChangeListener, firePropertyChange, getAsText, getCustomEditor, getJavaInitializationString, getSource, getTags, getValue, isPaintable, paintValue, removePropertyChangeListener, setAsText, setSource, setValue, supportsCustomEditor

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### NamelistIntEditor

```
public NamelistIntEditor()
```

Creates a new NamelistIntEditor

## Methods

### setValue

```
public void setValue(Object value)
```

---

### setContext

```
public void setContext(PropertyController owner,  
    java.beans.PropertyDescriptor descriptor)
```

Sets the owner and the description of the property currently being edited by this editor.

#### Parameters:

**owner** - the PropertyController that owns the property being edited by this editor. This may be null.  
**descriptor** - a PropertyDescriptor describing the property being edited by this editor.

---

### itemStateChanged

```
public void itemStateChanged(java.awt.event.ItemEvent e)
```

## com.cafean.client.ui.beans Class NamelistNamedIntEditor

```

java.lang.Object
  |
  +- java.beans.PropertyEditorSupport
        |
        +- com.cafean.client.ui.beans.NamedIntEditor
              |
              +- com.cafean.client.ui.beans.NamelistNamedIntEditor

```

### All Implemented Interfaces:

[NamelistEditor](#), [java.beans.PropertyEditor](#), [java.awt.event.ItemListener](#)

public class **NamelistNamedIntEditor**  
 extends [NamedIntEditor](#)  
 implements [java.awt.event.ItemListener](#), [java.beans.PropertyEditor](#), [NamelistEditor](#)

An enumeration editor for values that also include an *Activestate* as described in [NamelistEditor](#).

### Fields inherited from class [com.cafean.client.ui.beans.NamedIntEditor](#)

[DIFF\\_VAL](#), [INACTIVE](#), [UNKNOWN](#)

## Constructor Summary

|        |  |
|--------|--|
| public | NamelistNamedIntEditor(String[] strings,int[] numbers)<br>Creates a new instance of NamelistNamedIntEditor |
|--------|--|

### Methods inherited from class [com.cafean.client.ui.beans.NamedIntEditor](#)

[getAsText](#), [getCustomEditor](#), [getJavaInitializationString](#), [getTags](#), [getValue](#), [isPaintable](#), [isShowNumbers](#), [itemStateChanged](#), [paintValue](#), [setAsText](#), [setContext](#), [setShowNumbers](#), [setValue](#), [supportsCustomEditor](#)

### Methods inherited from class [java.beans.PropertyEditorSupport](#)

[addPropertyChangeListener](#), [firePropertyChange](#), [getAsText](#), [getCustomEditor](#), [getJavaInitializationString](#), [getSource](#), [getTags](#), [getValue](#), [isPaintable](#), [paintValue](#), [removePropertyChangeListener](#), [setAsText](#), [setSource](#), [setValue](#), [supportsCustomEditor](#)

### Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

(continued from last page)

## NamelistNamedIntEditor

```
public NamelistNamedIntEditor(String[] strings,  
                             int[] numbers)
```

Creates a new instance of NamelistNamedIntEditor

## com.cafean.client.ui.beans Class NamelistRealEditor

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JPanel
          |-- com.cafean.client.ui.beans.RealBeanEditor
            |-- com.cafean.client.ui.beans.NamelistRealEditor

```

### All Implemented Interfaces:

java.awt.event.ItemListener, [NamelistEditor](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, [ModelDependent](#), [IntrospectingEditor](#)

### public class **NamelistRealEditor**

extends [RealBeanEditor](#)

implements [IntrospectingEditor](#), [ModelDependent](#), javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, [NamelistEditor](#), java.awt.event.ItemListener

An editor for values of type Real that also include an *Activestate* as described in `NamelistEditor`.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |  |
|--------|--|
| public | <code>NamelistRealEditor()</code><br>Creates a new instance of DefaultableRealEditor |
|--------|--|

## Method Summary

|      |  |
|------|--|
| void | <code>itemStateChanged(java.awt.event.ItemEvent e)</code>  |
| void | <code>setContext(PropertyController owner, java.beans.PropertyDescriptor descriptor)</code><br>Sets the owner and the description of the property currently being edited by this editor. |

|      |                        |
|------|------------------------|
| void | setValue(Object value) |
|------|------------------------|

**Methods inherited from class [com.cafean.client.ui.beans.RealBeanEditor](#)**

[firePropertyChange](#), [getAsText](#), [getCustomEditor](#), [getJavaInitializationString](#), [getModel](#), [getTags](#), [getValue](#), [init](#), [isPaintable](#), [paintValue](#), [setAsText](#), [setForeground](#), [setModel](#), [setValue](#), [supportsCustomEditor](#)

**Methods inherited from class [javax.swing.JPanel](#)**

[getAccessibleContext](#), [getUI](#), [getUIClassID](#), [setUI](#), [updateUI](#)

**Methods inherited from class [javax.swing.JComponent](#)**

[addAncestorListener](#), [addNotify](#), [addVetoableChangeListener](#), [computeVisibleRect](#), [contains](#), [createToolTip](#), [disable](#), [enable](#), [firePropertyChange](#), [firePropertyChange](#), [firePropertyChange](#), [getAccessibleContext](#), [getActionForKeyStroke](#), [getActionMap](#), [getAlignmentX](#), [getAlignmentY](#), [getAncestorListeners](#), [getAutoscrolls](#), [getBorder](#), [getBounds](#), [getClientProperty](#), [getComponentPopupMenu](#), [getConditionForKeyStroke](#), [getDebugGraphicsOptions](#), [getDefaultLocale](#), [getFontMetrics](#), [getGraphics](#), [getHeight](#), [getInheritsPopupMenu](#), [getInputMap](#), [getInputMap](#), [getInputVerifier](#), [getInsets](#), [getInsets](#), [getListeners](#), [getLocation](#), [getMaximumSize](#), [getMinimumSize](#), [getNextFocusableComponent](#), [getPopupLocation](#), [getPreferredSize](#), [getRegisteredKeyStrokes](#), [getRootPane](#), [getSize](#), [getToolTipLocation](#), [getToolTipText](#), [getToolTipText](#), [getTopLevelAncestor](#), [getTransferHandler](#), [getUIClassID](#), [getVerifyInputWhenFocusTarget](#), [getVetoableChangeListeners](#), [getVisibleRect](#), [getWidth](#), [getX](#), [getY](#), [grabFocus](#), [isDoubleBuffered](#), [isLightweightComponent](#), [isManagingFocus](#), [isOpaque](#), [isOptimizedDrawingEnabled](#), [isPaintingTile](#), [isRequestFocusEnabled](#), [isValidateRoot](#), [paint](#), [paintImmediately](#), [paintImmediately](#), [print](#), [printAll](#), [putClientProperty](#), [registerKeyboardAction](#), [registerKeyboardAction](#), [removeAncestorListener](#), [removeNotify](#), [removeVetoableChangeListener](#), [repaint](#), [repaint](#), [requestDefaultFocus](#), [requestFocus](#), [requestFocus](#), [requestFocusInWindow](#), [resetKeyboardActions](#), [reshape](#), [revalidate](#), [scrollRectToVisible](#), [setActionMap](#), [setAlignmentX](#), [setAlignmentY](#), [setAutoscrolls](#), [setBackground](#), [setBorder](#), [setComponentPopupMenu](#), [setDebugGraphicsOptions](#), [setDefaultLocale](#), [setDoubleBuffered](#), [setEnabled](#), [setFocusTraversalKeys](#), [setFont](#), [setForeground](#), [setInheritsPopupMenu](#), [setInputMap](#), [setInputVerifier](#), [setMaximumSize](#), [setMinimumSize](#), [setNextFocusableComponent](#), [setOpaque](#), [setPreferredSize](#), [setRequestFocusEnabled](#), [setToolTipText](#), [setTransferHandler](#), [setVerifyInputWhenFocusTarget](#), [setVisible](#), [unregisterKeyboardAction](#), [update](#), [updateUI](#)

**Methods inherited from class [java.awt.Container](#)**

[add](#), [add](#), [add](#), [add](#), [add](#), [addContainerListener](#), [addNotify](#), [addPropertyChangeListener](#), [addPropertyChangeListener](#), [applyComponentOrientation](#), [areFocusTraversalKeysSet](#), [countComponents](#), [deliverEvent](#), [doLayout](#), [findComponentAt](#), [findComponentAt](#), [getAlignmentX](#), [getAlignmentY](#), [getComponent](#), [getComponentAt](#), [getComponentAt](#), [getComponentCount](#), [getComponents](#), [getComponentZOrder](#), [getContainerListeners](#), [getFocusTraversalKeys](#), [getFocusTraversalPolicy](#), [getInsets](#), [getLayout](#), [getListeners](#), [getMaximumSize](#), [getMinimumSize](#), [getMousePosition](#), [getPreferredSize](#), [insets](#), [invalidate](#), [isAncestorOf](#), [isFocusCycleRoot](#), [isFocusCycleRoot](#), [isFocusTraversalPolicyProvider](#), [isFocusTraversalPolicySet](#), [layout](#), [list](#), [list](#), [locate](#), [minimumSize](#), [paint](#), [paintComponents](#), [preferredSize](#), [print](#), [printComponents](#), [remove](#), [remove](#), [removeAll](#), [removeContainerListener](#), [removeNotify](#), [setComponentZOrder](#), [setFocusCycleRoot](#), [setFocusTraversalKeys](#), [setFocusTraversalPolicy](#), [setFocusTraversalPolicyProvider](#), [setFont](#), [setLayout](#), [transferFocusBackward](#), [transferFocusDownCycle](#), [update](#), [validate](#)

**Methods inherited from class [java.awt.Component](#)**

```

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
bounds, checkImage, checkImage, contains, contains, createImage, createImage,
createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent,
doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds,
getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners,
getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor,
getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners,
getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize,
getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize,
getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit,
getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list,
list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent,
preferredSize, prepareImage, prepareImage, print, printAll, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize,
setBackground, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys,
setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale,
setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setSize, setSize, setVisible, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle, update, validate

```

#### Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

## Constructors

### NamelistRealEditor

```
public NamelistRealEditor()
```

Creates a new instance of DefaultableRealEditor

## Methods

### setValue

```
public void setValue(Object value)
```

---

## setContext

```
public void setContext(PropertyController owner,  
    java.beans.PropertyDescriptor descriptor)
```

Sets the owner and the description of the property currently being edited by this editor.

### Parameters:

- owner - the PropertyController that owns the property being edited by this editor. This may be null.
  - descriptor - a PropertyDescriptor describing the property being edited by this editor.
- 

## itemStateChanged

```
public void itemStateChanged(java.awt.event.ItemEvent e)
```



## com.cafean.client.ui.beans Interface PropertyController

**All Known Implementing Classes:**  
ViewComponent

public interface **PropertyController**

An interface describing an object that has methods to determine if one of its properties is enabled or disabled, and optional or required at the current moment.

This interface is used extensively in the Properties View to limit the properties displayed to those actually needed by the analyst.

### Field Summary

|  |   |
|--|---|
| <code>static final int</code>            | ALL<br>the group containing all properties, be they disabled required or optional.<br>Value: <b>7</b> |
| <code>static final java.awt.Color</code> | COLOR_OPTIONAL<br>the foreground color for optional properties  |
| <code>static final int</code>            | DISABLED<br>the group containing only disabled properties.<br>Value: <b>4</b>                         |
| <code>static final int</code>            | NONE<br>the group containing no properties.<br>Value: <b>0</b>  |
| <code>static final int</code>            | OPTIONAL<br>the group containing only optional properties.<br>Value: <b>2</b>                         |
| <code>static final int</code>            | REQUIRED<br>the group containing only required properties.<br>Value: <b>1</b>                         |

### Method Summary

|                      |  |
|----------------------|--|
| <code>int</code>     | <code>getAttributeIndex(String propertyName)</code><br>Returns a relative index that can be used to order property lists.  |
| <code>boolean</code> | <code>isPropertyActive(String propertyName)</code><br>Returns false if this object has a property with the given name that is considered inactive; otherwise true. |

|         |   |
|---------|---|
| boolean | <b>isEnabled(String propertyName)</b><br>Returns false if this object has a property with the given name that has dependency code that fails; true otherwise                    |
| boolean | <b>isRequired(String propertyName)</b><br>Returns false if this object has a property with the given name that has requirement code that fails; true otherwise.                 |
| boolean | <b>isResizable(String propertyName)</b><br>Returns false if this object has an array property with the given name that should not normally be resizable.                        |
| boolean | <b>isRestartEditable(String propertyName)</b><br>Returns true if this object has a property with the given name that should be editable during a restart edit; false otherwise. |
| boolean | <b>isRestartResizable(String propertyName)</b><br>Returns false if this object has an array property with the given name that should not be resizable while editing a restart.  |

## Fields

### COLOR\_OPTIONAL

```
public static final java.awt.Color COLOR_OPTIONAL
    the foreground color for optional properties
```

### NONE

```
public static final int NONE
    the group containing no properties.
```

### REQUIRED

```
public static final int REQUIRED
    the group containing only required properties.
```

### OPTIONAL

```
public static final int OPTIONAL
    the group containing only optional properties.
```

### DISABLED

```
public static final int DISABLED
    the group containing only disabled properties.
```

### ALL

```
public static final int ALL
    the group containing all properties, be they disabled required or optional.
```

(continued from last page)

## Methods

### isPropertyEnabled

```
public boolean isPropertyEnabled(String propertyName)
```

Returns false if this object has a property with the given name that has dependency code that fails; true otherwise

**Parameters:**

propertyName - a String containing the property name to check

### isPropertyRequired

```
public boolean isPropertyRequired(String propertyName)
```

Returns false if this object has a property with the given name that has requirement code that fails; true otherwise.

**Parameters:**

propertyName - a String containing the property name to check

### isPropertyRestartEditable

```
public boolean isPropertyRestartEditable(String propertyName)
```

Returns true if this object has a property with the given name that should be editable during a restart edit; false otherwise.

**Parameters:**

propertyName - a String containing the property name to check

### isPropertyActive

```
public boolean isPropertyActive(String propertyName)
```

Returns false if this object has a property with the given name that is considered inactive; otherwise true. A property's active state is separate from its enabled state to allow for a namelist-like property that is actually a boolean paired with a property of another type.

**Parameters:**

propertyName - a String containing the name of the property to check

**Returns:**

false if the property is inactive, true otherwise

**See Also:**

NamelistEditor

### getAttributeIndex

```
public int getAttributeIndex(String propertyName)
```

Returns a relative index that can be used to order property lists.

**Parameters:**

propertyName - a String containing the property name to check

### isPropertyResizable

```
public boolean isPropertyResizable(String propertyName)
```

Returns false if this object has an array property with the given name that should not normally be resizable.

**Parameters:**

(continued from last page)

propertyName - a String containing the property name to check

---

## **isRestartResizable**

public boolean **isRestartResizable**(String propertyName)

Returns false if this object has an array property with the given name that should not be resizable while editing a restart.

### **Parameters:**

propertyName - a String containing the property name to check

## com.cafean.client.ui.beans Class RealArrayEditor

```

java.lang.Object
  |
  +- java.beans.PropertyEditorSupport
        |
        +- com.cafean.client.ui.beans.RealArrayEditor
  
```

### All Implemented Interfaces:

[ModelDependent](#), [IntrospectingEditor](#), java.beans.PropertyEditor

public class **RealArrayEditor**  
 extends java.beans.PropertyEditorSupport  
 implements java.beans.PropertyEditor, [IntrospectingEditor](#), [ModelDependent](#)

This bean editor contains a label and a button for editing an array of Real values. This can be either an array of fixed size, or an array that can be redimensioned. The button launches a RealArrayDialog that actually performs the editing.

## Constructor Summary

|        |  |
|--------|--|
| public | RealArrayEditor(boolean fixedDimension,int dimension)<br>This creates a new instance of RealArrayEditor that has a fixed dimension |
| public | RealArrayEditor()<br>Creates a new instance of RealArrayEditor, and creates the action listener on the button.                     |

## Method Summary

|                               |   |
|-------------------------------|---|
| void                          | addPropertyChangeListener(java.beans.PropertyChangeListener listener) |
| String                        | getAsText()   |
| java.awt.Component            | getCustomEditor()   |
| String                        | getJavaInitializationString()<br>Does nothing, and returns null.      |
| <a href="#">AbstractModel</a> | getModel()  |
| String[]                      | getTags()<br>Does nothing, and returns null.                          |
| Object                        | getValue()  |
| void                          | init(java.beans.PropertyDescriptor descriptor)                        |
| boolean                       | isPaintable()   |

|         |   |
|---------|---|
| void    | <code>paintValue(java.awt.Graphics gfx, java.awt.Rectangle box)</code>                |
| void    | <code>removePropertyChangeListener(java.beans.PropertyChangeListener listener)</code> |
| void    | <code>setAsText(String text)</code>   |
| void    | <code>setDimension(int dimension)</code><br>Sets the dimension on a given array.      |
| void    | <code>setModel(AbstractModel model)</code>  |
| void    | <code>setValue(Object value)</code><br>If this array is of fixed width.               |
| boolean | <code>supportsCustomEditor()</code>   |

#### Methods inherited from class `java.beans.PropertyEditorSupport`

`addPropertyChangeListener, firePropertyChange, getAsText, getCustomEditor, getJavaInitializationString, getSource, getTags, getValue, isPaintable, paintValue, removePropertyChangeListener, setAsText, setSource, setValue, supportsCustomEditor`

#### Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructors

### RealArrayEditor

```
public RealArrayEditor(boolean fixedDimension,
                       int dimension)
```

This creates a new instance of `RealArrayEditor` that has a fixed dimension

### RealArrayEditor

```
public RealArrayEditor()
```

Creates a new instance of `RealArrayEditor`, and creates the action listener on the button.

## Methods

### addPropertyChangeListener

```
public void addPropertyChangeListener(java.beans.PropertyChangeListener listener)
```

### getAsText

```
public String getAsText()
```

(continued from last page)

---

## getCustomEditor

```
public java.awt.Component getCustomEditor()
```

---

## getJavaInitializationString

```
public String getJavaInitializationString()
```

Does nothing, and returns null.

---

## getTags

```
public String[] getTags()
```

Does nothing, and returns null.

---

## getValue

```
public Object getValue()
```

---

## isPaintable

```
public boolean isPaintable()
```

---

## paintValue

```
public void paintValue(java.awt.Graphics gfx,  
    java.awt.Rectangle box)
```

---

## removePropertyChangeListener

```
public void removePropertyChangeListener(java.beans.PropertyChangeListener listener)
```

---

## setAsText

```
public void setAsText(String text)  
    throws IllegalArgumentException
```

---

## init

```
public void init(java.beans.PropertyDescriptor descriptor)
```

---

## setValue

```
public void setValue(Object value)
```

If this array is of fixed width. This will adjust the passed value to fit the given dimension.

---

## supportsCustomEditor

```
public boolean supportsCustomEditor()
```

---

## setDimension

```
public void setDimension(int dimension)
```

Sets the dimension on a given array. This also sets the editor to be fixed dimensioned, and resizes the array to match the new dimension.

---

## getModel

```
public AbstractModel getModel()
```

---

## setModel

```
public void setModel(AbstractModel model)
```



## com.cafean.client.ui.beans

### Class RealBeanEditor

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JPanel
                          |-- com.cafean.client.ui.beans.RealBeanEditor

```

#### All Implemented Interfaces:

[ModelDependent](#), [IntrospectingEditor](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

#### Direct Known Subclasses:

[NamelistRealEditor](#)

public class **RealBeanEditor**

extends JPanel

implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, [IntrospectingEditor](#), [ModelDependent](#)

A PropertyEditor wrapper for a RealTextField used when editing a property of type Real of a JavaBean.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

|        |   |
|--------|---|
| public | RealBeanEditor()<br>Creates new form RealBeanEditor |
|--------|---|

## Method Summary

|                    |  |
|--------------------|--|
| void               | firePropertyChange()<br>Report that we have been modified to any interested listeners. |
| String             | getAsText()  |
| java.awt.Component | getCustomEditor()  |

|                               |   |
|-------------------------------|---|
| String                        | getJavaInitializationString()                             |
| <a href="#">AbstractModel</a> | getModel()  |
| String[]                      | getTags()   |
| Object                        | getValue()  |
| void                          | init(java.beans.PropertyDescriptor descriptor)            |
| boolean                       | isPaintable()   |
| void                          | paintValue(java.awt.Graphics gfx, java.awt.Rectangle box) |
| void                          | setAsText(String text)                                    |
| void                          | setForeground(java.awt.Color fg)                          |
| void                          | setModel(AbstractModel model)                             |
| void                          | setValue(Object value)                                    |
| boolean                       | supportsCustomEditor()                                    |

**Methods inherited from class javax.swing.JPanel**

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

**Methods inherited from class javax.swing.JComponent**

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

**Methods inherited from class java.awt.Container**

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusBackward, transferFocusDownCycle, update, validate

**Methods inherited from class java.awt.Component**

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, dispatchEvent, doLayout, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, validate

**Methods inherited from class java.lang.Object**

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

## Constructors

### RealBeanEditor

```
public RealBeanEditor()  
    Creates new form RealBeanEditor
```

## Methods

### setForeground

```
public void setForeground(java.awt.Color fg)
```

### getAsText

```
public String getAsText()
```

### setAsText

```
public void setAsText(String text)  
    throws IllegalArgumentException
```

### init

```
public void init(java.beans.PropertyDescriptor descriptor)
```

### getValue

```
public Object getValue()
```

### setValue

```
public void setValue(Object value)
```

### isPaintable

```
public boolean isPaintable()
```

### getJavaInitializationString

```
public String getJavaInitializationString()
```

### getTags

```
public String[] getTags()
```

(continued from last page)

---

## paintValue

```
public void paintValue(java.awt.Graphics gfx,  
                        java.awt.Rectangle box)
```

---

## getCustomEditor

```
public java.awt.Component getCustomEditor()
```

---

## supportsCustomEditor

```
public boolean supportsCustomEditor()
```

---

## firePropertyChange

```
public void firePropertyChange()
```

Report that we have been modified to any interested listeners.

