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

Appendix A
Source Code Documentation

August 2005

Copyright © 2003-2005, Applied Programming Technology, Inc.

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

Method Summary

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.
void	`copyFrom(GenericObject obj)`
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.
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.
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.

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.
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.
void	`setComponentNumberConstrained(int val_)` Sets this bean-based component's component number to the given number.
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.

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, 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

Constructors

(continued on next page)

(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:
UserDefinedValue, UserDefinedFunction, 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()` Create a AbstractComponent object that does not belong to any model and has a display number of 0
public	`AbstractComponent(AbstractModel model,int componentNumber)` Creates an AbstractComponent object, adds it to `model`and sets the given component number on it.

Method Summary

void	`addComponentListener(ComponentListener listener)` Adds a new component listener to this abstract component.
void	`addConnection(Connection con)` 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)` Adds the given message as type `severityCode`
void	`addMessage(String text,int severityCode)` Adds the given message as type `severityCode`

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

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

AbstractModel	`getModel()` Return the AbstractModel to which this AbstractComponent belongs.
String	`getName()` This is overwriting creates a default name for a component in case it is never given one on import.
int	`getNewCompIdent(int dbid,boolean preserveUnresolved,boolean useDbId)` Retrieves the ident of the component with the given DB_ID in this component's model.
int	`getOrder()` Gets this component's relative *order*value.
static Comparator	`getOrderComparator()` Returns a comparator object for comparing the component order of two AbstractComponent objects; for use in sorting or searching.
ComponentElement	`getOwner()`
boolean	`getRealSize(Real length,Real avgDiam,Real avgAngle)` This method is mearly a placeholder for the HydroComponent derivative.
SharedComponent[]	`getSharedComponents()` Returns an array of the SharedComponents that this component depends on and should carry with it during copy/paste operations.
boolean	`includeInLoopcheck()` Determines if this component should be included in Loopcheck calculations.
boolean	`isOkayForExport()`
boolean	`isOkayForExport(boolean prompt)`
String	`label()` Returns a String suitable for describing the component type on a dialog.
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.
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`.
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.
void	`removeComponentListener(ComponentListener listener)` Removes a component listener from this abstract component.

void	`removeFromModel(AbstractModel model)` This method removes this AbstractComponent from `model`.
boolean	`removeVerify()` Verifies that this component can be removed from it's model without unforeseen side effects and returns the truth of this assumption.
void	`restoreState(Hashtable state)`
void	`setComponentNumber(int num)` Setter for Componet number (user defines) also know as display number.
void	`setDeleted(boolean del)` If call with a parmater of true, the AbstractComponent object flagged as deleted.
void	`setModel(AbstractModel model)` Set the AbstractModel.
void	`setOrder(int order)` Sets this component's relative *order*value.
String	`toString()`
void	`updateVersion()` Updates this object's current version to be greater than or equal to it's model's version.
String	`writeName()` Returns the string to be used when writing a component out to ASCII file.

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`, `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`, `writeMuxLoadArray`, `writeMuxLoadArray`, `writeSP`, `writeSP`

Methods inherited from class `java.lang.Object`

`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 assisgned 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 parmater 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, JMenuItems, and JSeparators.

setComponentNumber

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

Setter for Componet 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 overriden 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 mearly 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 overriden. The version with the conID parameter should be overriden 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 disconected.