---

## getModel

```
public AbstractModel getModel()
```

---

## setModel

```
public void setModel(AbstractModel model)
```

---

**Package**

**com.cafean.client.ui.tools**

## com.cafean.client.ui.tools Class AnnotationAction

```

java.lang.Object
  |
  +- javax.swing.AbstractAction
    |
    +- com.cafean.client.ui.tools.ToolboxAction
      |
      +- com.cafean.client.ui.tools.AnnotationAction

```

public abstract class **AnnotationAction**  
 extends [ToolboxAction](#)

A ToolboxAction derivative used to create various Annotations.

### Constructor Summary

|        |   |
|--------|---|
| public | AnnotationAction(Toolbox toolbox,String name,Icon icon,String description)<br><br>Creates a new action for the given toolbox. |
|--------|---|

### Method Summary

|                     |   |
|---------------------|---|
| abstract JComponent | createComponent()<br><br>Creates a new instance of this action's Annotation type. |
|---------------------|---|

#### Methods inherited from class [com.cafean.client.ui.tools.ToolboxAction](#)

[actionPerformed](#), [getDescription](#), [getIcon](#), [getName](#)

#### Methods inherited from class javax.swing.AbstractAction

addPropertyChangeListener, getKeys, getPropertyChangeListeners, getValue, isEnabled, putValue, removePropertyChangeListener, setEnabled

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### AnnotationAction

```

public AnnotationAction(Toolbox toolbox,
                        String name,
                        Icon icon,
                        String description)

```

(continued from last page)

Creates a new action for the given toolbox.

**Parameters:**

`toolbox` - the Toolbox that is using this action

`name` - a String containing the display name of this action

`description` - a String containing a description of this action suitable for use as a tooltip for a menu item or button.

## Methods

**createComponent**

```
public abstract JComponent createComponent()
```

Creates a new instance of this action's Annotation type.



## com.cafean.client.ui.tools Class BeanAction

```
java.lang.Object
├-- javax.swing.AbstractAction
│   └-- com.cafean.client.ui.tools.ToolboxAction
│       └-- com.cafean.client.ui.tools.BeanAction
```

```
public class BeanAction
extends ToolboxAction
```

A ToolboxAction derivative used to create any kind of bean. The bean's BeanInfo is used to determine its name, icon and description.

### Constructor Summary

|        |  |
|--------|--|
| public | <code>BeanAction(Toolbox toolbox, Class beanClass)</code><br>Creates a new action for creating the given bean. |
|--------|--|

### Method Summary

|            |  |
|------------|--|
| JComponent | <code>createComponent()</code><br>Creates a new instance of the bean this action represents. |
|------------|--|

#### Methods inherited from class [com.cafean.client.ui.tools.ToolboxAction](#)

[actionPerformed](#), [getDescription](#), [getIcon](#), [getName](#)

#### Methods inherited from class javax.swing.AbstractAction

`addPropertyChangeListener`, `getKeys`, `getPropertyChangeListeners`, `getValue`, `isEnabled`, `putValue`, `removePropertyChangeListener`, `setEnabled`

#### Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### BeanAction

```
public BeanAction(Toolbox toolbox,
                  Class beanClass)
```

Creates a new action for creating the given bean.

(continued from last page)

**Parameters:**

`beanClass` - the Class of the bean to be created by this action

## Methods

**createComponent**

```
public JComponent createComponent()
```

Creates a new instance of the bean this action represents.

## com.cafean.client.ui.tools

### Class CategoryAction

```

java.lang.Object
  |
  +- javax.swing.AbstractAction
    |
    +- com.cafean.client.ui.tools.ToolboxAction
      |
      +- com.cafean.client.ui.tools.CategoryAction
  
```

```

public class CategoryAction
extends ToolboxAction
  
```

#### Constructor Summary

|        |  |
|--------|--|
| public | <code>CategoryAction(Toolbox toolbox,Category category)</code> |
|--------|--|

#### Method Summary

|                          |                            |
|--------------------------|----------------------------|
| <a href="#">Category</a> | <code>getCategory()</code> |
|--------------------------|----------------------------|

Methods inherited from class [com.cafean.client.ui.tools.ToolboxAction](#)

[actionPerformed](#), [getDescription](#), [getIcon](#), [getName](#)

Methods inherited from class javax.swing.AbstractAction

`addPropertyChangeListener`, `getKeys`, `getPropertyChangeListeners`, `getValue`, `isEnabled`, `putValue`, `removePropertyChangeListener`, `setEnabled`

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

#### Constructors

##### CategoryAction

```

public CategoryAction(Toolbox toolbox,
                     Category category)
  
```

#### Methods

(continued from last page)

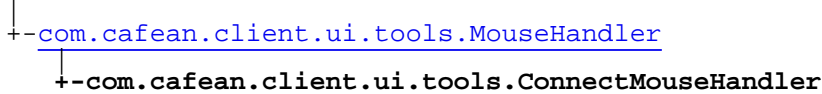
## getCategory

```
public Category getCategory()
```

## com.cafean.client.ui.tools

# Class ConnectMouseHandler

java.lang.Object



public class **ConnectMouseHandler**  
 extends [MouseHandler](#)

A MouseHandler for Connection events on the ZoomablePanel

## Field Summary

|                                 |   |
|---------------------------------|---|
| static final<br>java.awt.Cursor | CURSOR_CONNECT<br>The cursor for the connect tool   |
| static final<br>java.awt.Cursor | CURSOR_CONNECT_COMPLETE<br>The cursor for the connect tool when connection completion is possible |
| static final<br>java.awt.Cursor | CURSOR_CONNECT_OFF<br>The cursor for the connect tool when no connection is possible              |

## Constructor Summary

|        |  |
|--------|--|
| public | ConnectMouseHandler(ZoomablePanel parent)<br>Creates a new mouse handler with the given parent |
|--------|--|

## Method Summary

|                                   |   |
|-----------------------------------|---|
| void                              | activate()<br>{ @inheritDoc}  |
| void                              | completeConnection(DrawnComponent targetDrawing, ConnectingPt point)<br>Completes the current connection.                       |
| void                              | deactivate()<br>{ @inheritDoc}  |
| ImageIcon                         | getButtonIcon()<br>Returns the icon for use in creating an activation toolbar button for this handler.                          |
| <a href="#">AbstractComponent</a> | getConnectionSource()<br>Returns the AbstractComponent that started the connection.   |
| java.awt.Cursor                   | getCurrentCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for the views based off of the currently selected tool. |

|                 |   |
|-----------------|---|
| java.awt.Cursor | getCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for this MouseHandler within the context of the given MouseEvent   |
| boolean         | isInProgress()<br>Returns true if a connection draw is in progress.   |
| void            | mouseClicked(java.awt.event.MouseEvent orig_evt)<br>When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component. |
| void            | mouseDragged(java.awt.event.MouseEvent orig_evt)  |
| void            | mouseMoved(java.awt.event.MouseEvent orig_evt)  |
| void            | mousePressed(java.awt.event.MouseEvent orig_evt)<br>When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component. |
| void            | mouseReleased(java.awt.event.MouseEvent orig_evt)   |
| boolean         | requiresTarget()<br>Determines if the ConnectingPt that started this connection requires a connection point for its target.   |
| void            | setConnectionStart(java.awt.Point start, DrawnComponent comp, ConnectingPt point)   |
| String          | toString()<br>returns a string representation of this handler   |

#### Methods inherited from class [com.cafean.client.ui.tools.MouseHandler](#)

[activate](#), [deactivate](#), [getButtonIcon](#), [getCurrentCursor](#), [getCursor](#), [getHandlerID](#), [getTooltipText](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#), [setHandlerID](#), [toString](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Fields

### CURSOR\_CONNECT

```
public static final java.awt.Cursor CURSOR_CONNECT
```

The cursor for the connect tool

### CURSOR\_CONNECT\_OFF

```
public static final java.awt.Cursor CURSOR_CONNECT_OFF
```

The cursor for the connect tool when no connection is possible

---

## CURSOR\_CONNECT\_COMPLETE

```
public static final java.awt.Cursor CURSOR_CONNECT_COMPLETE
```

The cursor for the connect tool when connection completion is possible

## Constructors

### ConnectMouseHandler

```
public ConnectMouseHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### getButtonIcon

```
public ImageIcon getButtonIcon()
```

Returns the icon for use in creating an activation toolbar button for this handler. The returned icon need not be cached as this method will only be called once per instance.

**Returns:**

the javax.swing.ImageIcon to be used as this handler's toggle button icon.

---

### toString

```
public String toString()
```

returns a string representation of this handler

---

### activate

```
public void activate()
```

Sets up this MouseHandler to be ready to receive MouseEvents

---

### deactivate

```
public void deactivate()
```

Deactivates this MouseHandler and cleans up any current operations.

---

### isInProgress

```
public boolean isInProgress()
```

Returns true if a connection draw is in progress.

---

### getCurrentCursor

```
public java.awt.Cursor getCurrentCursor(java.awt.event.MouseEvent evt)
```

Retrieves the cursor for the views based off of the currently selected tool.

---

### getCursor

```
public java.awt.Cursor getCursor(java.awt.event.MouseEvent evt)
```

Retrieves the cursor for this MouseHandler within the context of the given MouseEvent

---

## mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component.

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component.

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent orig_evt)
```

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

---

## setConnectionStart

```
public void setConnectionStart(java.awt.Point start,  
    DrawnComponent comp,  
    ConnectingPt point)
```

---

## completeConnection

```
public void completeConnection(DrawnComponent targetDrawing,  
    ConnectingPt point)
```

Completes the current connection.

### Parameters:

targetDrawing - the DrawnComponent to connect to the component of  
point - the ConnectingPt the connection ended on

### See Also:

```
.setConnectionStart(Point,DrawnComponent,ConnectingPt)()
```

---

## requiresTarget

```
public boolean requiresTarget()
```

Determines if the ConnectingPt that started this connection requires a connection point for its target.

---



(continued from last page)

## **getConnectionSource**

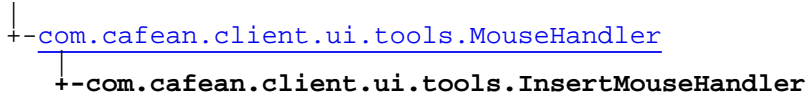
```
public AbstractComponent getConnectionSource( )
```

Returns the AbstractComponent that started the connection.

## com.cafean.client.ui.tools

### Class InsertMouseHandler

java.lang.Object



public class **InsertMouseHandler**  
 extends [MouseHandler](#)

The Insert Tool, a MouseHandler for insertion into a ZoomablePanel. Handles the creation and insertion of new components and the management of AbstractInsertHandler instances used to create and dimension various Insertable components.

#### See Also:

Insertable, AbstractInsertHandler

### Constructor Summary

|        |   |
|--------|---|
| public | InsertMouseHandler(ZoomablePanel parent)<br>Creates a new mouse handler with the given parent |
|--------|---|

### Method Summary

|                 |   |
|-----------------|---|
| void            | cancelInsert()<br>Cancels the current insertion   |
| void            | finishInsert(java.awt.event.MouseEvent orig_evt)<br>Completes the current insertion by adding the component and initializing it.  |
| java.awt.Cursor | getCurrentCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for the views based off of the currently selected tool.   |
| void            | mouseClicked(java.awt.event.MouseEvent orig_evt)<br>Handles mouse clicked events by inserting a new component.  |
| void            | mouseDragged(java.awt.event.MouseEvent orig_evt)<br>Forwards mouseDragged events to the child AbstractInsertHandler, if one is currently in use.  |
| void            | mouseMoved(java.awt.event.MouseEvent orig_evt)<br>Forwards mouseMoved events to the child AbstractInsertHandler, if one is currently in use.  |
| void            | mousePressed(java.awt.event.MouseEvent orig_evt)<br>the mouse pressed event is not handled by this tool but is instead forwarded.   |
| void            | mouseReleased(java.awt.event.MouseEvent orig_evt)<br>Forwards mouseReleased events to the child AbstractInsertHandler, if one is currently in use, otherwise events are forwarded via #forwardMouseEvent. |

**Methods inherited from class [com.cafean.client.ui.tools.MouseHandler](#)**

[activate](#), [deactivate](#), [getButtonIcon](#), [getCurrentCursor](#), [getCursor](#), [getHandlerID](#), [getTooltipText](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#), [setHandlerID](#), [toString](#)

**Methods inherited from class [java.lang.Object](#)**

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### InsertMouseHandler

```
public InsertMouseHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### getCurrentCursor

```
public java.awt.Cursor getCurrentCursor(java.awt.event.MouseEvent evt)
```

Retrieves the cursor for the views based off of the currently selected tool.

### mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

the mouse pressed event is not handled by this tool but is instead forwarded.

### mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent orig_evt)
```

Forwards mouseMoved events to the child AbstractInsertHandler, if one is currently in use.

### mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

Forwards mouseDragged events to the child AbstractInsertHandler, if one is currently in use.

### mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

Forwards mouseReleased events to the child AbstractInsertHandler, if one is currently in use, otherwise events are forwarded via #forwardMouseEvent.

### mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

Handles mouse clicked events by inserting a new component. If a child AbstractInsertHandler is in use, this event is forwarded to it instead of being handled here.

(continued from last page)

**See Also:**`.mousePressed()`

---

**finishInsert**

```
public void finishInsert(java.awt.event.MouseEvent orig_evt)
```

Completes the current insertion by adding the component and initializing it.

---

**cancelInsert**

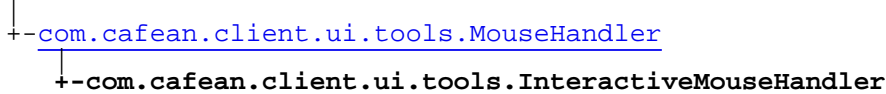
```
public void cancelInsert()
```

Cancels the current insertion

## com.cafean.client.ui.tools

# Class InteractiveMouseHandler

java.lang.Object



public class **InteractiveMouseHandler**  
 extends [MouseHandler](#)

A MouseHandler for interactive handling of events on the ZoomablePanel

## Constructor Summary

|        |  |
|--------|--|
| public | InteractiveMouseHandler(ZoomablePanel parent)<br>Creates a new mouse handler with the given parent |
|--------|--|

## Method Summary

|           |  |
|-----------|--|
| void      | activate()<br>Returns the icon for use in creating an activation toolbar button for this handler.      |
| ImageIcon | getButtonIcon()<br>Returns the icon for use in creating an activation toolbar button for this handler. |
| void      | mouseDragged(java.awt.event.MouseEvent orig_evt)   |
| void      | mousePressed(java.awt.event.MouseEvent orig_evt)   |
| void      | mouseReleased(java.awt.event.MouseEvent orig_evt)  |
| String    | toString()<br>returns a string representation of this handler  |

### Methods inherited from class [com.cafean.client.ui.tools.MouseHandler](#)

[activate](#), [deactivate](#), [getButtonIcon](#), [getCurrentCursor](#), [getCursor](#), [getHandlerID](#), [getTooltipText](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#), [setHandlerID](#), [toString](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

(continued from last page)

## InteractiveMouseHandler

```
public InteractiveMouseHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### activate

```
public void activate()
```

### getButtonIcon

```
public ImageIcon getButtonIcon()
```

Returns the icon for use in creating an activation toolbar button for this handler. The returned icon need not be cached as this method will only be called once per instance.

**Returns:**

the javax.swing.ImageIcon to be used as this handler's toggle button icon.

### toString

```
public String toString()
```

returns a string representation of this handler

### mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

### mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

### mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

## com.cafean.client.ui.tools Class MouseHandler

java.lang.Object  
└─com.cafean.client.ui.tools.MouseHandler

### All Implemented Interfaces:

java.awt.event.MouseMotionListener, java.awt.event.MouseListener

### Direct Known Subclasses:

[SelectMouseHandler](#), [ZoomMouseHandler](#), [InsertMouseHandler](#), [InteractiveMouseHandler](#), [PanMouseHandler](#), [ConnectMouseHandler](#)

public abstract class **MouseHandler**

extends Object

implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener

The base class for the MouseListeners/MouseMotionListeners in a DrawnView that correspond directly to the "Tool" buttons (select, pan, zoom, etc.)

These tools all have a unique ID that in many places to determine the proper path for event dispatching. This ID is allocated and managed automatically by ZoomablePanel's addMouseHandler method and should not be altered.

This base class provides a key handler to disable this handler and switch to the select handler. To preserve this behavior ensure that the MouseHandler methods activate() and deactivate() are called when overridden. Further customization of this cancelling behavior can be achieved by overriding escapePressed() and performing additional operations before calling the base implementation.

Extensions of this class should override the necessary MouseListener and MouseMotionListener methods to achieve the desired mouse feedback.

### See Also:

[com.cafean.client.ui.ZoomablePanel.addMouseHandler\(\)](#), [com.cafean.client.ui.ZoomablePanel.removeMouseHandler\(\)](#)

## Constructor Summary

|        |  |
|--------|--|
| public | MouseHandler(ZoomablePanel parent)<br>Creates a new instance of MouseHandler |
|--------|--|

## Method Summary

|           |  |
|-----------|--|
| void      | activate()<br>Sets up this MouseHandler to be ready to receive MouseEvents                             |
| void      | deactivate()<br>Deactivates this MouseHandler and cleans up any current operations.                    |
| ImageIcon | getButtonIcon()<br>Returns the icon for use in creating an activation toolbar button for this handler. |

|                 |  |
|-----------------|--|
| java.awt.Cursor | getCurrentCursor( java.awt.event.MouseEvent evt )<br>Retrieves the cursor for the views based off of the currently selected tool.    |
| java.awt.Cursor | getCursor( java.awt.event.MouseEvent evt )<br>Retrieves the cursor for this MouseHandler within the context of the given MouseEvent. |
| int             | getHandlerID()<br>Gets this handler's ID.  |
| String          | getTooltipText()<br>returns the tooltip text to use for this MouseHandler's toggle button or null if no text is desired.             |
| void            | mouseClicked( java.awt.event.MouseEvent orig_evt )<br>Forward the mouseClicked event to the underlying child container.              |
| void            | mouseEntered( java.awt.event.MouseEvent orig_evt )<br>Forward the mouseEntered event to the underlying child container.              |
| void            | mouseExited( java.awt.event.MouseEvent orig_evt )<br>Forward the mouseExited event to the underlying child container.                |
| void            | mouseMoved( java.awt.event.MouseEvent orig_evt )<br>Forward the mouseMoved event to the underlying child container.                  |
| void            | setHandlerID(int id)<br>Sets this handler's ID.  |
| String          | toString()<br>returns a string representation of this handler including its class name and handler id.                               |

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### MouseHandler

public **MouseHandler**([ZoomablePanel](#) parent)

Creates a new instance of MouseHandler

## Methods

### activate

public void **activate**()

Sets up this MouseHandler to be ready to receive MouseEvents



(continued from last page)

---

## deactivate

```
public void deactivate()
```

Deactivates this MouseHandler and cleans up any current operations.

---

## getCurrentCursor

```
public java.awt.Cursor getCurrentCursor( java.awt.event.MouseEvent evt )
```

Retrieves the cursor for the views based off of the currently selected tool.

---

## getCursor

```
public java.awt.Cursor getCursor( java.awt.event.MouseEvent evt )
```

Retrieves the cursor for this MouseHandler within the context of the given MouseEvent.

### Parameters:

evt - the MouseEvent corresponding to the current Cursor change. **May be null!**

---

## getButtonIcon

```
public ImageIcon getButtonIcon()
```

Returns the icon for use in creating an activation toolbar button for this handler. The returned icon need not be cached as this method will only be called once per instance. The default implementation simply returns null.

### Returns:

the javax.swing.ImageIcon to be used as this handler's toggle button icon.

---

## setHandlerID

```
public final void setHandlerID(int id)
```

Sets this handler's ID. This ID is generated at view creation time by ZoomablePanel and should not be stored, cached or changed.

---

## getHandlerID

```
public final int getHandlerID()
```

Gets this handler's ID. This ID is generated at view creation time by ZoomablePanel and should not be stored, cached or changed.

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

Forward the mouseClicked event to the underlying child container.

### See Also:

.mousePressed()

---

## mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent orig_evt)
```

Forward the mouseEntered event to the underlying child container.

### See Also:

.mousePressed()

---

## mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent orig_evt)
```

Forward the mouseMoved event to the underlying child container. Change the cursor as it moves over selected components.

### See Also:

```
.mousePressed()
```

---

## mouseExited

```
public void mouseExited(java.awt.event.MouseEvent orig_evt)
```

Forward the mouseExited event to the underlying child container.

### See Also:

```
.mousePressed()
```

---

## toString

```
public String toString()
```

returns a string representation of this handler including its class name and handler id.

---

## getTooltipText

```
public String getTooltipText()
```

returns the tooltip text to use for this MouseHandler's toggle button or null if no text is desired.

## com.cafean.client.ui.tools

### Class PanMouseHandler

```

java.lang.Object
├── com.cafean.client.ui.tools.MouseHandler
│   └── com.cafean.client.ui.tools.PanMouseHandler

```

public class **PanMouseHandler**  
 extends [MouseHandler](#)

A MouseHandler for Pan events on the ZoomablePanel

#### Field Summary

|                                 |   |
|---------------------------------|---|
| static final<br>java.awt.Cursor | CURSOR_PAN<br>The cursor for the pan tool, when a mouse button is not being pressed   |
| static final<br>java.awt.Cursor | CURSOR_PAN_GRIP<br>The cursor for the pan tool, while a mouse button is being pressed |

#### Constructor Summary

|        |  |
|--------|--|
| public | PanMouseHandler(ZoomablePanel parent)<br>Creates a new mouse handler with the given parent |
|--------|--|

#### Method Summary

|                 |   |
|-----------------|---|
| ImageIcon       | getButtonIcon()<br>Returns the icon for use in creating an activation toolbar button for this handler.  |
| java.awt.Cursor | getCurrentCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for the views based off of the currently selected tool.   |
| void            | mouseClicked(java.awt.event.MouseEvent orig_evt)  |
| void            | mouseDragged(java.awt.event.MouseEvent orig_evt)<br>Handle mouseDragged events for move and resize operations.  |
| void            | mousePressed(java.awt.event.MouseEvent orig_evt)<br>When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component. |
| void            | mouseReleased(java.awt.event.MouseEvent orig_evt)   |
| String          | toString()<br>returns a string representation of this handler   |

**Methods inherited from class** [com.cafean.client.ui.tools.MouseHandler](#)

[activate](#), [deactivate](#), [getButtonIcon](#), [getCurrentCursor](#), [getCursor](#), [getHandlerID](#), [getTooltipText](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#), [setHandlerID](#), [toString](#)

**Methods inherited from class** `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Fields

### **CURSOR\_PAN**

```
public static final java.awt.Cursor CURSOR_PAN
```

The cursor for the pan tool, when a mouse button is not being pressed

### **CURSOR\_PAN\_GRIP**

```
public static final java.awt.Cursor CURSOR_PAN_GRIP
```

The cursor for the pan tool, while a mouse button is being pressed

## Constructors

### **PanMouseHandler**

```
public PanMouseHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### **getButtonIcon**

```
public ImageIcon getButtonIcon()
```

Returns the icon for use in creating an activation toolbar button for this handler. The returned icon need not be cached as this method will only be called once per instance.

**Returns:**

the `javax.swing.ImageIcon` to be used as this handler's toggle button icon.

### **toString**

```
public String toString()
```

returns a string representation of this handler

### **mousePressed**

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component.

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

Handle mouseDragged events for move and resize operations.

### See Also:

```
.mousePressed()
```

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

---

## mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

---

## getCurrentCursor

```
public java.awt.Cursor getCurrentCursor(java.awt.event.MouseEvent evt)
```

Retrieves the cursor for the views based off of the currently selected tool.

## com.cafean.client.ui.tools

### Class SelectMouseHandler

java.lang.Object

```

  |
+-com.cafean.client.ui.tools.MouseHandler
  |
+-com.cafean.client.ui.tools.SelectMouseHandler

```

public class **SelectMouseHandler**  
 extends [MouseHandler](#)

A MouseHandler for the Select Tool, handling typical movement and resizing manipulation in the ZoomablePanel.

### Constructor Summary

|        |   |
|--------|---|
| public | SelectMouseHandler(ZoomablePanel parent)<br>Creates a new mouse handler with the given parent |
|--------|---|

### Method Summary

|                 |   |
|-----------------|---|
| static boolean  | canBeResized(java.awt.Component component)<br>Returns true if the given component can be resized by this handler.   |
| void            | deactivate()  |
| ImageIcon       | getButtonIcon()<br>Returns the icon for use in creating an activation toolbar button for this handler.  |
| java.awt.Cursor | getCurrentCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for the views based off of the currently selected tool.   |
| java.awt.Cursor | getCursor(java.awt.event.MouseEvent orig_evt)   |
| void            | mouseDragged(java.awt.event.MouseEvent orig_evt)<br>Handle mouseDragged events for move and resize operations.  |
| void            | mousePressed(java.awt.event.MouseEvent orig_evt)<br>When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component. |
| void            | mouseReleased(java.awt.event.MouseEvent orig_evt)<br>Handle mouseReleased events to support move, resize, select operations.  |
| String          | toString()<br>returns a string representation of this handler   |

Methods inherited from class [com.cafean.client.ui.tools.MouseHandler](#)

[activate](#), [deactivate](#), [getButtonIcon](#), [getCurrentCursor](#), [getCursor](#), [getHandlerID](#), [getTooltipText](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#), [setHandlerID](#), [toString](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### SelectMouseHandler

```
public SelectMouseHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### deactivate

```
public void deactivate()
```

### getButtonIcon

```
public ImageIcon getButtonIcon()
```

Returns the icon for use in creating an activation toolbar button for this handler. The returned icon need not be cached as this method will only be called once per instance.

**Returns:**

the `javax.swing.ImageIcon` to be used as this handler's toggle button icon.

### toString

```
public String toString()
```

returns a string representation of this handler

### mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component.

### mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

Handle mouseDragged events for move and resize operations.

**See Also:**

`.mousePressed()`

(continued from last page)

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

Handle mouseReleased events to support move, resize, select operations. Forward the mouseExited event to the underlying child container.

### See Also:

```
.mousePressed()
```

---

## canBeResized

```
public static boolean canBeResized(java.awt.Component component)
```

Returns true if the given component can be resized by this handler.

---

## getCurrentCursor

```
public java.awt.Cursor getCurrentCursor(java.awt.event.MouseEvent evt)
```

Retrieves the cursor for the views based off of the currently selected tool.

---

## getCursor

```
public java.awt.Cursor getCursor(java.awt.event.MouseEvent orig_evt)
```



## com.cafean.client.ui.tools

### Class ToolboxAction

```

java.lang.Object
  |
  +- javax.swing.AbstractAction
    |
    +- com.cafean.client.ui.tools.ToolboxAction
  
```

**Direct Known Subclasses:**

[BeanAction](#), [AnnotationAction](#), [CategoryAction](#)

public abstract class **ToolboxAction**  
 extends `AbstractAction`

The base class for all Action objects used in the Toolbox to create components, beans, or annotations.

### Constructor Summary

|        |  |
|--------|--|
| public | <code>ToolboxAction(Toolbox toolbox,String name,Icon icon,String description)</code><br>Creates a new <code>ToolboxAction</code> with the given toolbox. |
|--------|--|

### Method Summary

|        |   |
|--------|---|
| void   | <code>actionPerformed(java.awt.event.ActionEvent e)</code><br>Responds to <code>ActionEvents</code> by setting this as the current <code>ToolboxAction</code> in this action's Toolbox. |
| String | <code>getDescription()</code><br>Retrieves this action's description.   |
| Icon   | <code>getIcon()</code><br>Retrieves this action's icon.   |
| String | <code>getName()</code><br>Retrieves this action's name.   |

#### Methods inherited from class `javax.swing.AbstractAction`

`addPropertyChangeListener`, `getKeys`, `getPropertyChangeListeners`, `getValue`, `isEnabled`, `putValue`, `removePropertyChangeListener`, `setEnabled`

#### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructors

(continued from last page)

## ToolboxAction

```
public ToolboxAction(Toolbox toolbox,  
                    String name,  
                    Icon icon,  
                    String description)
```

Creates a new `ToolboxAction` with the given toolbox. The given name, icon and description are passed to the super class via `#putValue`.

### Parameters:

`toolbox` - the `Toolbox` that is using this action  
`name` - a `String` containing the display name of this action  
`description` - a `String` containing a description of this action suitable for use as a tooltip for a menu item or button.

## Methods

### actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

Responds to `ActionEvents` by setting this as the current `ToolboxAction` in this action's `Toolbox`.

### getName

```
public String getName()
```

Retrieves this action's name.

### getDescription

```
public String getDescription()
```

Retrieves this action's description.

### getIcon

```
public Icon getIcon()
```

Retrieves this action's icon.

## com.cafean.client.ui.tools Interface ToolChangeListener

**All Known Implementing Classes:**  
ZoomablePanel

---

public interface **ToolChangeListener**

An interface describing a listener for changes in the current tool.

---

### Method Summary

|      |   |
|------|---|
| void | toolChanged(int oldTool,int newTool)  |
|      | Notifies this listener that the current tool has changed from the old tool to the new tool. |

---

### Methods

#### toolChanged

```
public void toolChanged(int oldTool,  
                        int newTool)
```

Notifies this listener that the current tool has changed from the old tool to the new tool. Tools are one of the TOOL\_\* enumerations in Toolbox.

**See Also:**

Toolbox

## com.cafean.client.ui.tools Class ZoomMouseHandler

```
java.lang.Object
├── com.cafean.client.ui.tools.MouseHandler
│   └── com.cafean.client.ui.tools.ZoomMouseHandler
```

```
public class ZoomMouseHandler
extends MouseHandler
```

A MouseHandler for Zoom events on the ZoomablePanel

### Field Summary

|                                 |   |
|---------------------------------|---|
| static final<br>java.awt.Cursor | CURSOR_ZOOM_IN<br>The cursor for the Zoom Tool  |
| static final<br>java.awt.Cursor | CURSOR_ZOOM_OUT<br>The cursor for the Zoom Tool |

### Constructor Summary

|        |   |
|--------|---|
| public | ZoomMouseHandler(ZoomablePanel parent)<br>Creates a new mouse handler with the given parent |
|--------|---|

### Method Summary

|                 |   |
|-----------------|---|
| void            | activate()  |
| void            | deactivate()  |
| ImageIcon       | getButtonIcon()<br>Returns the icon for use in creating an activation toolbar button for this handler.                          |
| java.awt.Cursor | getCurrentCursor(java.awt.event.MouseEvent evt)<br>Retrieves the cursor for the views based off of the currently selected tool. |
| void            | mouseDragged(java.awt.event.MouseEvent evt)<br>Handle mouseDragged events for move and resize operations.                       |
| void            | mousePressed(java.awt.event.MouseEvent orig_evt)  |
| void            | mouseReleased(java.awt.event.MouseEvent orig_evt)<br>Handle mouseReleased events to support move, resize, select operations.    |

|        |  |
|--------|--|
| String | <code>toString()</code><br>returns a string representation of this handler |
|--------|--|

Methods inherited from class [com.cafean.client.ui.tools.MouseHandler](#)

[activate](#), [deactivate](#), [getButtonIcon](#), [getCurrentCursor](#), [getCursor](#), [getHandlerID](#), [getTooltipText](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#), [setHandlerID](#), [toString](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Fields

### CURSOR\_ZOOM\_IN

```
public static final java.awt.Cursor CURSOR_ZOOM_IN
```

The cursor for the Zoom Tool

### CURSOR\_ZOOM\_OUT

```
public static final java.awt.Cursor CURSOR_ZOOM_OUT
```

The cursor for the Zoom Tool

## Constructors

### ZoomMouseHandler

```
public ZoomMouseHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### getCurrentCursor

```
public java.awt.Cursor getCurrentCursor(java.awt.event.MouseEvent evt)
```

Retrieves the cursor for the views based off of the currently selected tool.

### getButtonIcon

```
public ImageIcon getButtonIcon()
```

Returns the icon for use in creating an activation toolbar button for this handler. The returned icon need not be cached as this method will only be called once per instance.

**Returns:**

the `javax.swing.ImageIcon` to be used as this handler's toggle button icon.

(continued from last page)

---

## toString

```
public String toString()
```

returns a string representation of this handler

---

## activate

```
public void activate()
```

---

## deactivate

```
public void deactivate()
```

---

## mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

---

## mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent evt)
```

Handle mouseDragged events for move and resize operations.

**See Also:**

`.mousePressed()`

---

## mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

Handle mouseReleased events to support move, resize, select operations. Forward the mouseExited event to the underlying child container.

**See Also:**

`.mousePressed()`

---

---

**Package**

**com.cafean.client.ui.tools.insert**

## com.cafean.client.ui.tools.insert Class AbstractInsertHandler

java.lang.Object

└-com.cafean.client.ui.tools.insert.AbstractInsertHandler

### All Implemented Interfaces:

java.awt.event.MouseMotionListener, java.awt.event.MouseListener

### Direct Known Subclasses:

[RectangularInsertHandler](#), [AbstractPathHandler](#)

public abstract class **AbstractInsertHandler**

extends Object

implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener

The base class of a mouse handler used to control the insertion of a bean, annotation, etc.

## Constructor Summary

|        |   |
|--------|---|
| public | AbstractInsertHandler(ZoomablePanel parent)<br>Creates a new instance of MouseHandler |
|--------|---|

## Method Summary

|                 |   |
|-----------------|---|
| void            | cancelInsert()<br>Cancels the insertion by calling InsertMouseHandler#finishInsert.                                     |
| void            | finishInsert(java.awt.event.MouseEvent orig_evt)<br>Completes the insertion by calling InsertMouseHandler#finishInsert. |
| java.awt.Cursor | getCurrentCursor()<br>Retrieves the cursor for the views based off of the currently selected tool.                      |
| void            | mouseClicked(java.awt.event.MouseEvent orig_evt)<br>This handler ignores mouse clicked events                           |
| void            | mouseEntered(java.awt.event.MouseEvent orig_evt)<br>sets the zoomable panel's cursor to the handler's current cursor    |
| void            | mouseExited(java.awt.event.MouseEvent orig_evt)<br>sets the zoomable panel's cursor to the handler's current cursor     |
| void            | mouseMoved(java.awt.event.MouseEvent orig_evt)<br>sets the zoomable panel's cursor to the handler's current cursor      |

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



## Constructors

### AbstractInsertHandler

```
public AbstractInsertHandler(ZoomablePanel parent)
```

Creates a new instance of MouseHandler

## Methods

### finishInsert

```
public void finishInsert(java.awt.event.MouseEvent orig_evt)
```

Completes the insertion by calling InsertMouseHandler#finishInsert.

**Parameters:**

orig\_evt - the MouseEvent passed from the GlassPane, untransformed

---

### cancelInsert

```
public void cancelInsert()
```

Cancels the insertion by calling InsertMouseHandler#finishInsert.

**Parameters:**

orig\_evt - the MouseEvent passed from the GlassPane, untransformed

---

### getCurrentCursor

```
public java.awt.Cursor getCurrentCursor()
```

Retrieves the cursor for the views based off of the currently selected tool.

---

### mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

This handler ignores mouse clicked events

---

### mouseEntered

```
public void mouseEntered(java.awt.event.MouseEvent orig_evt)
```

sets the zoomable panel's cursor to the handler's current cursor

**See Also:**

.getCurrentCursor()

---

### mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent orig_evt)
```

sets the zoomable panel's cursor to the handler's current cursor

**See Also:**

.getCurrentCursor()

---

## mouseExited

public void **mouseExited**(java.awt.event.MouseEvent orig\_evt)

sets the zoomable panel's cursor to the handler's current cursor

### See Also:

.getCurrentCursor()

## com.cafean.client.ui.tools.insert Class AbstractPathHandler

java.lang.Object

```

  |
+-com.cafean.client.ui.tools.insert.AbstractInsertHandler
  |
+-com.cafean.client.ui.tools.insert.AbstractPathHandler

```

public abstract class **AbstractPathHandler**  
extends [AbstractInsertHandler](#)

A base class for insert handlers that are based on a path of points.

### See Also:

[LineAnnotInsertHandler](#), [PolygonInsertHandler](#)

### Field Summary

|                  |  |
|------------------|--|
| static final int | TOLERANCE<br><br>The number of pixels the mouse is required to move before another point can be added.<br>Value: 5 |
|------------------|--|

### Constructor Summary

|        |  |
|--------|--|
| public | AbstractPathHandler(ZoomablePanel parent)<br><br>Creates a new mouse handler with the given parent |
|--------|--|

### Method Summary

|      |   |
|------|---|
| void | mouseClicked(java.awt.event.MouseEvent orig_evt)<br><br>This insert handler ignores mouseClicked events   |
| void | mouseDragged(java.awt.event.MouseEvent orig_evt)<br><br>Handles mouse moved events by repainting the current rubber bands.  |
| void | mouseMoved(java.awt.event.MouseEvent orig_evt)<br><br>Handles mouse dragged events by repainting the current rubber bands.  |
| void | mousePressed(java.awt.event.MouseEvent orig_evt)<br><br>When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component. |
| void | mouseReleased(java.awt.event.MouseEvent orig_evt)<br><br>Handles mouse released events by adding points to the path (left button) or removing points from the path (right button).            |

Methods inherited from class [com.cafean.client.ui.tools.insert.AbstractInsertHandler](#)

[cancelInsert](#), [finishInsert](#), [getCurrentCursor](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Fields

### TOLERANCE

```
public static final int TOLERANCE
```

The number of pixels the mouse is required to move before another point can be added.

## Constructors

### AbstractPathHandler

```
public AbstractPathHandler(ZoomablePanel parent)
```

Creates a new mouse handler with the given parent

## Methods

### mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

When inactive, mouse events are forwarded as appropriate either to the UI to activate the frame or to the underlying child component.

### mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

This insert handler ignores mouseClicked eventss

### mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

Handles mouse moved events by repainting the current rubber bands. This implementation simply calls #mouseMoved

#### Parameters:

`orig_event` - the MouseEvent fired by the glass pane, untransformed

#### See Also:

`.mouseMoved()`

### mouseMoved

```
public void mouseMoved(java.awt.event.MouseEvent orig_evt)
```

Handles mouse dragged events by repainting the current rubber bands.

#### Parameters:

(continued from last page)

`orig_event` - the MouseEvent fired by the glass pane, untransformed

---

## **mouseReleased**

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

Handles mouse released events by adding points to the path (left button) or removing points from the path (right button).

### **Parameters:**

`orig_event` - the MouseEvent fired by the glass pane, untransformed

# com.cafean.client.ui.tools.insert Interface Insertable

**All Known Implementing Classes:**  
LineAnnotation, RectangularAnnotation, EllipticalAnnotation

public interface **Insertable**

An interface describing JComponents that can be inserted using an AbstractInsertHandler derivative to guide the dimensioning etc.

| Method Summary                        |  |
|---------------------------------------|--|
| <a href="#">AbstractInsertHandler</a> | <code>getNewInsertHandler(ZoomablePanel parent)</code><br><br>Creates a new insert handler appropriate for handling the mouse events required to properly insert this insertable object into the given parent. |

## Methods

### getNewInsertHandler

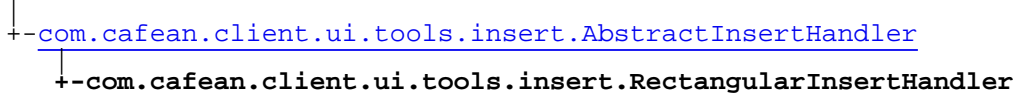
public [AbstractInsertHandler](#) **getNewInsertHandler**([ZoomablePanel](#) parent)

Creates a new insert handler appropriate for handling the mouse events required to properly insert this insertable object into the given parent.

**Returns:**  
an AbstractInsertHandler properly configured for handling the insertion of this component; or null if none is appropriate

## com.cafean.client.ui.tools.insert Class RectangularInsertHandler

java.lang.Object



public class **RectangularInsertHandler**  
extends [AbstractInsertHandler](#)

An insertion handler intended for inserting JComponent derivatives that have rectangular bounds. This handler requires the user to draw a rectangular rubber box, the box is then used to dimension the created JComponent.

### Constructor Summary

|        |   |
|--------|---|
| public | RectangularInsertHandler(ZoomablePanel parent, JComponent comp)<br>Creates a new instance of RectangularInsertHandler |
|--------|---|

### Method Summary

|      |  |
|------|--|
| void | mouseClicked(java.awt.event.MouseEvent orig_evt)<br>Completes the insertion by setting the location of the component to the clicked location and calling finishInsert. |
| void | mouseDragged(java.awt.event.MouseEvent orig_evt)<br>Continues the process of drawing the rubber band rectangle for bounds selection.                                   |
| void | mousePressed(java.awt.event.MouseEvent orig_evt)<br>Begins the process of drawing the rubber band rectangle for bounds selection.                                      |
| void | mouseReleased(java.awt.event.MouseEvent orig_evt)<br>Completes the process of drawing the rubber band rectangle for bounds selection.                                  |

Methods inherited from class [com.cafean.client.ui.tools.insert.AbstractInsertHandler](#)

[cancelInsert](#), [finishInsert](#), [getCurrentCursor](#), [mouseClicked](#), [mouseEntered](#), [mouseExited](#), [mouseMoved](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructors

(continued from last page)

## RectangularInsertHandler

```
public RectangularInsertHandler(ZoomablePanel parent,  
                                JComponent comp)
```

Creates a new instance of RectangularInsertHandler

## Methods

### mouseClicked

```
public void mouseClicked(java.awt.event.MouseEvent orig_evt)
```

Completes the insertion by setting the location of the component to the clicked location and calling finishInsert.

### mousePressed

```
public void mousePressed(java.awt.event.MouseEvent orig_evt)
```

Begins the process of drawing the rubber band rectangle for bounds selection.

### mouseDragged

```
public void mouseDragged(java.awt.event.MouseEvent orig_evt)
```

Continues the process of drawing the rubber band rectangle for bounds selection.

### mouseReleased

```
public void mouseReleased(java.awt.event.MouseEvent orig_evt)
```

Completes the process of drawing the rubber band rectangle for bounds selection. setBounts is then called on the component with the selected rubber band rectangle.



---

**Package**

**com.cafean.CodePlugins**

## com.cafean.CodePlugins

### Class MECodePlugin

```
java.lang.Object
|
+-com.cafean.CodePlugins.MEPlugin
|
+-com.cafean.CodePlugins.MECodePlugin
```

public abstract class **MECodePlugin**  
 extends [MEPlugin](#)

This is used to implement a new analysis code in the model editor. MECodePlugins contain all the functions common between analysis code plugins.