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 *order*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 *order*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 *(*`GenericObject`*s)*accessable by `ident` and `DB_ID`

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

Field Summary

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

static final Category	CAT_VALUES The Category inside this model that contains all user defined `values`.
static final Category	CAT_VARIABLE The Category inside this model that contains the user user defined `variables`.
static final Category	CAT_VIEW A Catagory for ViewComponent objects.
static final String	THEORYMAN Value: **`Theory Manual`**
static final String	USERSMAN Value: **`User's Manual`**

Fields inherited from class `com.cafean.client.analysis.GenericObject`
`DATA_COMPLETE`, `DATA_ERROR`, `DATA_INCOMPLETE`, `DATA_WARNING`

Constructor Summary

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

Method Summary

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

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.
abstract boolean	`checkModel()` Performs checks for accuracy on the current AbstractModel
void	`cleanUpDeleted()` Removes all deleted components from the lists maintained by this model.
void	`clearComponentLookup()` clears the pasted component lookup table.
void	`clearDbIds()` Clears the DB_ID's of this model's directly contained objects.
int	`compareVersion(GenericObject obj)` Compares the version of the given object with that of this model to dermine if the object's version is less than, equal to or greater than the version of this model.
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.
AbstractComponent	`createComponent(Category category)` Creates an appropriate AbstractComponent for the given leaf node category.
void	`deleteAllComments()` {@inheritDoc} Proxied to program options.
void	`deleteComment(int num)` {@inheritDoc} Proxied to program options.
boolean	`executeModelValidations(boolean printErrors)` This executes all of the `tests` that are found in `this` model.
void	`executeUserDefinedFunctions()` Executes all user defined functions in this model.
boolean	`exportModelMetrics(java.io.File dFile)` Export a file containing the state of all testable metrics for the given model.
boolean	`exportModelMetricsSpec(java.io.File dFile)` Export a file containing the metrics specification for the model.
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.
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.

AbstractComponent	`findComponentByDB_ID(int dbid)` Retrieves the component in this model that has the given dbid by searching through each category.
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.
AbstractComponent	`findComponentByIdent(int ident)` Retrieves the component in this model that has the given ident by searching through each category.
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.
ComponentNumberGroup	`findComponentGroup(Category category)` Retrieves the appropriate ComponentNumberGroup instance for the given Category.
GenericObject	`findElementByDB_ID(int dbid)` Retrieves the GenericObject in this model with the given DB_ID.
GenericObject	`findElementByIdent(int ident)` Retrieves the element in this model with the given ident.
SharedComponent	`findEquivalentSharedComponent(SharedComponent shared)` Retrieves an equivalent SharedComponent for the given component.
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.
Category[]	`getCategories()` Retrieves the root Categories for this model type.
Object	`getCategoryObject(Category category)` Retrieves the optional object associated with the given category.
int	`getCCNumberIncrement()` Retrieves the increment number for retrieved CC numbers.
String	`getComment(int num)` {@inheritDoc} Proxied to program options.
int	`getComponentCount()` Retrieves a count of the components in this model by calling `getComponentCount` on each Category returned by {@link #getCategories}.
int	`getComponentCount(Category category)` Retrieves a count of the components in this model that are in a subset of the given category.
ComponentNumberGroup[]	`getComponentGroups()` Retrieves an array of the ComponentNumberGroup instances used by this model to ensure that each component has a valid component number.

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

AbstractModel	getModel()
int	getModelDB_ID() Retrieves this model's ID on the database server.
int	getModelMajorVersion() Returns the major version number of this model.
int	getModelMinorVersion() Retrieves this model's minor version number
Object	getModelOptions() Returns the java bean object that contains this model's "model options" if this model uses such a thing.
String	getName() Returns this model's name.
int	getNewComponentIdent(int dbid,Category category,boolean preserveUnresolved,boolean useDbId) 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) Retrieves the ident of the element with the given DB_ID or ident in the given model.
int	getNextCCNumber(Category category) Retrieves a unique and available CC number for the given component Category.
int	getNextCCNumber(Category category,int requestedNumber) Retrieves a new and available CC number for the given component Category.
int	getNumComments() {@inheritDoc} Proxied to program options.
String	getOwner() Retrieves the username of the creator of this file.
int	getParentDB_ID() Retrieves the ID of this model's parent model in the database.
MECodePlugin	getPlugin() Returns a reference to the plugin that created this model.
abstract String	getPluginId() Retrieves the name of this model's code family.
ProgramOptions	getProgramOptions() **Deprecated.**
int	getProjectDB_ID() Retrieves the ID of this model's project in the database.