The following is an example implementation of #submitModel.

```
public void submitModel(AbstractModel model)
{
    File tmpFile;
    File tmpSaveFile;
    try {
        tmpFile = File.createTempFile("snap", ".inp");
        tmpSaveFile = File.createTempFile("snap", ".sam");
    } catch( java.io.IOException ioe ) {
        ioe.printStackTrace();
        OptionPane.showMessageDialog( MainFrame.instance,
                                     "Unable to create required temporary files.",
                                     "Submit Failed",
                                     OptionPane.ERROR_MESSAGE );
    }

    return;

    String oldSaveName = model.getSaveFileName();
    model.setSaveFileName( tmpSaveFile.getPath() );
    model.saveModel();
    model.setSaveFileName( oldSaveName );

    boolean success = exportASCII( model, tmpFile );

    if( success ) {
        SubmitDialog dialog = new SubmitDialog( MainFrame.instance,
                                                MainFrame.instance.getOrb(),
                                                getPluginId(),
                                                tmpFile,
                                                tmpSaveFile,
                                                MainFrame.instance.getPrefs() );

        dialog.setVisible(true);
    }
    tmpFile.delete();
    tmpSaveFile.delete();
}
```

**Fields inherited from class** [com.cafean.CodePlugins.MEPlugin](#)

[TYPE\\_CODE\\_PLUGIN](#), [TYPE\\_FEATURE\\_PLUGIN](#)

## Constructor Summary

|        |  |
|--------|--|
| public | MECodePlugin(String name)<br>Creates a new MECodePlugin with the given name. |
|--------|--|

## Method Summary

|   |   |
|---|---|
| abstract<br><a href="#">AbstractModel</a> | createNewModel(boolean initNew)<br>Creates a new instance of the AbstractModel defined by this plugin.  |
| java.io.PrintWriter                       | getAsciiPrintWriter(java.io.Writer output)<br>Retrieves an appropriately configured print writer for use in writing an ASCII representation of individual components. Plugins may choose to override this method to create derivative PrintWriters that allow for different formatting. |
| StyledDocument                            | getAsciiStyledDocument(Writable writeable)<br>Retrieves a StyledDocument suitable for displaying the written ASCII of the given Writable component.   |
| String                                    | getLabel()<br>Retrieves a short string that describes a singular unit of this plugin.   |
| String                                    | getLabelPlural()<br>Retrieves a short string that describes multiple units of this plugin.  |
| ImageIcon                                 | getPluginIcon()<br>Retrieves the icon associated with this plugin.  |
| int                                       | getPluginType()   |
| abstract String                           | getSamPackage()<br>This returns the package name of the SAM files for this plugin.  |
| boolean                                   | isAnimatable()<br>Returns true if this MECodePlugin includes support for View animation.  |
| boolean                                   | isBeanBased()<br>Returns true if this MECodePlugin's design is based on the JavaBeans architecture.   |
| abstract<br><a href="#">AbstractModel</a> | open(java.io.File file)<br>Loads a new AbstractModel from the given local File.   |
| abstract void                             | submitModel(AbstractModel model,boolean local)<br>Submits an AbstractModel to the Calculation Server for execution.   |

### Methods inherited from class [com.cafean.CodePlugins.MEPlugin](#)

[addCurrentExportItems](#), [addCurrentToolsItems](#), [getName](#), [getPluginId](#), [getPluginInfo](#), [getPluginPreferences](#), [getPluginPrereqs](#), [getPluginType](#), [getVersion](#), [loadMainMenuItems](#), [loadSettings](#), [loadViewMenuItems](#), [processCommand](#), [setName](#), [storeSettings](#)

### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### MECodePlugin

```
public MECodePlugin(String name)
```

Creates a new MECodePlugin with the given name.

## Methods

### getPluginType

```
public final int getPluginType()
```

---

### createNewModel

```
public abstract AbstractModel createNewModel(boolean initNew)
```

Creates a new instance of the AbstractModel defined by this plugin. This is a clean model with no components or properties set.

**Parameters:**

initNew - if true, the newly created model should be initialized as if just created by the user.

**Returns:**

an AbstractModel created by this plugin.

---

### open

```
public abstract AbstractModel open(java.io.File file)
```

Loads a new AbstractModel from the given local File.

**Parameters:**

file - the File being read in.

**Returns:**

an AbstractModel created by this plugin and loaded from file.

---

### getSamPackage

```
public abstract String getSamPackage()
```

This returns the package name of the SAM files for this plugin. This is used to determine which plugin a SAM file is specifying.

**Returns:**

String containing the package of the SAM definitions.

---

### submitModel

```
public abstract void submitModel(AbstractModel model,  
    boolean local)
```

Submits an AbstractModel to the Calculation Server for execution.

**Parameters:**

(continued from last page)

`model` - the `AbstractModel` to submit.  
`local` - `TRUE` if this is a local submission.

**See Also:**

`com.cafean.utils.SubmitDialog`

---

## isBeanBased

```
public boolean isBeanBased()
```

Returns true if this `MECodePlugin`'s design is based on the JavaBeans architecture.

---

## isAnimatable

```
public boolean isAnimatable()
```

Returns true if this `MECodePlugin` includes support for View animation. This method is used to enable or disable the playback controls and animated display bean toolbars, as well as other source manager related functionality.

---

## getAsciiStyledDocument

```
public StyledDocument getAsciiStyledDocument(Writeable writeable)
```

Retrieves a `StyledDocument` suitable for displaying the written ASCII of the given `Writeable` component.

**Parameters:**

`writeable` - the `Writeable` component to retrieve an appropriate document for

**Returns:**

the `StyledDocument` to use when displaying ASCII for the given `Writeable`

---

## getAsciiPrintWriter

```
public java.io.PrintWriter getAsciiPrintWriter(java.io.Writer output)
```

Retrieves an appropriately configured print writer for use in writing an ASCII representation of individual components. Plugins may choose to override this method to create derivative `PrintWriters` that allow for different formatting.

**Parameters:**

`output` - the `java.io.Writer` that the returned `PrintWriter` will use for it's output.

**Returns:**

a `java.io.PrintWriter` suitable for printing ASCII representations of this plugin's data.

---

## getPluginIcon

```
public ImageIcon getPluginIcon()
```

Retrieves the icon associated with this plugin.

---

## getLabel

```
public String getLabel()
```

Retrieves a short string that describes a singular unit of this plugin. This is used as the label associated with this plugin in the new file dialog. The default is the ID of the plugin followed by the word "model".

---

## getLabelPlural

```
public String getLabelPlural()
```

Retrieves a short string that describes multiple units of this plugin. This is used as the label associated with this plugin in the navigator. The default is the ID of the plugin followed by the word "models".

---

## com.cafean.CodePlugins

### Class MEFeaturePlugin

```

java.lang.Object
|
+-com.cafean.CodePlugins.MEPlugin
|   |
|   +-com.cafean.CodePlugins.MEFeaturePlugin

```

public abstract class **MEFeaturePlugin**  
 extends [MEPlugin](#)

This is used to implement a new feature plugin for the model editor. It contains all of the functions common between feature plugins in the model editor.

Fields inherited from class [com.cafean.CodePlugins.MEPlugin](#)

[TYPE\\_CODE\\_PLUGIN](#), [TYPE\\_FEATURE\\_PLUGIN](#)

## Constructor Summary

|        |   |
|--------|---|
| public | MEFeaturePlugin(String name)<br>Creates a new instance of MEFeaturePlugin |
|--------|---|

## Method Summary

|                  |   |
|------------------|---|
| int              | getPluginType()<br>Returns the plugin type.   |
| abstract boolean | isAssociated(AbstractModel model)<br>This returns true if this plugin contains data that should be saved in the same SAM file as the given model. |
| abstract void    | load(AbstractModel loadedModel, com.appt.xdr.PibFile pibfile)<br>This is called on all Feature plugins when a model has been loaded.              |
| abstract void    | modelAdded(AbstractModel model)<br>This is called on all Feature plugins when a new model has been added to the model editor.                     |
| abstract void    | modelRemoved(AbstractModel model)<br>This is called on all Feature plugins when a model has been removed from the model editor.                   |
| abstract void    | save(AbstractModel savedModel, com.appt.xdr.PibFile pibfile)<br>This is called on all Feature plugins when a model has been stored.               |

Methods inherited from class [com.cafean.CodePlugins.MEPlugin](#)

[addCurrentExportItems](#), [addCurrentToolsItems](#), [getName](#), [getPluginId](#), [getPluginInfo](#), [getPluginPreferences](#), [getPluginPrereqs](#), [getPluginType](#), [getVersion](#), [loadMainMenuItems](#), [loadSettings](#), [loadViewMenuItems](#), [processCommand](#), [setName](#), [storeSettings](#)

Methods inherited from class [java.lang.Object](#)

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

---

## Constructors

### MEFeaturePlugin

```
public MEFeaturePlugin(String name)
```

Creates a new instance of MEFeaturePlugin

## Methods

### getPluginType

```
public final int getPluginType()
```

---

### load

```
public abstract void load(AbstractModel loadedModel,  
    com.apt.xdr.PibFile pibfile)
```

This is called on all Feature plugins when a model has been loaded. This allows feature plugins that need to load local data associated with that model to read that data from the end of the file.

#### Parameters:

loadedModel - the AbstractModel that has just been loaded from disk.  
pibfile - the local file that is being read in.

---

### save

```
public abstract void save(AbstractModel savedModel,  
    com.apt.xdr.PibFile pibfile)
```

This is called on all Feature plugins when a model has been stored. This allows feature plugins that need to store local data associated with that model to append the data to the end of the file.

#### Parameters:

savedModel - the AbstractModel that has just been saved to disk.  
pibfile - the local file that is being written out.

---

### modelRemoved

```
public abstract void modelRemoved(AbstractModel model)
```

---

### modelAdded

```
public abstract void modelAdded(AbstractModel model)
```

This is called on all Feature plugins when a new model has been added to the model editor. This allows all plugins that need to store local data associated with that type of model to initialize any data they need.

#### Parameters:

model - the AbstractModel just added to the model Editor.

---

## isAssociated

public abstract boolean **isAssociated**([AbstractModel](#) model)

This returns true if this plugin contains data that should be saved in the same SAM file as the given model.

### Parameters:

model – the AbstractModel to check association with

### Returns:

true if this plugin has data that needs to save in the sam file as the model.



## com.cafean.CodePlugins

### Class MEPlugin

java.lang.Object

└--com.cafean.CodePlugins.MEPlugin

**Direct Known Subclasses:**

[MECodePlugin](#), [MEFeaturePlugin](#)

public abstract class **MEPlugin**  
extends Object

The base class for a Model Editor plugin. User defined plugins should not extend this class, but rather extend either MECodePlugin or MEFeaturePlugin.

### Field Summary

|                  |  |
|------------------|--|
| static final int | TYPE_CODE_PLUGIN<br>the typecode for analysis code support plugins<br>Value: 1 |
| static final int | TYPE_FEATURE_PLUGIN<br>the typecode for feature addition plugins<br>Value: 2   |

### Constructor Summary

|        |  |
|--------|--|
| public | MEPlugin(String name)<br>Creates a new MEPlugin with the given name. |
|--------|--|

### Method Summary

|                 |  |
|-----------------|--|
| void            | addCurrentExportItems(JMenu menu)<br>Adds appropriate export menu items to the given JMenu with respect to the type and state of the { @link MainFrame#getCurrentModel current model}. |
| void            | addCurrentToolsItems(JMenu menu)<br>Adds appropriate Tools menu items to the given JMenu with respect to the type and state of the { @link MainFrame#getCurrentModel current model}.   |
| String          | getName()<br>This returns the descriptive name of this plugin.   |
| abstract String | getPluginId()<br>This returns the static plugin id.  |
| abstract String | getPluginInfo()<br>Returns the information about this plugin that will appear in the Plugins dialog.   |

|                   |   |
|-------------------|---|
| Object            | <code>getPluginPreferences()</code><br>Retrieves an Object that follows the Java Beans design paradigm that contains the preferences for this plugin. |
| abstract String[] | <code>getPluginPrereqs()</code><br>Returns the names and versions of the plugins this plugin depends upon.  |
| abstract int      | <code>getPluginType()</code><br>This returns the static type flag that indicates what type of plugin this is.   |
| abstract String   | <code>getVersion()</code><br>Returns the version number of this plugin as a String.   |
| abstract void     | <code>loadMainMenuItems()</code><br>Loads this plugin's menu items into the MainFrame menu bar.   |
| abstract void     | <code>loadSettings(com.cafean.utils.Configurator config)</code><br>Loads user settings for this plugin from the configuration file.                   |
| abstract void     | <code>loadViewMenuItems(DrawnView view)</code><br>Loads this plugin's menu items into a DrawnView's Tools menu.                                       |
| abstract void     | <code>processCommand(Vector command)</code><br>Processes the given batch command.   |
| void              | <code>setName(String name)</code><br>This sets the descriptive name of this plugin.   |
| abstract void     | <code>storeSettings(com.cafean.utils.Configurator config)</code><br>Stores user settings for this plugin into the configuration file.                 |

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Fields

### TYPE\_CODE\_PLUGIN

`public static final int TYPE_CODE_PLUGIN`  
the typecode for analysis code support plugins

### TYPE\_FEATURE\_PLUGIN

`public static final int TYPE_FEATURE_PLUGIN`  
the typecode for feature addition plugins

## Constructors

(continued from last page)

## MEPlugin

```
public MEPlugin(String name)
```

Creates a new MEPlugin with the given name.

**Parameters:**

name - a String containing the new name.

## Methods

### getName

```
public String getName()
```

This returns the descriptive name of this plugin.

**Returns:**

The Descriptive name of this plugin.

### setName

```
public void setName(String name)
```

This sets the descriptive name of this plugin.

**Parameters:**

name - The Descriptive name of this plugin.

### getPluginId

```
public abstract String getPluginId()
```

This returns the static plugin id. This is used to clearly and easily differentiate between plugins.

**Returns:**

The short identifier for this plugin.

### getPluginType

```
public abstract int getPluginType()
```

This returns the static type flag that indicates what type of plugin this is.

**See Also:**

.TYPE\_CODE\_PLUGIN()  
.TYPE\_FEATURE\_PLUGIN()

### processCommand

```
public abstract void processCommand(Vector command)  
throws java.io.IOException
```

Processes the given batch command. This method should assume that the {[@link #getPluginId plugin id](#)} prefix has been removed from the command before passing to processCommand.

**Parameters:**

command - a Vector of Strings containing the batch command to be executed.

**Throws:**

IOException - on failure to process the command.

---

## loadMainMenuItems

```
public abstract void loadMainMenuItems()
```

Loads this plugin's menu items into the MainFrame menu bar. Use the menu insertion function in MainFrame to add new JMenuItem's to the toolbar's menus.

**See Also:**

```
MainFrame.addImportItem()  
MainFrame.addExportItem()  
MainFrame.addMenuItem()
```

---

## loadViewMenuItems

```
public abstract void loadViewMenuItems(DrawnView view)
```

Loads this plugin's menu items into a DrawnView's Tools menu. Use DrawnView#addMenuItem to add new JMenuItem's to tools menu.

**See Also:**

```
DrawnView.addMenuItem()
```

---

## addCurrentExportItems

```
public void addCurrentExportItems(JMenu menu)
```

Adds appropriate export menu items to the given JMenu with respect to the type and state of the { @link MainFrame#getCurrentModel current model}. Note: This is called each time the export menu is selected. To add permanent items use .

**Parameters:**

menu - the JMenu to add export related items to.

---

## addCurrentToolsItems

```
public void addCurrentToolsItems(JMenu menu)
```

Adds appropriate Tools menu items to the given JMenu with respect to the type and state of the { @link MainFrame#getCurrentModel current model}. Note: This is called each time the Tools menu is selected. To add permanent items use .

**Parameters:**

menu - the JMenu to add export related items to.

---

## loadSettings

```
public abstract void loadSettings(com.cafean.utils.Configurator config)
```

Loads user settings for this plugin from the configuration file.

**Parameters:**

config - the Configurator used to access the locally stored settings.

---

## storeSettings

```
public abstract void storeSettings(com.cafean.utils.Configurator config)
```

Stores user settings for this plugin into the configuration file.

**Parameters:**

config - the Configurator used to access the locally stored settings.

---

## getVersion

```
public abstract String getVersion()
```

Returns the version number of this plugin as a String. The returned version number should be formatted as: [major].[minor].[bugfix]. For example: 1.0.2.

**Returns:**

a String containing the version number.

---

## getPluginPrereqs

```
public abstract String[] getPluginPrereqs()
```

Returns the names and versions of the plugins this plugin depends upon. Each plugin prerequisite should be in the form of: [PLUGINID]:[VERSION] with :VERSION being optional.

**Returns:**

a String[] containing all the plugins this depends upon.

**See Also:**

MEPluginData.getPluginPrereqs()

---

## getPluginInfo

```
public abstract String getPluginInfo()
```

Returns the information about this plugin that will appear in the Plugins dialog. This should be either unformatted text, or alternatively, HTML to be placed between existing <HTML> and <BODY> tags.

**Returns:**

a String containing a description of this plugin.

---

## getPluginPreferences

```
public Object getPluginPreferences()
```

Retrieves an Object that follows the Java Beans design paradigm that contains the preferences for this plugin. These preferences must be stored along with other plugin values.

**Returns:**

an Object that contains the plugin's preferences.

---

## com.cafean.CodePlugins

### Class MEPluginData

java.lang.Object

└─com.cafean.CodePlugins.MEPluginData

public abstract class **MEPluginData**  
extends Object

The MEPluginData is used to hold all the data necessary to determine when and how the plugin should be read in. This allows the plugin loader to determine if the appropriate prerequisite plugins have been loaded, and that the necessary classes are available.

### Constructor Summary

|        |  |
|--------|--|
| public | MEPluginData()<br>Creates a new instance of MEPluginData |
|--------|--|

### Method Summary

|                                   |  |
|-----------------------------------|--|
| abstract String[]                 | getClassPrereqs()<br>Returns the name of any specific classes required by this plugin that are not included in the main ModelEditor distribution, the standard Java distribution or this plugin. |
| abstract String                   | getPluginClass()<br>Returns the absolute path to the primary class for the plugin in string form.  |
| abstract String                   | getPluginId()<br>Returns this plugin's plugin-id.  |
| abstract String[]                 | getPluginPrereqs()<br>Returns the names and versions of the plugins this plugin depends upon.  |
| abstract String                   | getVersion()<br>Returns the current plugin version.  |
| abstract <a href="#">MEPlugin</a> | loadPlugin()<br>Creates an instance of the MEPlugin that is described by this MEPluginData object.   |

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### MEPluginData

public **MEPluginData**()

(continued from last page)

Creates a new instance of MEPluginData

## Methods

### getPluginClass

```
public abstract String getPluginClass()
```

Returns the absolute path to the primary class for the plugin in string form. This is used to tell the plugin manager which class in the plugin extends MEPlugin.

**Returns:**

a String containing the path to the plugin class.

### getPluginId

```
public abstract String getPluginId()
```

Returns this plugin's plugin-id. The value returned should be the same statically declared value returned by the plugin's MEPlugin#getPluginId and the model's AbstractModel#getPluginId.

**Returns:**

a String containing the plugin-id

### getVersion

```
public abstract String getVersion()
```

Returns the current plugin version. This is used by the plugin manager to determine whether a dependent plugin will work with this version.

**Returns:**

a String containing the current version of this plugin.

### getPluginPrereqs

```
public abstract String[] getPluginPrereqs()
```

Returns the names and versions of the plugins this plugin depends upon. Each dependency is formatted as: [plugin id]:[version] where the version is considered the minimum acceptable version. For example: RELAP5:0.1.0

**Returns:**

a String[] containing all the plugins this depends upon.

### getClassPrereqs

```
public abstract String[] getClassPrereqs()
```

Returns the name of any specific classes required by this plugin that are not included in the main ModelEditor distribution, the standard Java distribution or this plugin.

**Returns:**

a String[] containing the full path of each Class required by this plugin.

### loadPlugin

```
public abstract MEPlugin loadPlugin()
```

Creates an instance of the MEPlugin that is described by this MEPluginData object. Any initialization required by MEPlugin subclasses (except for MEPlugin#loadMainMenuItems and MEPlugin#loadSettings which are called after the plugin is loaded), should be done at this time.

**Returns:**

(continued from last page)

an MEPlugin instance loaded with this MEPluginData's classloader.



---

## Package

# com.cafean.Number

Provides classes for handling various types of numbers as mutable wrappers. Wrappers are provided for [real](#) numbers as a double precision float as well as [integers](#) and `booleans`.

Each class handles conversion between SI and English units and stores all data in SI units internally.

Various number related utility classes have been provided, such as [com.cafean.Number.IntegerArray](#).

## com.cafean.Number Class Angle

```

java.lang.Object
  |
+-com.cafean.Number.BaseNumber
  |
+-com.cafean.Number.Real
  |
+-com.cafean.Number.Angle

```

```

public class Angle
extends Real

```

A Real derivative for storing angle values. Stores radians internally and degrees externally.

English Units: degrees, SI Units: degrees.