Real	`getRealByIndex(int index)` Returns the Real at the given index.
AbstractComponent[]	`getRootComponents()` Returns the set of AbstractComponents that should appear in the Navigator as peer nodes to ModelOptions.
java.io.File	`getSaveFile()` Retrieves the file this model was last saved to or opened from.
String	`getSaveFileName()` Retrieves the name of the file this model was last saved to or opened from.
int	`getUnitIndex(Real unit)` Finds the index of the given unit or -1 if the unit is not found inside this model.
int	`getUnitIndex(String siUnitsName)` Finds the index of the unit who's SI units are equal to the given String or -1.
int	`getUnits()` Returns the current units for this model.
String[]	`getUnitsDisplay()` Creates an array of strings for displaying model units for selection.
ValidationOptions	`getValidationOptions()` This gets the ValidationOptions that is stored inside a given model.
ValidationTest[]	`getValidationTests()` This gets the array of ValidationTests from the current model.
double	`getWidthScaleFactor()` Gets the scale factor used to adjust cell width for drawn components rendering components of this model.
boolean	`hasModelMetrics()` Return true if this model supports output of model metrics.
void	`incrementMajorVersion()` Increments this model's major version number and sets it's minor version to 0.
void	`incrementMinorVersion()`
boolean	`isDirty()` if true, this model has been modified since it was saved.
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.
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.

boolean	`isValidCCNumber(int ccNumber,Category category)` Returns true if the given cc number is theoretically valid.
boolean	`joinPipe(Vector selected)` This function joins two pipes into one.
abstract void	`layoutComponents(Vector drawnComponents,ProgressMon progress)` Lays out the given drawn components using a plugin-specific organization method.
DrawnComponent	`loadDrawnComponent(com.apt.xdr.PibBlock block)` Creates a custom DrawnComponent derivative from the given PibBlock.
abstract void	`loadRestartData(RestartData data)` Loads the given timeslice of RestartData into this model's components.
AbstractComponent	`lookupComponent(int dbid)` finds the component for the given dbid in the component lookup table
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.
boolean	`performLoopCheck(Vector components)` Performs a loop check on the given vector of components.
void	`reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)` Reconnects the ident references of this models directly contained objects.
boolean	`removeComponent(AbstractComponent component)` Removes the given component from the model view.
boolean	`removeElement(GenericObject element)` Removes the specified GenericObject from the model.
boolean	`renodalizePipeCells(AbstractComponent component)` This function renodalizes a pipe cells.
void	`renumberComponents(AbstractComponent[] components,int offset)` Ensures that the given components have appropriately unique CC numbers within their model.
void	`reportModelCheck(boolean status)` Displays a report of the model check to the message window and a OptionPane.
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.
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.
void	`setCCNumberIncrement(int increment)` Sets the increment number for retrieved CC numbers.

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

void	`setWidthScaleFactor(double fact)` Gets the scale factor used to adjust cell width for drawn components rendering components of this model.
String	`showComment(int num)` {@inheritDoc} Proxied to program options.
boolean	`splitPipe(Vector selected)` This function splits a pipe into two pipes.
com.apt.xdr.PibBlock	`storeDrawnComponent(java.awt.Component component)` Creates a custom PibBlock derivative from the given Component.
String	`toString()`
void	`validateAllComponents()` 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, 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 Cagetory 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`**`(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,
          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 dermine if the object's version is less than, equal to or greater than the version of this model. This is 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 objs 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 <= 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 overriden 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 overriden 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 overriden 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`**`()`

Returns this model's name. The default implementation currently uses this model's ProgramOptions object.

setName

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

Override GenericObject SetName to make sure view title is updated when this is called. The default implementation currently uses this model's ProgramOptions object.