### Field Summary

|                     |                                  |
|---------------------|----------------------------------|
| static final double | halfPI                           |
|                     | Value: <b>1.5707963267948966</b> |

Fields inherited from class [com.cafean.Number.Real](#)

[Unknown](#)

Fields inherited from class [com.cafean.Number.BaseNumber](#)

[BRITISH](#), [SI](#), [UNIT\\_NAMES](#)

### Constructor Summary

|        |                 |  |
|--------|-----------------|--|
| public | Angle()         | Creates a new instance with an unknown value.  |
| public | Angle(double d) | Creates a new instance initialized to a value. |

### Method Summary

|      |                   |                 |
|------|-------------------|-----------------|
| void | convert(double d) | { @inheritDoc } |
| void | convert(Number n) | { @inheritDoc } |

|                         |  |
|-------------------------|--|
| double                  | getConversionFactor()<br>{ @inheritDoc}                                |
| double                  | getDisplayValue(int unittypes)   |
| String                  | getENG_Units()<br>{ @inheritDoc}                                       |
| java.text.DecimalFormat | getFormat(double value)<br>{ @inheritDoc}                              |
| String                  | getSI_Units()<br>{ @inheritDoc}  |
| boolean                 | isMostlyVerticalDown()<br>Return true if the angle is < -45 degrees.   |
| boolean                 | isMostlyVerticalUp()<br>Return true if the angle is > 45 degrees.      |
| boolean                 | isVerticalDown()<br>Return true if the angle is close to -180 degrees. |
| boolean                 | isVerticalUp()<br>Return true if the angle is close to 180 degrees.    |

#### Methods inherited from class [com.cafean.Number.Real](#)

[abs](#), [add](#), [add](#), [arraysEqual](#), [clone](#), [compareTo](#), [convert](#), [convert](#), [convert](#), [convert](#), [convert](#), [divide](#), [divide](#), [divideby](#), [divideby](#), [equals](#), [equals](#), [equals](#), [equals](#), [equals](#), [getConversionFactor](#), [getCurrentDisplayValue](#), [getDisplayName](#), [getDoubleValue](#), [getENG\\_Units](#), [getFormat](#), [getHtmlUnits](#), [getHtmlUnits](#), [getLoadFormat](#), [getName](#), [getReferencedValue](#), [getSI\\_Units](#), [getSignificantFigs](#), [getStrippedValue](#), [getTransferData](#), [getTransferDataFlavors](#), [getUnitName](#), [getUnits](#), [getValue](#), [greaterthan](#), [greaterthan](#), [isDataFlavorSupported](#), [isKnown](#), [isUnknown](#), [lessthan](#), [lessthan](#), [lockToString](#), [multiply](#), [multiply](#), [normalizeArray](#), [restoreState](#), [setSignificantFigs](#), [setUnknown](#), [setValue](#), [setValue](#), [setValue](#), [setValue](#), [sortParallelRealArrays](#), [sqrt](#), [storeState](#), [subtract](#), [subtract](#), [toLoadString](#), [toLoadString](#), [toString](#), [toString](#), [unlockToString](#)

#### Methods inherited from class [com.cafean.Number.BaseNumber](#)

[clone](#), [compareDouble](#), [convert](#), [getENG\\_Units](#), [getFormat](#), [getSI\\_Units](#), [getUnits](#), [getUnitType](#), [isKnown](#), [isUnknown](#), [setKnown](#), [setUnitType](#), [setUnknown](#), [setValue](#), [setValue](#), [toString](#)

#### Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Fields

(continued from last page)

## halfPI

```
public static final double halfPI
```

## Constructors

### Angle

```
public Angle()
```

Creates a new instance with an unknown value.

### Angle

```
public Angle(double d)
```

Creates a new instance initialized to a value.

#### Parameters:

d - The initial value.

## Methods

### getConversionFactor

```
public double getConversionFactor()
```

Retrieves the conversion factor used to convert this Real's value from SI to British units.

### getSI\_Units

```
public String getSI_Units()
```

Retrieves a String representation of this Real's SI unit type.

### getENG\_Units

```
public String getENG_Units()
```

Retrieves a String representation of this Real's English unit type.

### convert

```
public void convert(Number n)
```

Converts the doubleValue of the Number given to the global units. The result of the conversion is set as the current value of this Real.

### convert

```
public void convert(double d)
```

Converts the double given into the global units, and sets the result as the value of this Real.

### getFormat

```
public java.text.DecimalFormat getFormat(double value)
```

---

## getDisplayValue

```
public double getDisplayValue(int unittypes)
```

---

## isVerticalUp

```
public boolean isVerticalUp()
```

Return true if the angle is close to 180 degrees.

---

## isVerticalDown

```
public boolean isVerticalDown()
```

Return true if the angle is close to -180 degrees.

---

## isMostlyVerticalDown

```
public boolean isMostlyVerticalDown()
```

Return true if the angle is < -45 degrees.

---

## isMostlyVerticalUp

```
public boolean isMostlyVerticalUp()
```

Return true if the angle is > 45 degrees.

---

## com.cafean.Number Class BaseNumber

java.lang.Object

└─com.cafean.Number.BaseNumber

### All Implemented Interfaces:

Cloneable

### Direct Known Subclasses:

[Real](#), [Int](#)

public abstract class **BaseNumber**

extends Object

implements Cloneable

BaseNumber is the abstract class used to represent all numbers in the ModelEditor.

### Field Summary

|                       |  |
|-----------------------|--|
| static final int      | BRITISH<br>The enumeration value for British units.<br>Value: <b>1</b> |
| static final int      | SI<br>The enumeration value for SI units.<br>Value: <b>0</b>           |
| static final String[] | UNIT_NAMES<br>A list of supported unit Types and their labels.         |

### Constructor Summary

|        |   |
|--------|---|
| public | BaseNumber()<br>Creates a new instance of a BaseNumber. |
|--------|---|

### Method Summary

|                |  |
|----------------|--|
| Object         | clone()<br>{ @inheritDoc }   |
| static boolean | compareDouble(double a,double b,double eps)<br>Compares two doubles to see if they are within epsilon of each other. |
| abstract void  | convert(Number n)<br>convert set the object to a value using the current unit type conversion.                       |

|                                     |  |
|-------------------------------------|--|
| abstract String                     | getENG_Units()<br>Return the string used to represent BRITISH units.                       |
| abstract<br>java.text.DecimalFormat | getFormat(double value)<br>Return the format used to represent a value.                    |
| abstract String                     | getSI_Units()<br>Return the string used to represent SI units.                             |
| String                              | getUnits()<br>Retrieves a string representation of the current units.                      |
| static int                          | getUnitType()<br>Access routine for the unit type.   |
| boolean                             | isKnown()<br>Return true if the value is known, otherwise return false.                    |
| boolean                             | isUnknown()  |
| void                                | setKnown()<br>Set the unknown flag to false.   |
| static void                         | setUnitType(int t)<br>Set the unit type.   |
| void                                | setUnknown()<br>Set the unknown flag to true.  |
| abstract void                       | setValue(BaseNumber n)<br>setValue sets the object to a value from another BaseNumber.     |
| abstract void                       | setValue(Number n)<br>setValue sets the object to a value without unit type conversion.    |
| abstract String                     | toString()<br>toString translates the internal value to a string in the current unit type. |

**Methods inherited from class** java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Fields

### UNIT\_NAMES

public static final java.lang.String **UNIT\_NAMES**

A list of supported unit Types and their labels.

(continued from last page)

## SI

```
public static final int SI
```

The enumeration value for SI units.

## BRITISH

```
public static final int BRITISH
```

The enumeration value for British units.

## Constructors

### BaseNumber

```
public BaseNumber()
```

Creates a new instance of a BaseNumber.

## Methods

### getUnitType

```
public static int getUnitType()
```

Access routine for the unit type.

**Returns:**

SI or BRITISH.

### setUnitType

```
public static void setUnitType(int t)
```

Set the unit type.

**Parameters:**

t - SI or BRITISH.

### getSI\_Units

```
public abstract String getSI_Units()
```

Return the string used to represent SI units.

**Returns:**

The SI unit label.

### getENG\_Units

```
public abstract String getENG_Units()
```

Return the string used to represent BRITISH units.

**Returns:**

The BRITISH unit label.

### setUnknown

```
public void setUnknown()
```



(continued from last page)

Set the unknown flag to true.

---

## setKnown

```
public void setKnown()
```

Set the unknown flag to false.

---

## isUnknown

```
public boolean isUnknown()
```

---

## isKnown

```
public boolean isKnown()
```

Return true if the value is known, otherwise return false.

### Returns:

true if the value is known, otherwise return false.

---

## toString

```
public abstract String toString()
```

toString translates the internal value to a string in the current unit type.

---

## setValue

```
public abstract void setValue(Number n)
```

setValue sets the object to a value without unit type conversion.

---

## setValue

```
public abstract void setValue(BaseNumber n)
```

setValue sets the object to a value from another BaseNumber.

### Parameters:

n - The base number that the value will be copied from.

---

## convert

```
public abstract void convert(Number n)
```

convert set the object to a value using the current unit type conversion.

---

## getFormat

```
public abstract java.text.DecimalFormat getFormat(double value)
```

Return the format used to represent a value.

### Parameters:

value - The number to be formatted.

### Returns:

The format.

(continued from last page)

## **clone**

```
public Object clone()
```

---

## **getUnits**

```
public String getUnits()
```

Retrieves a string representation of the current units.

---

## **compareDouble**

```
public static boolean compareDouble(double a,  
    double b,  
    double eps)
```

Compares two doubles to see if they are within epsilon of each other.

### **Parameters:**

- a - the first number
- b - the second number
- eps - epsilon

### **Returns:**

true if a and b are within eps of each other, false otherwise.

## com.cafean.Number Class Dimless

```

java.lang.Object
  +- com.cafean.Number.BaseNumber
    +- com.cafean.Number.Real
      +- com.cafean.Number.Dimless
  
```

public class **Dimless**  
 extends [Real](#)

A mutable dimensionless value.

Fields inherited from class [com.cafean.Number.Real](#)

[Unknown](#)

Fields inherited from class [com.cafean.Number.BaseNumber](#)

[BRITISH](#), [SI](#), [UNIT\\_NAMES](#)

### Constructor Summary

|        |   |
|--------|---|
| public | Dimless()<br>Creates a new instance with an unknown value.          |
| public | Dimless(double d)<br>Creates a new instance initialized to a value. |

### Method Summary

|                                  |  |
|----------------------------------|--|
| double                           | getConversionFactor()<br>{ @inheritDoc}  |
| String                           | getENG_Units()<br>{ @inheritDoc}   |
| String                           | getSI_Units()<br>{ @inheritDoc}  |
| static <a href="#">Dimless[]</a> | makeArray(double[] dArray)<br>Create an array of Dimless objects from an array of doubles. |
| static <a href="#">Dimless[]</a> | makeArray(Vector vector)<br>Create an array of Dimless objects from a vector of Dimless.   |

Methods inherited from class [com.cafean.Number.Real](#)

[abs](#), [add](#), [add](#), [arraysEqual](#), [clone](#), [compareTo](#), [convert](#), [convert](#), [convert](#), [convert](#), [convert](#), [divide](#), [divide](#), [divide](#), [divideby](#), [divideby](#), [equals](#), [equals](#), [equals](#), [equals](#), [equals](#), [getConversionFactor](#), [getCurrentDisplayValue](#), [getDisplayName](#), [getDoubleValue](#), [getENG\\_Units](#), [getFormat](#), [getHtmlUnits](#), [getHtmlUnits](#), [getLoadFormat](#), [getName](#), [getReferencedValue](#), [getSI\\_Units](#), [getSignificantFigs](#), [getStrippedValue](#), [getTransferData](#), [getTransferDataFlavors](#), [getUnitName](#), [getUnits](#), [getValue](#), [greaterthan](#), [greaterthan](#), [isDataFlavorSupported](#), [isKnown](#), [isUnknown](#), [lessthan](#), [lessthan](#), [lockToString](#), [multiply](#), [multiply](#), [normalizeArray](#), [restoreState](#), [setSignificantFigs](#), [setUnknown](#), [setValue](#), [setValue](#), [setValue](#), [setValue](#), [sortParallelRealArrays](#), [sqrt](#), [storeState](#), [subtract](#), [subtract](#), [toLoadString](#), [toLoadString](#), [toString](#), [toString](#), [unlockToString](#)

Methods inherited from class [com.cafean.Number.BaseNumber](#)

[clone](#), [compareDouble](#), [convert](#), [getENG\\_Units](#), [getFormat](#), [getSI\\_Units](#), [getUnits](#), [getUnitType](#), [isKnown](#), [isUnknown](#), [setKnown](#), [setUnitType](#), [setUnknown](#), [setValue](#), [setValue](#), [toString](#)

Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### Dimless

```
public Dimless()
```

Creates a new instance with an unknown value.

### Dimless

```
public Dimless(double d)
```

Creates a new instance initialized to a value.

#### Parameters:

d – The initial value.

## Methods

### getConversionFactor

```
public double getConversionFactor()
```

Retrieves the conversion factor used to convert this Real's value from SI to British units.

### getSI\_Units

```
public String getSI_Units()
```

Retrieves a String representation of this Real's SI unit type.

### getENG\_Units

```
public String getENG_Units()
```

Retrieves a String representation of this Real's English unit type.

## makeArray

```
public static Dimless\[\] makeArray(double[] dArray)
```

Create an array of Dimless objects from an array of doubles.

**Parameters:**

dArray - The array of doubles.

**Returns:**

An array of Dimless initialized to the values in dArray.

---

## makeArray

```
public static Dimless\[\] makeArray(Vector vector)
```

Create an array of Dimless objects from a vector of Dimless.

**Parameters:**

vector - The vector of Dimless objects.

**Returns:**

An array of Dimless initialized to the values in vector.

---

## com.cafean.Number Class DoubleArray

java.lang.Object

└─com.cafean.Number.DoubleArray

### All Implemented Interfaces:

Cloneable

```
public class DoubleArray
extends Object
implements Cloneable
```

An unsynchronized dynamically sized array of double.

### Constructor Summary

|        |   |
|--------|---|
| public | DoubleArray(int capacity)<br>Creates a new DoubleArray with the given initial capacity. |
| public | DoubleArray(double[] array)   |

### Method Summary

|         |  |
|---------|--|
| void    | add(double value)<br>Appends the given double to the DoubleArray's data.                                     |
| void    | add(double[] values)<br>Appends the given data to the DoubleArray's data.                                    |
| int     | capacity()<br>Returns this DoubleArray's current capacity.   |
| void    | clear()<br>Sets this DoubleArray empty.  |
| Object  | clone()  |
| boolean | contains(double value)<br>Returns true if the given value is contained in this DoubleArray.                  |
| double  | get(int index)<br>Returns the value of the double at the given index.  |
| int     | indexOf(double value)<br>Retrieves the index of the first occurrence of the given value in this DoubleArray. |

|             |  |
|-------------|--|
| static void | main(String[] args)<br>Performs a thorough test of the DoubleArray's functionality.            |
| void        | remove(int index)<br>Removes the double at the given index. NOTE: O(n) complexity.             |
| void        | set(int index,double value)<br>Sets the given value at the given index.                        |
| int         | size()<br>Returns the number of ints stored in this DoubleArray                                |
| void        | sort()<br>Sorts this DoubleArray via <code>java.util.Arrays.sort()</code> .                    |
| double[]    | toArray()<br>Returns an double array containing only the used elements of this DoubleArray.    |
| void        | trimToSize()<br>Trims the capacity of this DoubleArray instance to be the list's current size. |

#### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### DoubleArray

public **DoubleArray**(int capacity)

Creates a new DoubleArray with the given initial capacity.

### DoubleArray

public **DoubleArray**(double[] array)

## Methods

### clear

public void **clear**()

Sets this DoubleArray empty.

### clone

public Object **clone**()

(continued from last page)

---

## contains

```
public boolean contains(double value)
```

Returns true if the given value is contained in this DoubleArray.

---

## add

```
public void add(double value)
```

Appends the given double to the DoubleArray's data. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## add

```
public void add(double[] values)
```

Appends the given data to the DoubleArray's data. If size is currently 0, the values given are cloned. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## main

```
public static void main(String[] args)
```

Performs a thorough test of the DoubleArray's functionality.

---

## indexOf

```
public int indexOf(double value)
```

Retrieves the index of the first occurrence of the given value in this DoubleArray.

---

## remove

```
public void remove(int index)
```

Removes the double at the given index. NOTE: O(n) complexity.

---

## get

```
public double get(int index)
```

Returns the value of the double at the given index. NOTE: O(1) complexity.

---

## set

```
public void set(int index,  
               double value)
```

Sets the given value at the given index.

---

## size

```
public final int size()
```

Returns the number of ints stored in this DoubleArray

---

## capacity

```
public final int capacity()
```

Returns this DoubleArray's current capacity.

---



(continued from last page)

## **sort**

```
public void sort()
```

Sorts this DoubleArray via `java.util.Arrays.sort()`. First calls `trimToSize()`.

---

## **trimToSize**

```
public void trimToSize()
```

Trims the capacity of this DoubleArray instance to be the list's current size. An application can use this operation to minimize the storage of an DoubleArray instance.

---

## **toArray**

```
public double[] toArray()
```

Returns an double array containing only the used elements of this DoubleArray. NOTE: This is a defensive copy, even if sizes match.

## com.cafean.Number Class FloatArray

java.lang.Object

└─com.cafean.Number.FloatArray

### All Implemented Interfaces:

Cloneable

```
public class FloatArray
  extends Object
  implements Cloneable
```

An unsynchronized dynamically sized array of float.

### Constructor Summary

|        |  |
|--------|--|
| public | <code>FloatArray(int capacity)</code><br>Creates a new FloatArray with the given initial capacity. |
| public | <code>FloatArray(float[] array)</code>   |

### Method Summary

|         |   |
|---------|---|
| void    | <code>add(float value)</code><br>Appends the given float to the FloatArray's data.                                      |
| void    | <code>add(float[] values)</code><br>Appends the given data to the FloatArray's data.                                    |
| int     | <code>capacity()</code><br>Returns this FloatArray's current capacity.  |
| void    | <code>clear()</code><br>Sets this FloatArray empty.   |
| Object  | <code>clone()</code>  |
| boolean | <code>contains(float value)</code><br>Returns true if the given value is contained in this FloatArray.                  |
| float   | <code>get(int index)</code><br>Returns the value of the float at the given index.                                       |
| int     | <code>indexOf(float value)</code><br>Retrieves the index of the first occurrence of the given value in this FloatArray. |

|             |   |
|-------------|---|
| static void | main(String[] args)<br>Performs a thorough test of the FloatArray's functionality.            |
| void        | remove(int index)<br>Removes the float at the given index. NOTE: O(n) complexity.             |
| void        | set(int index, float value)<br>Sets the given value at the given index.                       |
| int         | size()<br>Returns the number of ints stored in this FloatArray                                |
| void        | sort()<br>Sorts this FloatArray via <code>java.util.Arrays.sort()</code> .                    |
| float[]     | toArray()<br>Returns an float array containing only the used elements of this FloatArray.     |
| void        | trimToSize()<br>Trims the capacity of this FloatArray instance to be the list's current size. |

#### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### FloatArray

`public FloatArray(int capacity)`  
Creates a new FloatArray with the given initial capacity.

### FloatArray

`public FloatArray(float[] array)`

## Methods

### clear

`public void clear()`  
Sets this FloatArray empty.

### clone

`public Object clone()`

(continued from last page)

---

## contains

```
public boolean contains(float value)
```

Returns true if the given value is contained in this FloatArray.

---

## add

```
public void add(float value)
```

Appends the given float to the FloatArray's data. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## add

```
public void add(float[] values)
```

Appends the given data to the FloatArray's data. If size is currently 0, the values given are cloned. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## main

```
public static void main(String[] args)
```

Performs a thorough test of the FloatArray's functionality.

---

## indexOf

```
public int indexOf(float value)
```

Retrieves the index of the first occurrence of the given value in this FloatArray.

---

## remove

```
public void remove(int index)
```

Removes the float at the given index. NOTE: O(n) complexity.

---

## get

```
public float get(int index)
```

Returns the value of the float at the given index. NOTE: O(1) complexity.

---

## set

```
public void set(int index,  
               float value)
```

Sets the given value at the given index.

---

## size

```
public final int size()
```

Returns the number of ints stored in this FloatArray

---

## capacity

```
public final int capacity()
```

Returns this FloatArray's current capacity.

---

(continued from last page)

## **sort**

```
public void sort()
```

Sorts this FloatArray via `java.util.Arrays.sort()`. First calls `trimToSize()`.

---

## **trimToSize**

```
public void trimToSize()
```

Trims the capacity of this FloatArray instance to be the list's current size. An application can use this operation to minimize the storage of an FloatArray instance.

---

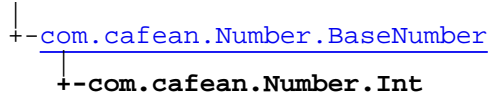
## **toArray**

```
public float[] toArray()
```

Returns an float array containing only the used elements of this FloatArray. NOTE: This is a defensive copy, even if sizes match.

## com.cafean.Number Class Int

java.lang.Object



**All Implemented Interfaces:**  
Comparable, Cloneable

public class **Int**  
extends [BaseNumber](#)  
implements Cloneable, Comparable

A mutable wrapper for an int.

### Field Summary

|                  |                                    |
|------------------|------------------------------------|
| static final int | Unknown<br>Value: <b>-99119900</b> |
| int              | value                              |

Fields inherited from class [com.cafean.Number.BaseNumber](#)

[BRITISH](#), [SI](#), [UNIT\\_NAMES](#)

### Constructor Summary

|        |               |
|--------|---------------|
| public | Int()         |
| public | Int(int aInt) |
| public | Int(Int aInt) |

### Method Summary

|                     |                     |
|---------------------|---------------------|
| <a href="#">Int</a> | add(int x)          |
| <a href="#">Int</a> | add(Int aInt)       |
| int                 | compareTo(Object o) |
| void                | convert(Long n)     |

|                         |                          |
|-------------------------|--------------------------|
| void                    | convert(Number n)        |
| void                    | convert(String s)        |
| <a href="#">Int</a>     | divide(Int aInt)         |
| boolean                 | equals(int aInt)         |
| boolean                 | equals(Int aInt)         |
| boolean                 | equals(Object o)         |
| int                     | getConversionFactor()    |
| String                  | getENG_Units()           |
| java.text.DecimalFormat | getFormat()              |
| java.text.DecimalFormat | getFormat(double d)      |
| String                  | getSI_Units()            |
| int                     | getValue()               |
| boolean                 | greaterthan(int aInt)    |
| boolean                 | greaterthan(Int aInt)    |
| int                     | hashCode()               |
| boolean                 | lessthan(int aInt)       |
| boolean                 | lessthan(Int aInt)       |
| <a href="#">Int</a>     | multiply(Int aInt)       |
| void                    | setValue(BaseNumber a)   |
| void                    | setValue(int aInt)       |
| void                    | setValue(Number n)       |
| void                    | setValue(String aString) |
| <a href="#">Int</a>     | sqrt()                   |
| <a href="#">Int</a>     | subtract(int x)          |

|                     |                    |
|---------------------|--------------------|
| <a href="#">Int</a> | subtract(Int aInt) |
| String              | toString()         |

**Methods inherited from class** [com.cafean.Number.BaseNumber](#)

[clone](#), [compareDouble](#), [convert](#), [getENG\\_Units](#), [getFormat](#), [getSI\\_Units](#), [getUnits](#), [getUnitType](#), [isKnown](#), [isUnknown](#), [setKnown](#), [setUnitType](#), [setUnknown](#), [setValue](#), [setValue](#), [toString](#)

**Methods inherited from class** java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Fields

**value**

```
public int value
```

**Unknown**

```
public static final int Unknown
```

## Constructors

**Int**

```
public Int()
```

**Int**

```
public Int(int aInt)
```

**Int**

```
public Int(Int aInt)
```

## Methods

**getConversionFactor**

```
public int getConversionFactor()
```



---

## getSI\_Units

```
public String getSI_Units()
```

---

## getENG\_Units

```
public String getENG_Units()
```

---

## setValue

```
public void setValue(BaseNumber a)
```

---

## setValue

```
public void setValue(String aString)
```

---

## setValue

```
public void setValue(int aInt)
```

---

## setValue

```
public void setValue(Number n)
```

---

## getValue

```
public int getValue()
```

---

## convert

```
public void convert(Number n)
```

---

## convert

```
public void convert(Long n)
```

---

## convert

```
public void convert(String s)
```

---

(continued from last page)

---

## toString

```
public String toString()
```

---

## getFormat

```
public java.text.DecimalFormat getFormat(double d)
```

---

## getFormat

```
public java.text.DecimalFormat getFormat()
```

---

## hashCode

```
public int hashCode()
```

---

## compareTo

```
public int compareTo(Object o)
```

---

## equals

```
public boolean equals(int aInt)
```

---

## equals

```
public boolean equals(Int aInt)
```

---

## equals

```
public boolean equals(Object o)
```

---

## lessthan

```
public boolean lessthan(Int aInt)
```

---

## lessthan

```
public boolean lessthan(int aInt)
```

---

## greaterthan

```
public boolean greaterthan(Int aInt)
```

---

**greaterthan**

```
public boolean greaterthan(int aInt)
```

---

**add**

```
public Int add(Int aInt)
```

---

**add**

```
public Int add(int x)
```

---

**subtract**

```
public Int subtract(Int aInt)
```

---

**subtract**

```
public Int subtract(int x)
```

---

**multiply**

```
public Int multiply(Int aInt)
```

---

**divide**

```
public Int divide(Int aInt)
```

---

**sqrt**

```
public Int sqrt()
```

---

## com.cafean.Number Class IntegerArray

java.lang.Object

└─com.cafean.Number.IntegerArray

### All Implemented Interfaces:

Cloneable

```
public class IntegerArray
  extends Object
  implements Cloneable
```

An unsynchronized dynamically sized array of int.

### Constructor Summary

|        |  |
|--------|--|
| public | <code>IntegerArray(int capacity)</code><br>Creates a new IntegerArray with the given initial capacity. |
| public | <code>IntegerArray(int[] array)</code>   |

### Method Summary

|         |   |
|---------|---|
| void    | <code>add(int value)</code><br>Appends the given int to the IntegerArray's data.  |
| void    | <code>add(int[] values)</code><br>Appends the given data to the IntegerArray's data.                                    |
| int     | <code>capacity()</code><br>Returns this IntegerArray's current capacity.  |
| void    | <code>clear()</code><br>Sets this IntegerArray empty.   |
| Object  | <code>clone()</code>  |
| boolean | <code>contains(int value)</code><br>Returns true if the given value is contained in this IntegerArray.                  |
| int     | <code>get(int index)</code><br>Returns the value of the int at the given index.   |
| int     | <code>indexOf(int value)</code><br>Retrieves the index of the first occurrence of the given value in this IntegerArray. |

|             |   |
|-------------|---|
| static void | main(String[] args)<br>Performs a thorough test of the IntegerArray's functionality.            |
| void        | remove(int index)<br>Removes the int at the given index. NOTE: O(n) complexity.                 |
| void        | set(int index,int value)<br>Sets the given value at the given index.                            |
| int         | size()<br>Returns the number of ints stored in this IntegerArray                                |
| void        | sort()<br>Sorts this IntegerArray via java.util.Arrays.sort().                                  |
| int[]       | toArray()<br>Returns an int array containing only the used elements of this IntegerArray.       |
| void        | trimToSize()<br>Trims the capacity of this IntegerArray instance to be the list's current size. |

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### IntegerArray

public **IntegerArray**(int capacity)  
Creates a new IntegerArray with the given initial capacity.

### IntegerArray

public **IntegerArray**(int[] array)

## Methods

### clear

public void **clear**()  
Sets this IntegerArray empty.

### clone

public Object **clone**()

(continued from last page)

---

## contains

```
public boolean contains(int value)
```

Returns true if the given value is contained in this IntegerArray.

---

## add

```
public void add(int value)
```

Appends the given int to the IntegerArray's data. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## add

```
public void add(int[] values)
```

Appends the given data to the IntegerArray's data. If size is currently 0, the values given are cloned. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## main

```
public static void main(String[] args)
```

Performs a thorough test of the IntegerArray's functionality.

---

## indexOf

```
public int indexOf(int value)
```

Retrieves the index of the first occurrence of the given value in this IntegerArray.

---

## remove

```
public void remove(int index)
```

Removes the int at the given index. NOTE: O(n) complexity.

---

## get

```
public int get(int index)
```

Returns the value of the int at the given index. NOTE: O(1) complexity.

---

## set

```
public void set(int index,  
               int value)
```

Sets the given value at the given index.

---

## size

```
public final int size()
```

Returns the number of ints stored in this IntegerArray

---

## capacity

```
public final int capacity()
```

Returns this IntegerArray's current capacity.

---

(continued from last page)

## **sort**

```
public void sort()
```

Sorts this IntegerArray via `java.util.Arrays.sort()`. First calls `trimToSize()`.

---

## **trimToSize**

```
public void trimToSize()
```

Trims the capacity of this IntegerArray instance to be the list's current size. An application can use this operation to minimize the storage of an IntegerArray instance.

---

## **toArray**

```
public int[] toArray()
```

Returns an int array containing only the used elements of this IntegerArray. NOTE: This is a defensive copy, even if sizes match.

## com.cafean.Number Class LongArray

java.lang.Object

└─com.cafean.Number.LongArray

### All Implemented Interfaces:

Cloneable

```
public class LongArray
extends Object
implements Cloneable
```

An unsynchronized dynamically sized array of long.

### Constructor Summary

|        |   |
|--------|---|
| public | LongArray(int capacity)<br>Creates a new LongArray with the given initial capacity. |
| public | LongArray(long[] array)   |

### Method Summary

|         |  |
|---------|--|
| void    | add(long value)<br>Appends the given long to the LongArray's data.                                       |
| void    | add(long[] values)<br>Appends the given data to the LongArray's data.                                    |
| int     | capacity()<br>Returns this LongArray's current capacity.   |
| void    | clear()<br>Sets this LongArray empty.  |
| Object  | clone()  |
| boolean | contains(long value)<br>Returns true if the given value is contained in this LongArray.                  |
| long    | get(int index)<br>Returns the value of the long at the given index.                                      |
| int     | indexOf(long value)<br>Retrieves the index of the first occurrence of the given value in this LongArray. |



|             |  |
|-------------|--|
| static void | main(String[] args)<br>Performs a thorough test of the LongArray's functionality.            |
| void        | remove(int index)<br>Removes the long at the given index. NOTE: O(n) complexity.             |
| void        | set(int index, long value)<br>Sets the given value at the given index.                       |
| int         | size()<br>Returns the number of ints stored in this LongArray                                |
| void        | sort()<br>Sorts this LongArray via <code>java.util.Arrays.sort()</code> .                    |
| long[]      | toArray()<br>Returns an long array containing only the used elements of this LongArray.      |
| void        | trimToSize()<br>Trims the capacity of this LongArray instance to be the list's current size. |

#### Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### LongArray

`public LongArray(int capacity)`  
Creates a new LongArray with the given initial capacity.

### LongArray

`public LongArray(long[] array)`

## Methods

### clear

`public void clear()`  
Sets this LongArray empty.

### clone

`public Object clone()`

(continued from last page)

## contains

```
public boolean contains(long value)
```

Returns true if the given value is contained in this LongArray.

---

## add

```
public void add(long value)
```

Appends the given long to the LongArray's data. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## add

```
public void add(long[] values)
```

Appends the given data to the LongArray's data. If size is currently 0, the values given are cloned. NOTE: Expected O(1), Worst case O(n) complexity ( resize )

---

## main

```
public static void main(String[] args)
```

Performs a thorough test of the LongArray's functionality.

---

## indexOf

```
public int indexOf(long value)
```

Retrieves the index of the first occurrence of the given value in this LongArray.

---

## remove

```
public void remove(int index)
```

Removes the long at the given index. NOTE: O(n) complexity.

---

## get

```
public long get(int index)
```

Returns the value of the long at the given index. NOTE: O(1) complexity.

---

## set

```
public void set(int index,  
               long value)
```

Sets the given value at the given index.

---

## size

```
public final int size()
```

Returns the number of ints stored in this LongArray

---

## capacity

```
public final int capacity()
```

Returns this LongArray's current capacity.

---

(continued from last page)

## **sort**

```
public void sort()
```

Sorts this LongArray via `java.util.Arrays.sort()`. First calls `trimToSize()`.

---

## **trimToSize**

```
public void trimToSize()
```

Trims the capacity of this LongArray instance to be the list's current size. An application can use this operation to minimize the storage of an LongArray instance.

---

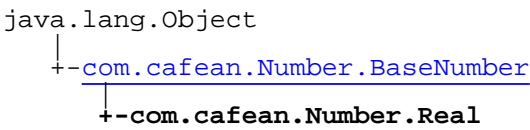
## **toArray**

```
public long[] toArray()
```

Returns an long array containing only the used elements of this LongArray. NOTE: This is a defensive copy, even if sizes match.

# com.cafean.Number

## Class Real



**All Implemented Interfaces:**  
java.awt.datatransfer.Transferable, Comparable, Cloneable, Cloneable

**Direct Known Subclasses:**  
[Time](#), [Dimless](#), [Angle](#)

public class **Real**  
extends [BaseNumber](#)  
implements Cloneable, Cloneable, Comparable, java.awt.datatransfer.Transferable

Real is base class for all ModelEditor floating point numbers.

| Field Summary       |                                      |
|---------------------|--------------------------------------|
| static final double | Unknown<br><br>Value: <b>-1.0E30</b> |

| Fields inherited from class <a href="#">com.cafean.Number.BaseNumber</a>  |
|---|
| <a href="#">BRITISH</a> , <a href="#">SI</a> , <a href="#">UNIT_NAMES</a> |

| Constructor Summary |  |
|---------------------|--|
| public              | Real()<br>Creates a new instance of Real with an unknown value.                |
| public              | Real(double aDouble)<br>Creates a new instance of Real initialized to a value. |
| public              | Real(Real aReal)<br>Creates a new instance of Real by copying another.         |

| Method Summary       |                     |
|----------------------|---------------------|
| <a href="#">Real</a> | abs()               |
| <a href="#">Real</a> | add(double aDouble) |
| <a href="#">Real</a> | add(Real aReal)     |

|                      |   |
|----------------------|---|
| static boolean       | arraysEqual(Real[] a,Real[] b)  |
| Object               | clone()   |
| int                  | compareTo(Object o)   |
| void                 | convert(double d)<br>Converts the double given into the global units, and sets the result as the value of this Real.                        |
| void                 | convert(double d,int unitType)<br>Converts the double given into the units requested, and sets the result as the value for this Real.       |
| void                 | convert(Number n)<br>Converts the doubleValue of the Number given to the global units.  |
| void                 | convert(String aString)<br>Converts the String given into the global units, and sets the result as the value for this Real.                 |
| void                 | convert(String aString,int unitType)<br>Converts the String given into the units requested, and sets the result as the value for this Real. |
| <a href="#">Real</a> | divide(double aDouble)  |
| <a href="#">Real</a> | divide(Real aReal)  |
| <a href="#">Real</a> | divideby(double aDouble)  |
| <a href="#">Real</a> | divideby(Real aReal)  |
| boolean              | equals(double aDouble)  |
| static boolean       | equals(double a,double b)<br>Returns true if the given doubles differ by less than 5E-9.  |
| static boolean       | equals(double a,double b,double delta)<br>Returns true if the given doubles differ by less than the given ( a + b ) * delta.                |
| boolean              | equals(Object o)  |
| boolean              | equals(Real aReal)<br>NOTE: This does not get called instead of Object.equals   |
| double               | getConversionFactor()<br>Retrieves the conversion factor used to convert this Real's value from SI to British units.                        |
| double               | getCurrentDisplayValue(int unittypes)<br>Returns the value to display.  |
| String               | getDisplayName()<br>Retrieves a Display name for this unit.   |

|                                    |  |
|------------------------------------|--|
| double                             | getDoubleValue()<br>Returns the numerical value of this Real in SI units.  |
| String                             | getENG_Units()<br>Retrieves a String representation of this Real's English unit type.  |
| java.text.DecimalFormat            | getFormat(double value)<br>Returns the current global units formatted for HTML   |
| String                             | getHtmlUnits()<br>Returns the units requested formatted for HTML.  |
| String                             | getLoadFormat(double value)<br>Decides what DecimalFormat to use to display the given value in 12 characters.                                    |
| String                             | getName()<br>Returns the name of this Real.  |
| static double                      | getReferencedValue(double value)   |
| String                             | getSI_Units()<br>Retrieves a String representation of this Real's SI unit type.  |
| int                                | getSignificantFigs()<br>Getter for property significantFigs.   |
| static double                      | getStrippedValue(double value)   |
| Object                             | getTransferData(java.awt.datatransfer.DataFlavor flavor)<br>Returns an object which represents the data to be transferred.                       |
| java.awt.datatransfer.DataFlavor[] | getTransferDataFlavors()<br>Returns an array of DataFlavor objects indicating the flavors the data can be provided in.                           |
| String                             | getUnitName()<br>Returns the name of this Real value   |
| String                             | getUnits(int units)<br>Returns this real's units in the given format.  |
| double                             | getValue()<br>Returns the current value of this Real in SI units.  |
| boolean                            | greaterthan(double aDouble)  |
| boolean                            | greaterthan(Real aReal)  |
| boolean                            | isDataFlavorSupported(java.awt.datatransfer.DataFlavor flavor)<br>Returns whether or not the specified data flavor is supported for this object. |

|                      |  |
|----------------------|--|
| boolean              | isKnown()<br>Return true if the value is known, otherwise return false.                                      |
| boolean              | isUnknown()<br>Determines whether this value has been specified to a value or is not a number.               |
| boolean              | lessthan(double aDouble)   |
| boolean              | lessthan(Real aReal)   |
| static void          | lockToString(Thread t)<br>Locks Real's toString() method for use only by the given thread.                   |
| <a href="#">Real</a> | multiply(double aDouble)   |
| <a href="#">Real</a> | multiply(Real aReal)   |
| static void          | normalizeArray(Real[] array)   |
| void                 | restoreState(String prefix,Hashtable state)<br>Restore the state of the Real from an earlier edit.           |
| void                 | setSignificantFigs(int significantFigs)<br>Setter for property significantFigs.                              |
| void                 | setUnknown()   |
| void                 | setValue(BaseNumber a)   |
| void                 | setValue(double aDouble)<br>Sets the current value of this Real.   |
| void                 | setValue(Number n)   |
| void                 | setValue(UserDefinedValue input)<br>Sets the current value to a reference to a given UserDefinedValue.       |
| static void          | sortParallelRealArrays(Real[][] arrays)<br>Sorts the given parallel arrays by the values in the first array. |
| <a href="#">Real</a> | sqrt()   |
| void                 | storeState(String prefix,Hashtable state)<br>Store the state of the Real to permit undo.                     |
| <a href="#">Real</a> | subtract(double aDouble)   |
| <a href="#">Real</a> | subtract(Real aReal)   |

|             |   |
|-------------|---|
| String      | toLoadString()<br>Returns a formatted string for this Real in the current global units that is limited to 12 characters in length.        |
| String      | toLoadString(int unitType)<br>Returns a formatted string for this Real in the requested units that is limited to 12 characters in length. |
| String      | toString()  |
| String      | toString(int unitType)<br>Returns a formatted string containing the current value of this Real.   |
| static void | unlockToString()<br>Unlocks toString() to be used by any thread.  |

#### Methods inherited from class [com.cafean.Number.BaseNumber](#)

[clone](#), [compareDouble](#), [convert](#), [getENG\\_Units](#), [getFormat](#), [getSI\\_Units](#), [getUnits](#), [getUnitType](#), [isKnown](#), [isUnknown](#), [setKnown](#), [setUnitType](#), [setUnknown](#), [setValue](#), [setValue](#), [toString](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Fields

### Unknown

public static final double **Unknown**

## Constructors

### Real

public **Real**()

Creates a new instance of Real with an unknown value.

### Real

public **Real**(double aDouble)

Creates a new instance of Real initialized to a value.

#### Parameters:

aDouble - The floating point value represented by this real.

### Real

public **Real**([Real](#) aReal)



(continued from last page)

Creates a new instance of Real by copying another.

**Parameters:**

aReal - The Real being copied.

## Methods

### lockToString

```
public static void lockToString(Thread t)
```

Locks Real's toString() method for use only by the given thread.

### unlockToString

```
public static void unlockToString()
```

Unlocks toString() to be used by any thread.

### getConversionFactor

```
public double getConversionFactor()
```

Retrieves the conversion factor used to convert this Real's value from SI to British units.

### getDisplayName

```
public String getDisplayName()
```

Retrieves a Display name for this unit.

### getSI\_Units

```
public String getSI_Units()
```

Retrieves a String representation of this Real's SI unit type.

### getENG\_Units

```
public String getENG_Units()
```

Retrieves a String representation of this Real's English unit type.

### getHtmlUnits

```
public String getHtmlUnits()
```

Returns the current global units formatted for HTML

### getHtmlUnits

```
public String getHtmlUnits(int units)
```

Returns the units requested formatted for HTML.

**Parameters:**

units - the unit type requested.

### getUnits

```
public String getUnits(int units)
```

Returns this real's units in the given format.

---

## setValue

```
public void setValue(Number n)
```

---

---

## setValue

```
public void setValue(BaseNumber a)
```

---

---

## setValue

```
public void setValue(double aDouble)
```

Sets the current value of this Real.

---

---

## setValue

```
public void setValue(UserDefinedValue input)
```

Sets the current value to a reference to a given UserDefinedValue.

---

---

## convert

```
public void convert(Number n)
```

Converts the doubleValue of the Number given to the global units. The result of the conversion is set as the current value of this Real.

---

---

## convert

```
public void convert(String aString)
```

Converts the String given into the global units, and sets the result as the value for this Real.

---

---

## convert

```
public void convert(String aString,  
                    int unitType)
```

Converts the String given into the units requested, and sets the result as the value for this Real.

---

---

## convert

```
public void convert(double d)
```

Converts the double given into the global units, and sets the result as the value of this Real.

---

---

## convert

```
public void convert(double d,  
                    int unitType)
```

Converts the double given into the units requested, and sets the result as the value for this Real.

---

---

## getValue

```
public double getValue()
```

Returns the current value of this Real in SI units. If this Real refers to a UserDefinedValue, this will return the reference.

---

---

## getDoubleValue

```
public double getDoubleValue()
```

Returns the numerical value of this Real in SI units. If this Real uses a UserDefinedValue, the numerical value of that UserDefinedValue is returned. This method should be used for mathematical functions.

---

## getFormat

```
public java.text.DecimalFormat getFormat(double value)
```

---

## toString

```
public String toString()
```

---

## getLoadFormat

```
public java.text.DecimalFormat getLoadFormat(double value)
```

Decides what DecimalFormat to use to display the given value in 12 characters.

---

## toLoadString

```
public String toLoadString()
```

Returns a formatted string for this Real in the current global units that is limited to 12 characters in length. The format is chosen so that no numerical precision is lost through this formatting.

---

## toLoadString

```
public String toLoadString(int unitType)
```

Returns a formatted string for this Real in the requested units that is limited to 12 characters in length. The format is chosen so that no numerical precision is lost through this formatting.

---

## getUnitName

```
public String getUnitName()
```

Returns the name of this Real value

---

## toString

```
public String toString(int unitType)
```

Returns a formatted string containing the current value of this Real. If this real uses a user defined numeric, the name of that numeric is returned unless the current mode is MODE\_EXPORT\_ASCII. The units for the formatted string are passed in.

---

## getCurrentDisplayValue

```
public double getCurrentDisplayValue(int unittypes)
```

Returns the value to display. This gets the numerical value and converts it if the given unittype is BRITISH.

---

## clone

```
public Object clone()
```

---

---

## setUnknown

```
public void setUnknown()
```

---

---

## compareTo

```
public int compareTo(Object o)
```

---

---

## equals

```
public boolean equals(Object o)
```

---

---

## equals

```
public boolean equals(Real aReal)
```

NOTE: This does not get called instead of Object.equals

---

---

## equals

```
public boolean equals(double aDouble)
```

---

---

## equals

```
public static boolean equals(double a,  
                             double b)
```

Returns true if the given doubles differ by less than 5E-9.

---

---

## equals

```
public static boolean equals(double a,  
                             double b,  
                             double delta)
```

Returns true if the given doubles differ by less than the given  $(a + b) * \text{delta}$ .

---

---

## lessthan

```
public boolean lessthan(Real aReal)
```

---

---

## lessthan

```
public boolean lessthan(double aDouble)
```

---

---

## greaterthan

```
public boolean greaterthan(Real aReal)
```

---

## greaterthan

```
public boolean greaterthan(double aDouble)
```

---

## arraysEqual

```
public static boolean arraysEqual(Real[] a,  
    Real[] b)
```

---

## add

```
public Real add(Real aReal)
```

---

## add

```
public Real add(double aDouble)
```

---

## subtract

```
public Real subtract(Real aReal)
```

---

## subtract

```
public Real subtract(double aDouble)
```

---

## multiply

```
public Real multiply(Real aReal)
```

---

## multiply

```
public Real multiply(double aDouble)
```

---

## divideby

```
public Real divideby(Real aReal)
```

---

## divide

```
public Real divide(Real aReal)
```

---

(continued from last page)

---

## divideby

```
public Real divideby(double aDouble)
```

---

## divide

```
public Real divide(double aDouble)
```

---

## sqrt

```
public Real sqrt()
```

---

## abs

```
public Real abs()
```

---

## getStrippedValue

```
public static double getStrippedValue(double value)
```

---

## getReferencedValue

```
public static double getReferencedValue(double value)
```

---

## getTransferDataFlavors

```
public java.awt.datatransfer.DataFlavor[] getTransferDataFlavors()
```

Returns an array of DataFlavor objects indicating the flavors the data can be provided in. The array should be ordered according to preference for providing the data (from most richly descriptive to least descriptive).

**Returns:**

an array of data flavors in which this data can be transferred

---

## isDataFlavorSupported

```
public boolean isDataFlavorSupported(java.awt.datatransfer.DataFlavor flavor)
```

Returns whether or not the specified data flavor is supported for this object.

**Parameters:**

`flavor` - the requested flavor for the data

**Returns:**

boolean indicating whether or not the data flavor is supported

---

## getTransferData

```
public Object getTransferData(java.awt.datatransfer.DataFlavor flavor)
    throws java.awt.datatransfer.UnsupportedFlavorException,
           java.io.IOException
```

---

(continued from last page)

Returns an object which represents the data to be transferred. The class of the object returned is defined by the representation class of the flavor.

**Parameters:**

`flavor` - the requested flavor for the data

**Throws:**

`IOException` - if the data is no longer available in the requested flavor.

`UnsupportedFlavorException` - if the requested data flavor is not supported.

**See Also:**

`DataFlavor.getRepresentationClass()`

---

## getSignificantFigs

```
public int getSignificantFigs()
```

Getter for property significantFigs.

**Returns:**

Value of property significantFigs.

---

## setSignificantFigs

```
public void setSignificantFigs(int significantFigs)
```

Setter for property significantFigs.

**Parameters:**

`significantFigs` - New value of property significantFigs.

---

## sortParallelRealArrays

```
public static void sortParallelRealArrays(Real[][] arrays)
```

Sorts the given parallel arrays by the values in the first array.

---

## normalizeArray

```
public static void normalizeArray(Real[] array)
```

---

## isUnknown

```
public boolean isUnknown()
```

Determines whether this value has been specified to a value or is not a number.

---

## isKnown

```
public boolean isKnown()
```

Return true if the value is known, otherwise return false.

**Returns:**

true if the value is known, otherwise return false.

---

## storeState

```
public void storeState(String prefix,  
    Hashtable state)
```

(continued from last page)

Store the state of the Real to permit undo.

**Parameters:**

`state` - a Hashtable containing modified parameters.  
`prefix` - a String containing the prefix for hash entries.

---

**restoreState**

```
public void restoreState(String prefix,  
                        Hashtable state)
```

Restore the state of the Real from an earlier edit.

**Parameters:**

`state` - a Hashtable containing modified parameters.  
`prefix` - a String containing the prefix for hash entries.

---

**getName**

```
public String getName()
```

Returns the name of this Real.



## com.cafean.Number Class Time

```

java.lang.Object
  |
+-com.cafean.Number.BaseNumber
  |
+-com.cafean.Number.Real
  |
+-com.cafean.Number.Time

```

```

public class Time
extends Real

```

A representation of time in seconds. English Units: s, SI Units: s.

Fields inherited from class [com.cafean.Number.Real](#)

[Unknown](#)

Fields inherited from class [com.cafean.Number.BaseNumber](#)

[BRITISH](#), [SI](#), [UNIT\\_NAMES](#)

### Constructor Summary

|        |                |
|--------|----------------|
| public | Time()         |
| public | Time(double d) |

### Method Summary

|                               |                            |
|-------------------------------|----------------------------|
| double                        | getConversionFactor()      |
| String                        | getENG_Units()             |
| java.text.DecimalFormat       | getFormat(double value)    |
| String                        | getSI_Units()              |
| static <a href="#">Time[]</a> | makeArray(double[] dArray) |
| static <a href="#">Time[]</a> | makeArray(Vector vector)   |

Methods inherited from class [com.cafean.Number.Real](#)

[abs](#), [add](#), [add](#), [arraysEqual](#), [clone](#), [compareTo](#), [convert](#), [convert](#), [convert](#), [convert](#), [convert](#), [divide](#), [divide](#), [divide](#), [divideby](#), [divideby](#), [equals](#), [equals](#), [equals](#), [equals](#), [equals](#), [getConversionFactor](#), [getCurrentDisplayValue](#), [getDisplayName](#), [getDoubleValue](#), [getENG\\_Units](#), [getFormat](#), [getHtmlUnits](#), [getHtmlUnits](#), [getLoadFormat](#), [getName](#), [getReferencedValue](#), [getSI\\_Units](#), [getSignificantFigs](#), [getStrippedValue](#), [getTransferData](#), [getTransferDataFlavors](#), [getUnitName](#), [getUnits](#), [getValue](#), [greaterthan](#), [greaterthan](#), [isDataFlavorSupported](#), [isKnown](#), [isUnknown](#), [lessthan](#), [lessthan](#), [lockToString](#), [multiply](#), [multiply](#), [normalizeArray](#), [restoreState](#), [setSignificantFigs](#), [setUnknown](#), [setValue](#), [setValue](#), [setValue](#), [setValue](#), [sortParallelRealArrays](#), [sqrt](#), [storeState](#), [subtract](#), [subtract](#), [toLoadString](#), [toLoadString](#), [toString](#), [toString](#), [unlockToString](#)

Methods inherited from class [com.cafean.Number.BaseNumber](#)

[clone](#), [compareDouble](#), [convert](#), [getENG\\_Units](#), [getFormat](#), [getSI\\_Units](#), [getUnits](#), [getUnitType](#), [isKnown](#), [isUnknown](#), [setKnown](#), [setUnitType](#), [setUnknown](#), [setValue](#), [setValue](#), [toString](#)

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### Time

```
public Time()
```

### Time

```
public Time(double d)
```

## Methods

### getConversionFactor

```
public double getConversionFactor()
```

### getSI\_Units

```
public String getSI_Units()
```

### getENG\_Units

```
public String getENG_Units()
```

(continued from last page)

## **makeArray**

```
public static Time\[\] makeArray(double[] dArray)
```

---

## **makeArray**

```
public static Time\[\] makeArray(Vector vector)
```

---

## **getFormat**

```
public java.text.DecimalFormat getFormat(double value)
```

## com.cafean.Number Class ValueOutOfRangeException

```

java.lang.Object
  |
  +- java.lang.Throwable
        |
        +- java.lang.Exception
              |
              +- com.cafean.Number.ValueOutOfRangeException

```

```

public class ValueOutOfRangeException
extends Exception

```

An Exception representing a checked value that is outside of its acceptable range.

### Constructor Summary

|        |   |
|--------|---|
| public | ValueOutOfRangeException()<br>Creates a new ValueOutOfRangeException with no message  |
| public | ValueOutOfRangeException(String message)<br>Creates a new ValueOutOfRangeException with the given message                                     |
| public | ValueOutOfRangeException(String name,int minVal,int maxVal,int val)<br>Creates a new ValueOutOfRangeException with the given values.          |
| public | ValueOutOfRangeException(String name,double minVal,double maxVal,double val)<br>Creates a new ValueOutOfRangeException with the given values. |
| public | ValueOutOfRangeException(String s,Real minVal,Real maxVal,Real val)<br>Creates a new ValueOutOfRangeException with the given values.          |

#### Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### ValueOutOfRangeException

```

public ValueOutOfRangeException()
    Creates a new ValueOutOfRangeException with no message

```

---

## ValueOutOfRangeException

```
public ValueOutOfRangeException(String message)
```

Creates a new ValueOutOfRangeException with the given message

### Parameters:

message - a String containing the message for this exception.

---

## ValueOutOfRangeException

```
public ValueOutOfRangeException(String name,
                                int minVal,
                                int maxVal,
                                int val)
```

Creates a new ValueOutOfRangeException with the given values.

### Parameters:

name - a String containing the name of the value that is out of range.

minVal - the minimum value of the checked range.

maxVal - the maximum value of the checked range.

val - the value checked.

---

## ValueOutOfRangeException

```
public ValueOutOfRangeException(String name,
                                double minVal,
                                double maxVal,
                                double val)
```

Creates a new ValueOutOfRangeException with the given values.

### Parameters:

name - a String containing the name of the value that is out of range.

minVal - the minimum value of the checked range.

maxVal - the maximum value of the checked range.

val - the value checked.

---

## ValueOutOfRangeException

```
public ValueOutOfRangeException(String s,
                                Real minVal,
                                Real maxVal,
                                Real val)
```

Creates a new ValueOutOfRangeException with the given values.

### Parameters:

name - a String containing the name of the value that is out of range.

minVal - a Real containing the minimum value of the checked range.

maxVal - a Real containing the maximum value of the checked range.

val - the Real value checked.

---

# Index

## A

abs 574  
AbstractBeanComponent 5, 6  
AbstractComponent 12, 13  
AbstractInsertHandler 505  
AbstractModel 32  
AbstractPathHandler 508  
actionPerformed 255, 498  
activate 479, 486, 488, 502  
activateRedoButton 305  
activateUndoButton 305  
add 89, 170, 544, 548, 555, 558, 562, 573  
addAbstractModel 308  
addAnnotation 131, 267  
addBoxSelectionListener 179  
addChild 57  
addComment 48, 99  
addComponent 34, 35, 128  
addComponentListener 20  
addComponentLookup 44  
addComponents 128, 266  
addConnection 18, 80  
addConnectionSelectionListener 201  
addContextHelp 316  
addCurrentExportItems 524  
addCurrentToolsItems 524  
addDrawnComponent 266  
addDrawnConnection 169  
addDrawnViewTab 302  
addElement 41  
addExportItem 312  
addImportItem 312  
addMenuItem 271, 313  
addMessage 13, 306, 307, 323, 324, 325  
addMouseHandler 372  
addMouseListenerToHeaderInTable 356  
addMultipleComments 48, 99  
addNotify 168, 236, 371  
addOverlapPanel 265  
addPibBlock 127  
addPlaybackPanel 316  
addPoint 246, 404

addPopupMenuEntries 37  
addPopupMenuItems 382, 393, 406  
addPropertyChangeListener 462  
addRegisteredDialog 315  
addSelectedComponent 175  
addSelectedComponents 176  
addSound 305, 325  
addToFront 171  
addToModel 15  
addToolbar 265  
addToolChangeListener 363  
AlertMsg 321  
AlertSound 322  
ALL 458  
Angle 532  
Annotation 380  
AnnotationAction 471  
ArrayChangedEvent 141  
arraysEqual 573  
ARROW\_FILLED 401  
ARROW\_HOLLOW 401  
ARROW\_LINE 402  
ARROW\_NAMES 402  
ARROW\_NONE 401  
ARROW\_TYPES 402  
AsciiViewer 157

## B

BaseNumber 536  
BeanAction 473  
BeanBox 168  
BooleanEditor 424  
BorderStyleEditor 428  
BOTTOM 216  
boundsChanged 170  
boxSelectionChanged 180  
BRITISH 536  
buildView 127

## C

calcSelectionBounds 176  
canAddPoint 246, 404  
canAlign 174

canBeResized 223, 496  
cancelInsert 484, 505  
canConnectTo 17, 127  
canCopy 173  
canDelete 173  
canPaste 173  
canRemovePoint 246, 404  
capacity 544, 548, 558, 562  
CAT\_CONNECTION 31  
CAT\_CONSTANT 32  
CAT\_DATA\_SOURCE 31  
CAT\_FUNCTION 32  
CAT\_NUMERICS 32  
CAT\_RANGE 32  
CAT\_SOURCE\_ROOT 31  
CAT\_VALUES 32  
CAT\_VARIABLE 32  
CAT\_VIEW 32  
Category 56  
CategoryAction 475  
ccNumberCheck 40  
CENTER 216  
CENTER\_H 216  
CENTER\_V 216  
checkModel 38, 354  
checkRealArrayList 105  
checkRealArrayTable 105  
CIRCLE 214  
cleanUpDeleted 46  
Clear 220  
clear 80, 89, 543, 547, 557, 561  
clearCheckErrors 308, 323  
clearCheckWarnings 308  
clearComponentLookup 43  
clearConnections 18  
clearDbIds 44, 88, 101  
clearLinks 220  
clearRubberBoxes 284  
clearSelection 176, 201, 270  
clearViewSelection 131  
clearWarningCount 323  
ClientPluginLoader 181  
clone 14, 58, 72, 78, 80, 91, 99, 118, 126, 193, 219, 537, 543, 547, 557, 561, 571  
closeAbstractModel 303  
closeAllViews 103  
closeMessageWindow 303  
COLOR\_OPTIONAL 458  
compare 353  
compareDouble 538  
compareRowsByColumn 353  
compareTo 58, 102, 554, 572  
compareVersion 45  
complete 17, 132  
completeConnection 480  
componentChanged 64, 158, 235, 249  
ComponentChangedEvent 143  
componentConnected 64, 158, 236  
componentDeleted 64, 158, 235  
componentDisconnected 65, 158, 236  
componentHidden 394  
ComponentList 62  
componentMoved 394  
componentRemoved 248  
componentReshaped 248  
componentResized 394  
ComponentSelectionEditor 432  
ComponentSelector 185, 186  
componentShown 394  
ConnectingPt 191, 192  
Connection 69, 70  
ConnectionBean 75  
ConnectionData 78  
ConnectionList 80  
connectionPointRemoved 170, 248  
ConnectionSetPanel 200  
connectLinks 221  
ConnectMouseHandler 479  
ConnectSound 322  
connectTo 19  
contains 81, 89, 226, 247, 402, 403, 543, 547, 557, 561  
containsRestartChanges 46  
convert 532, 537, 553, 570  
copy 7, 14, 126, 173  
copyFrom 6, 99, 127  
createBeanSelectionMenu 363  
createBorderRegion 225  
createCategoryToolBars 363  
createCenterShape 225  
createComponent 35, 472, 474

createConnectionPrototypes 219  
createConnectionPt 219  
createCursor 364  
createDataPages 103  
createDisplayBeans 217, 245, 365  
createDrawnComponent 20, 73, 131  
createDrawnConnection 250  
createModel 304  
createNewModel 516  
createPibBlock 395  
createPopupMenu 159, 176, 232  
createSelected 364  
createSourceData 19  
createTargetData 19  
createTemplateEntry 237, 250  
createZoomMenu 177  
CROSSFLOW\_DROP\_ZONE 191  
CROSSHATCH 215  
CURSOR\_CONNECT 478  
CURSOR\_CONNECT\_COMPLETE 479  
CURSOR\_CONNECT\_OFF 478  
CURSOR\_MANIP\_P 362  
CURSOR\_PAN 492  
CURSOR\_PAN\_GRIP 492  
CURSOR\_ZOOM\_IN 501  
CURSOR\_ZOOM\_OUT 501  
cut 173

## D

DATA\_COMPLETE 98  
DATA\_ERROR 98  
DATA\_INCOMPLETE 97  
DATA\_WARNING 98  
dateStamp 309, 325  
DBtypeCode 16  
deactivate 479, 488, 495, 502  
deactivateRedoButtons 310  
debug 105, 301  
deepClone 59  
DEFAULT\_SIZE 244  
delete 173  
deleteAllComments 49, 100  
deleteComment 49, 100  
deleteRubberBox 284

deleteRubberBoxes 284  
deleteRubberLine 284  
deleteRubberLines 283  
DIAMOND 215  
DIFF\_VAL 438  
Dimless 540  
disable 161  
DISABLED 458  
disableMainFrame 310  
disconnect 17, 72  
disconnectAllMyLinks 231  
disconnectFrom 17  
DisconnectSound 322  
divide 555, 573, 574  
divideby 573  
doMultiEdit 338  
DoubleArray 543  
DOWN 216  
draw 222, 247, 256, 277  
drawLabelStrings 222  
DrawnComponent 216  
DrawnConnection 244, 245  
DrawnUserValue 255  
DrawnView 264  
DrawnViewComponent 275  
drawRubberBox 284  
drawRubberBoxes 284  
drawRubberLine 283  
drawRubberLines 283  
dumpBlockParams 6, 75, 83

## E

editorChangeValue 428  
ElementBean 83  
ElementList 87  
ElementsAddedEvent 145  
ElementsRemovedEvent 147  
EllipticalAnnotation 387  
enableMainFrame 310  
ensureVersion21 420  
equals 58, 78, 103, 554, 572  
executeModelValidations 53  
executeUserDefinedFunctions 51  
exportModelMetrics 53



exportModelMetricsSpec 53

## F

findByCC 62  
findByDB\_ID 88  
findByIdent 87, 88  
findClientPlugin 313  
findComponentByCC 33  
findComponentByDB\_ID 34  
findComponentByIdent 34  
findComponentGroup 38  
findComponentsInside 174  
findDisplayBeanClass 365  
findElementByDB\_ID 42  
findElementByIdent 42  
findEmbedded 276  
findEquivalentSharedComponent 51  
findModelByIdent 308  
findPlugin 313  
findReal 52  
finishInsert 484, 505  
fireComponentChanged 20, 133  
fireComponentConnected 20  
fireComponentDeleted 20, 133  
fireComponentDisconnected 20  
firePropertyChange 445, 469  
fixme 105  
flip 227  
FloatArray 547  
formatChanged 346  
forwardMouseEvent 286

## G

GenericObject 98  
get 88, 544, 548, 558, 562  
getAbstractModelAt 308  
getAbstractModelCount 309  
getAnnotationArray 132  
getAnnotations 169, 271  
getArcHeight 414  
getArcWidth 413  
getArray 335  
getArrows 406

getArrowSize 407  
getArrowTypeName 402  
getAsciiPrintWriter 517  
getAsciiStyledDocument 517  
getAsText 425, 429, 433, 439, 462, 468  
getAttributeGroup 138  
getAttributeGroups 138  
getAttributeIndex 137, 459  
getAttributesForGroup 138  
getBackground 133, 419  
getBadgeColors 195  
getBadges 195  
getBeanBox 234, 271, 371  
getBorderType 429  
getButtonIcon 479, 486, 489, 492, 495, 501  
getCanvasSize 129, 269  
getCatCCCComparator 19  
getCategories 33  
getCategory 13, 72, 90, 127, 434, 475  
getCategoryObject 36  
getCCnumber 102, 138  
getCCNumberComparator 18  
getCCNumberIncrement 39  
getCellEditorValue 337  
getChildren 57  
getClassPrereqs 527  
getClientPluginAt 313  
getClientPluginCount 313  
getClipboard 302  
getClockwiseFace 229  
getCodePlugins 314  
getCollapsedIcon 58  
getColorArray 133  
getColumnClass 355  
getColumnCount 355  
getColumnName 355  
getComment 49, 100  
getComments 100  
getComponent 14, 60, 143, 229  
getComponentCategory 364  
getComponentCCNumber 103  
getComponentCount 36  
getComponentDependencies 14  
getComponentGroups 38  
getComponentID 230

getComponentIterator 35  
getComponentNumber 103  
getComponents 33, 177, 268  
getConnectingLocation 238  
getConnectingPt 231, 232, 276  
getConnectingPtAt 226  
getConnectionColor 73  
getConnectionCount 18, 81  
getConnectionData 192  
getConnectionIndex 78  
getConnectionName 21  
getConnections 18, 80  
getConnectionSource 287, 480  
getConnectionStroke 73  
getConnectionTypes 13  
getConnectSize 231  
getConversionFactor 532, 540, 552, 569, 578  
getCounterFace 229  
getCreateDate 47  
getCrossflowIndex 227  
getCurrentCursor 363, 364, 479, 483, 489, 493, 496, 501, 505  
getCurrentDisplayValue 571  
getCurrentModel 303  
getCurrentTool 364  
getCursor 479, 489, 496  
getCustomEditor 425, 430, 433, 440, 463, 469  
getCustomPopupActions 16, 233  
getCustomPopupItems 16, 73, 76, 130, 237, 276  
getDataState 102  
getDB\_ID 102  
getDeckWriter 37  
getDefaultDrawLength 221  
getDefaultDrawWidth 220  
getDescription 48, 100, 498  
getDimensionless 52  
getDisplayName 569  
getDisplayValue 533  
getDocDescription 72  
getDocumentLinks 50  
getDoubleValue 346, 571  
getDrawAngle 221  
getDrawingFace 227  
getDrawingObject 221  
getDrawnComponent 169  
getDrawnComponentAt 267  
getDrawnComponentCount 267  
getDrawnComponents 177, 267  
getDrawnView 129, 374, 380  
getElementAt 42  
getElementCount 41  
getElementIterator 41  
getEmbedConnArray 132  
getEmbeddedCons 134  
getENG\_Units 532, 536, 540, 553, 569, 578  
getExpandedIcon 58  
getExportNote 51  
getExportUnits 52  
getFaceByAngle 228  
getFeaturePlugins 314  
getFileChooser 315  
getFillColor 222  
getFont 419  
getForeground 419  
getFormat 345, 532, 537, 554, 571, 579  
getGlassPane 234, 371  
getGridLineColor 135  
getGroupedConnections 21  
getHandle 195  
getHandlerID 489  
getHandleSize 196, 231  
getHead1 407  
getHead2 407  
getHorizGridSpacing 136  
getHorizSnapSpacing 136  
getHtmlUnits 569  
getIcon 498  
getIdent 101  
getImage 131  
getImageData 131, 395  
getImageElement 134  
getImageURL 58  
getIndexes 146, 148  
getIndexForRow 356  
getIndicatorColor 217  
getInitialCCNumber 38  
getIterator 81  
getJavaInitializationString 425, 429, 433, 440, 463, 468  
getLabel 257, 420, 517  
getLabelPlural 517  
getLastCCNumber 40

getLeftComponent 71  
getLeftComponentID 71  
getLeftConnectData 70  
getLength 221  
getLenScaleFactor 50, 218  
getLineThickness 388, 405, 414  
getLoadFormat 571  
getLocation 129  
getLongValue 346  
getLoopCheck 38  
getMajorCreationVersion 104  
getMajorVersion 104  
getMaxCCNumber 39  
getMaxHeight 229  
getMaximumSize 171, 283, 373  
getMaxWidth 229  
getMenubar 265  
getMinimumSize 171, 277, 373, 394  
getMinorCreationVersion 105  
getMinorVersion 104  
getMinWidth 229  
getMirrorImageShape 225  
getMode 315  
getModel 14, 53, 114, 115, 169, 265, 303, 338, 348, 353, 434, 464, 469  
getModelDB\_ID 43  
getModelMajorVersion 45  
getModelMinorVersion 45  
getModelOptions 49  
getMouseHandler 372  
getName 16, 48, 99, 498, 523, 576  
getNeedsTarget 197  
getNewArray 142  
getNewCompIdent 14, 105  
getNewComponentIdent 36  
getNewElementIdent 42  
getNewInsertHandler 389, 407, 414, 510  
getNextCCNumber 39  
getNormalObj 219  
getNumberConnections 231  
getNumberOfErrorsFound 305, 325  
getNumberOfWarningsFound 305  
getNumberValue 346  
getNumComments 49, 100  
getNumSelected 176  
getOldArray 142  
getOppositeFace 228  
getOrb 312  
getOrder 21  
getOrderComparator 18  
getOrientation 228  
getOrientationMenu 232  
getOrientationName 217  
getOtherSide 70  
getOtherSideData 70  
getOutputStream 323  
getOwner 14, 47, 60  
getPad 194  
getPanel 257  
getPanelSize 171, 373  
getParent 57, 236  
getParentDB\_ID 44  
getPath 249, 403  
getPdfViewer 316  
getPixelsPerMeter 134  
getPlugin 33  
getPluginAt 313  
getPluginClass 527  
getPluginCount 314  
getPluginIcon 517  
getPluginId 36, 523, 527  
getPluginInfo 525  
getPluginPreferences 525  
getPluginPrereqs 525, 527  
getPluginType 516, 519, 523  
getPreferredSize 171, 233, 256, 373  
getPrefs 311  
getProgramOptions 49  
getProjectDB\_ID 43  
getRealByIndex 52  
getRealSize 17  
getReferencedValue 574  
getReferenceModel 309  
getRegisteredDialogs 315  
getRightComponent 71  
getRightComponentID 71  
getRightConnectData 70  
getRootComponents 49  
getRootParent 57  
getRotatedLocation 196

getRowCount 354  
getSamPackage 516  
getSaveFile 44  
getSaveFileName 44  
getScale 172, 374  
getScrollPane 374  
getSelected 176  
getSelectedComponents 268  
getSelectedConnector 226  
getSelectedDropZone 227  
getSelectedIndex 329  
getSelection 174, 176, 186  
getSelectionBounds 176  
getSharedComponents 13  
getShowPoints 130  
getSI\_Units 532, 536, 540, 553, 569, 578  
getSignificantFigs 575  
getSize 129  
getSnapHomeDirectory 315  
getSource 248  
getSplashStatus 303  
getSplashTitle 303  
getStrippedValue 574  
getSubComponentAt 237, 276  
getTableCellEditor 330, 434  
getTableCellEditorComponent 338  
getTags 425, 430, 433, 440, 463, 468  
getTarget 248  
getText 419  
getThisSideData 71  
getTitle 271  
getToolbox 265  
getToolTipText 58, 221, 247, 283, 405  
getTooltipText 490  
getTransferData 574  
getTransferDataFlavors 574  
getTransparency 135  
getType 193  
getUndoManager 304  
getUndoSupport 304  
getUnitIndex 52  
getUnitName 571  
getUnits 52, 257, 538, 569  
getUnitsDisplay 52  
getUnitType 536  
getUsedBounds 248, 278, 405  
getUserName 312  
getValidationOptions 53  
getValidationTests 53  
getValidLabel 314  
getValue 347, 425, 429, 433, 440, 463, 468, 553, 570  
getValueAt 354  
getValueLabel 258  
getValueText 258  
getVersion 303, 525, 527  
getVertGridSpacing 136  
getVertSnapSpacing 136  
getViewComponent 177, 265  
getViewportCenter 371  
getViewPosition 130, 269  
getViewSize 269  
getWarningCount 325  
getWidthScaleFactor 50, 138, 177, 218  
getWriteable 158  
getX\_Pos 230  
getXDistBetweenCPs 223  
getXDistBetweenXflowCPs 223  
getY\_Pos 230  
getZoomablePanel 235, 271  
getZoomScale 130, 269  
getZoomSize 373  
GlassPanel 282  
greaterthan 554, 555, 572, 573

## H

halfPI 531  
HALFWIDTH 244  
hasBlocks 128  
hashCode 554  
hasModelMetrics 53  
hasSubComponents 237, 276  
hideComponentNumber 186  
hideComponentType 186

## I

ICON\_ELL\_ANNOT 362  
ICON\_IMAGE\_ANNOT 362  
ICON\_LINE\_ANNOT 362

ICON\_NEW 362  
ICON\_REC\_ANNOT 362  
ICON\_TXT\_ANNOT 362  
ImageAnnotation 393  
INACTIVE 439  
includeInLoopcheck 21  
incrementMajorVersion 46  
incrementMinorVersion 46  
indexOf 89, 544, 548, 558, 562  
InfoMsg 321  
InfoSound 322  
init 200, 311, 436, 463, 468  
InitDrawing 217, 245, 255, 276  
INLET\_DROP\_ZONE 191  
INLET\_MOVEABLE\_CONNECTOR 191  
insertFailed 348  
InsertMouseHandler 483  
instance 300  
Int 552  
IntegerArray 557  
InteractiveMouseHandler 485  
InternalErrMsg 322  
InternalErrSound 322  
inverseTransformPoint 374  
INVISIBLE\_CONNECTOR 191  
isAnimatable 517  
isAssociated 520  
isAutoScale 233  
isBeanBased 517  
isCanceled 187  
isCancelled 329, 334, 432  
isCellEditable 355  
isCompActionSelected 364  
isConnected 194  
isContained 63  
isDashed 407  
isDataFlavorSupported 574  
isDeleted 101  
isDirty 51  
isDrawBadges 233  
isEditable 256  
isEditingRestart 46  
isEqualTo 73  
isEquivalent 116  
isFilled 388, 413  
isGeneratePoints 249  
isGridAbove 137  
isIndependentComponent 70  
isInProgress 479  
isKnown 537, 575  
isLocked 137  
isMostlyVerticalDown 533  
isMostlyVerticalUp 533  
isObjectInsideBounds 225, 247, 405  
isOkayForExport 17, 18  
isOpaque 413  
isPaintable 425, 429, 433, 440, 463, 468  
isPlenumShaped 218  
isPosnSet 232  
isPropertyActive 138, 459  
isPropertyEnabled 137, 458  
isPropertyRequired 137, 459  
isPropertyResizable 138, 459  
isPropertyRestartEditable 137, 459  
isRepresenting 276  
isRestartableModel 46  
isRestartResizable 138, 460  
isRounded 413  
isScalable 223  
isSegmentSet 217  
isSelected 176, 236  
isShowGrid 134  
isShowLabel 256  
isShowNumbers 440  
isShowUnits 256  
isSnapToGrid 135  
isSorted 89  
isSubset 57  
isSuperset 57  
isUncertain 348  
isUnknown 346, 537, 575  
isValidCCNumber 40  
isValveShaped 218  
isVerticalDown 533  
isVerticalUp 533  
isViewVisible 131  
isVisible 194  
isVisual 58, 73  
itemStateChanged 428, 439, 445, 447, 450, 456  
iterator 63, 90

**J**

joinPipe 47

**L**

label 16, 73, 126

layoutComponents 40

layoutView 127, 267

LEFT 215

lessthan 554, 572

LineAnnotation 402

load 126, 519

loadComponent 134

loadComponents 174

loadDisplayBean 365

loadDisplayBeanClasses 366

loadDrawnComponent 50, 237, 250, 259, 276

loadMainMenuItems 524

loadPlugin 527

loadReferenceModel 309

loadRestartData 37

loadSettings 524

loadUserConstant 152

loadUserFunction 152

loadUserVariable 151

loadVedaExport 365

loadViewMenuItems 524

loadVisualComponents 151

LocalSubmitDialog 290

locateComponentAt 286

lockToString 569

LongArray 561

lookupComponent 44

lostOwnership 311, 348

**M**

main 312, 544, 548, 558, 562

MainFrame 301

makeArray 541, 578, 579

max\_positions 216

MECodePlugin 516

MEDReader 151

MEFeaturePlugin 519

menuCanceled 270

menuDeselected 270

menuSelected 270

MEPlugin 522

MEPluginData 526

MessageWindow 322

minimizeView 178

MODE\_CLEAR\_CONSTANTS 301

MODE\_EXPORT\_ASCII 301

MODE\_IMPORT 301

MODE\_NORMAL 301

MODE\_SAVE\_MED 301

modelAdded 519

modelRemoved 519

mouseClicked 159, 234, 246, 277, 285, 382, 403, 480, 483, 489, 493, 505, 508, 512

mouseDragged 234, 247, 285, 404, 480, 483, 486, 493, 495, 502, 508, 512

mouseEntered 159, 233, 247, 285, 382, 405, 489, 505

mouseExited 159, 234, 247, 285, 382, 405, 490, 506

MouseHandler 488

mouseMoved 234, 247, 277, 285, 405, 480, 483, 490, 505, 508

mousePressed 159, 234, 245, 285, 382, 403, 480, 483, 486, 492, 495, 502, 508, 512

mouseReleased 159, 234, 246, 285, 382, 404, 480, 483, 486, 493, 495, 502, 509, 512

move 196

moveRel 224

moveTo 224, 381, 402

multiply 555, 573

**N**

NamedIntEditor 439

NamedValueSelector 328, 329

NamelistBooleanEditor 444

NamelistDoubleEditor 447

NamelistIntEditor 450

NamelistNamedIntEditor 451

NamelistRealEditor 455

NONE 214, 458

normalize 347

normalizeArray 575

NoSound 322

---

## O

objectAdded 40  
offsetWindowLocation 311  
open 516  
openFileSelector 310  
openMessageArea 307  
OPTIONAL 458  
organizeSelection 178  
organizeView 177  
OUTLET\_DROP\_ZONE 191  
OUTLET\_MOVEABLE\_CONNECTOR 191

## P

paint 172, 222  
paintComponent 222, 286, 388, 394, 402, 412  
paintImmediately 172, 173  
paintValue 425, 429, 434, 440, 463, 469  
PanMouseHandler 492  
paste 174  
performExport 202  
performLoopCheck 38  
PIXELS\_P\_METER 216  
PIXELS\_PER\_METER 125  
pluginException 182  
pluginLoaded 181  
popupBeanDataDialog 6  
popupDataDialog 7, 16, 76, 103, 134  
popupEditor 258, 381, 388, 394, 403, 413, 419  
print 172, 222  
printView 268  
processCommand 304, 523

## R

rangeCheck 106, 107, 108  
readAnnotationArray 132  
readBlock 83  
readBlockParams 6, 76, 84  
readByteArray 259  
readDrawingBlock 151  
readEmbedConnArray 132  
readTemplateEntry 237, 250

Real 568  
RealArrayDialog 333, 334  
RealArrayEditor 462  
RealBeanEditor 467  
RealEditor 337  
RealTextField 344, 345  
rebuildConnections 16  
reconnectIdentReferences 43, 72, 80, 88, 101, 113  
reconnectImage 15  
RectangularAnnotation 412  
RectangularInsertHandler 511  
redrawSelection 178  
refresh 158, 172, 201, 350, 372, 382  
refreshCompBounds 372  
refreshSteamTable 310  
remove 89, 170, 544, 548, 558, 562  
removeAbstractModel 308  
removeBoxSelectionListener 179  
removeClosestPoint 246, 403  
removeComponent 35  
removeComponentListener 20  
removeConnection 81  
removeConnectionSelectionListener 201  
removeDeleted 88  
removeDrawnViewTab 302  
removeElement 41  
removeFromModel 15, 126  
removeMouseHandler 372  
removeNotify 236, 283  
removePlaybackPanel 316  
removePoint 246, 404  
removePropertyChangeListener 463  
removeRegisteredDialog 315  
removeSelectedComponent 175  
removeToolChangeListener 363  
removeVerify 15, 126  
renodalizePipeCells 48  
renumberComponents 50  
renumberSelectedComponents 177  
repaint 172, 248, 286, 405  
reportModelCheck 41  
repositionLinks 227  
requestSaveFileName 305  
REQUIRED 458  
requiresTarget 287, 480

reset 394  
resetAllUnits 304  
resetConnections 178  
resetCursor 310  
resetPosition 224  
resetSize 382, 388, 394, 405, 413, 420  
resetWindowingMode 302  
resetZoomMenu 269  
restoreState 21, 84, 98, 175, 235, 249, 258, 277, 381, 388, 395, 406, 419, 576  
revalidate 372, 406  
RIGHT 215  
rotate 196  
rotateTo 224  
run 161

**S**

save 519  
saveModel 37  
scaleIt 223  
SEGMENT\_BOTH 215  
SEGMENT\_INLET 215  
SEGMENT\_NONE 215  
SEGMENT\_OUTLET 215  
SEGMENT\_SPECIAL 215  
selectCategory 178  
selectComp 325  
SelectMouseHandler 495  
SELECTOR\_OPEN 301  
SELECTOR\_SAVE 301  
set 544, 548, 558, 562  
setArcHeight 414  
setArcWidth 413  
setArrows 406  
setArrowSize 407  
setAsText 425, 429, 434, 440, 463, 468  
setAutoScale 233  
setBackground 133, 259, 419  
setBackupComponent 217  
setBadgeColors 195  
setBadges 195  
setBorder 381  
setBounds 219, 277, 381  
setCanvasSize 129, 269  
setCategory 90, 434  
setCCNumberIncrement 39  
setChildLocation 311  
setColorArray 133  
setComments 101  
setComponent 230  
setComponentNumber 17, 103  
setComponentNumberConstrained 7  
setConnected 194  
setConnectionIndex 78  
setConnectionStart 480  
setConnectPtType 194  
setContext 439, 445, 447, 448, 450, 456  
setCreateDate 47  
setCreationVersion 104  
setCreator 433  
setCurrentAction 364  
setCurrentModel 303  
setCurrentTool 363  
setDashed 407  
setDataState 102  
setDB\_ID 102  
setDeleted 16, 101  
setDescription 48, 100  
setDimension 464  
setDirty 51  
setDisconnected 194  
setDrawAngle 221  
setDrawBadges 233  
setDrawHeight 231  
setDrawWidth 231  
setEditable 256, 433  
setEmbeddedCons 134  
setEqualTo 220, 388, 403, 413  
setExportNote 51  
setFileChooserLocation 311  
setFilled 388, 413  
setFont 419, 433  
setForeground 259, 419, 468  
setFormat 345  
setGeneratePoints 249  
setGridAbove 137  
setGridLineColor 135  
setHandle 195  
setHandlerID 489



setHandleSize 195  
setHead1 407  
setHead2 407  
setHeight 381  
setHorizGridSpacing 136  
setHorizSnapSpacing 137  
setIdent 42, 43, 101  
setImageData 131, 395  
setImageElement 134  
setInitialValues 291  
setKnown 537  
setLabel 257, 420  
setLabelString 232  
setLeftComponent 71  
setLeftComponentID 71  
setLenScaleFactor 50, 218  
setLineThickness 389, 406, 414  
setLocation 129, 245  
setLocked 137, 265, 363  
setLockedConstrained 137  
setMajorCreationVersion 104  
setMajorVersion 104  
setMinorCreationVersion 104  
setMinorVersion 104  
setMode 316  
setModel 15, 114, 338, 348, 353, 434, 464, 469  
setModelDB\_ID 42  
setModelMajorVersion 45  
setModelMinorVersion 46  
setName 48, 99, 523  
setNeedsTarget 197  
setOpaque 387  
setOrder 21  
setOrientation 228  
setOrientationByAngle 228  
setOutputStream 323  
setOwner 47  
setPad 194  
setPaintEnabled 172  
setPanel 257  
setPanelSize 372  
setParent 236  
setParentDB\_ID 44  
setPath 245, 403  
setPixelsPerMeter 134  
setPlaybackTime 316  
setProgramOptions 50  
setProjectDB\_ID 43  
setRightComponent 72  
setRightComponentID 71  
setRounded 413  
setRunOptions 291  
setSaveFile 45  
setScale 171, 373  
setSelected 186, 236, 249  
setSelectedComponent 175  
setShowGrid 135  
setShowLabel 256  
setShowNumbers 441  
setShowPoints 130  
setShowUnits 257  
setSignificantFigs 575  
setSize 129, 420  
setSizeTo 223  
setSnapToGrid 135  
setSplashStatus 302  
setStoredBackground 133  
setTargetComponent 158  
setText 346, 419  
setTitle 271  
setTransparency 135  
setUnits 52, 257  
setUnitsConstrained 52  
setUnitType 536  
setUnknown 536, 572  
setupContextHelp 323  
setUpEditor 338  
setUpRealEditor 338  
setUserName 311  
setValue 347, 425, 428, 434, 440, 444, 447, 450, 455, 463, 468, 537, 553, 570  
setValueAt 354  
setValueLabel 258, 329  
setValueText 258  
setValueUncertain 348  
setValueUnknown 347  
setVertGridSpacing 136  
setVertSnapSpacing 136  
setViewCenter 374  
setViewPosition 131, 270

setVisible 127, 158, 268, 290, 310  
setWaitCursor 309  
setWidth 381  
setWidthScaleFactor 50, 139, 218  
setWindowLocation 311  
setX\_Pos 230  
setY\_Pos 230  
setZoomScale 130, 269  
showAllConnections 178  
showCancelButton 187  
showComment 49, 100  
showConnections 234  
showCreateDialog 304  
showOpenDialog 304  
SI 535  
size 62, 89, 544, 548, 558, 562  
snapCorba 301  
snapshotView 268  
sort 90, 354, 544, 548, 558, 562  
sortByColumn 355  
sortParallelRealArrays 575  
SpecialConnectionData 117, 118  
splitPipe 47  
sqrt 555, 574  
SQUARE 215  
STATIC\_CONNECTOR 191  
store 128, 237, 250, 259, 276, 381  
storeComponent 128  
storeDisplayBean 365  
storeDrawnComponent 51  
storeSettings 524  
storeState 84, 98, 99, 175, 235, 249, 258, 277, 381, 388, 394, 406, 418, 575  
storeUserConstant 151  
storeUserFunction 152  
storeUserVariable 151  
submitModel 516  
subtract 555, 573  
supportsCustomEditor 426, 430, 434, 440, 464, 469

## T

tableChanged 354  
TableSorter 353  
TextAnnotation 418

THEORYMAN 32  
Time 578  
toArray 62, 90, 545, 549, 559, 563  
toFront 271, 325  
toFrontDrawnViewTab 302  
TOLERANCE 508  
toLoadString 571  
TOOL\_CONNECT 362  
TOOL\_INSERT 362  
TOOL\_INTERACTIVE 362  
TOOL\_NONE 361  
TOOL\_PAN 361  
TOOL\_SELECT 361  
TOOL\_ZOOM 362  
Toolbox 363  
ToolboxAction 497  
toolChanged 371, 499  
TOP 216  
toString 18, 52, 58, 63, 72, 78, 91, 118, 127, 192, 219, 249, 348, 383, 479, 486, 490, 492, 495, 501, 537, 553, 571  
trace 105  
transformPoint 374  
translate 248, 278, 405  
translateConnectionToScreen 235  
translatePointToScreen 235  
TRIANGLE 214  
trimToSize 545, 549, 559, 563  
TYPE\_CODE\_PLUGIN 522  
TYPE\_FEATURE\_PLUGIN 522  
typeIsConnector 192  
typeIsCrossflow 193  
typeIsDropZone 192  
typeIsInlet 193  
typeIsOutlet 193  
typeIsStatic 193

## U

undockView 130  
UNIT\_NAMES 535  
unitsChanged 159, 350  
UNKNOWN 439  
Unknown 552, 568  
unlockToString 569  
UP 216

update 161  
updateComponentList 178  
updatePlaybackButtons 316  
Updater 161  
updateSelection 179  
updateToolbars 270  
updateVersion 18, 104  
userDisconnect 72  
UserErrorMsg 322  
UserErrorSound 322  
USERSMAN 32

## V

validate 102, 245, 406  
validateAllComponents 36  
value 552  
ValueOutOfRangeException 580, 581  
verifyBeansAvailable 365  
ViewComponent 126

## W

WarningMsg 321  
WarningSound 322  
write 153  
writeArrayLoadValue 110, 111  
writeBlockParams 6, 76, 84  
writeMuxLoadArray 109, 110  
writeName 21  
writeSP 109

## Z

ZoomablePanel 370  
ZoomMouseHandler 501  
zoomToFit 375