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()
```

Deprecated.

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 DocumentLink[] getDocumentLinks(Object obj,
          String docType)
```

Return the array of document links for a given object.

loadDrawnComponent

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

Creates a custom DrawnComponent derivative from the given PibBlock.

Returns:

null of 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 of 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:

(continued from last page)

`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 Categ
ory("Pumps", null, ICON_PUMP,
ImageMgr.class.getResource("pump32.gif"));       private static final Category[] HYDROS = { CAT_PI
PE, 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 overriden 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	`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.

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.
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.

Method Summary

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

boolean	isVisual() Returns true if this category is of Visual Components.
String	toString()

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 Catagories

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 it's 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 ComponentCreator

public interface **ComponentCreator**

This interface allows for custom creation code to be passed around.

Method Summary

`void`	`createComponent(AbstractModel model)` Creates a new component, and adds it to the given model

Methods

createComponent

`public void` **`createComponent`**`(AbstractModel model)`

Creates a new component, and adds it to the given model

com.cafean.client.analysis
Interface ComponentElement

All Superinterfaces:
ModelElement

All Known Implementing Classes:
SnapPreferences, DummyCompElement, 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 contained by an `AbstractComponent`.

Method Summary

`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.
`ComponentElement`	`getOwner()` Retrieves the owner of this ComponentElement.

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

Method Summary

AbstractComponent	`findByCC(int cc,Category category)` Retrieves the AbstractComponent with the given component number.
boolean	`isContained(Category category)` Returns true if the given Category is contained in this list.
Iterator	`iterator(Category category)` Retutrns an Iterator object for iterating through this list.
int	`size(Category category)` Returns the number of components in this list that are part of the given category.
AbstractComponent[]	`toArray(Category category)` Returns an array containing all of the components in this list that are in the given Category.
String	`toString()` 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)`

Retutrns 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)` A component changed event tells the listener that the internal data of the component has changed.
void	`componentConnected(Connection con)` A component connected event tells the listener when a component completes a connection to a different component.
void	`componentDeleted()` A component deleted event tells the listener that the component has been deleted by the user.
void	`componentDisconnected(Connection con)` 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()` Creates a new instance of Connection with no model and no left or right component.
public	`Connection(AbstractModel model,AbstractComponent leftComp)` Creates a new instance of Connection with the given model and the given left component.
public	`Connection(AbstractModel model,AbstractComponent leftComp,int cc)` 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)` 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)` Creates a new instance of Connection with the given model and the given left component.

Method Summary

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

ConnectionData	getThisSideData(AbstractComponent component) Retrieves the ConnectionData for this side of this Connection.
boolean	isEqualTo(Connection con) This determines if the Connection passed in is the equivelant of this Connection.
boolean	isIndependentComponent() Returns true if this Connection can exists without it's left and right components. Non-Independent connections are deleted if and when their surrounding components are no longer available.
boolean	isVisual() Returns true if this connection should be visually represented.
String	label()
void	reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)
void	setLeftComponent(AbstractComponent component) Sets the left component of this connection.
void	setLeftComponentID(int ident) Sets the left component of this Connection by ident.
void	setRightComponent(AbstractComponent component) Sets the right component of this connection.
void	setRightComponentID(int ident) Sets the right component of this Connection by ident.
String	toString()
void	userDisconnect() Disconnects the connection from the Navigator.

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, 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`

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 exists without it's 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 contianing 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 equivelant 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	ConnectionBean()

Method Summary

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.
Vector	getCustomPopupItems() Creates Custom Menu Items for any popup dialog involving this component.
void	popupDataDialog(java.awt.Window parent,boolean modal)
boolean	readBlockParams(com.apt.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.
boolean	writeBlockParams(com.apt.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.

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, 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

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.apt.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.apt.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, JMenuItems, 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.DrawnViewComponent`s.

See Also:
com.cafean.client.ui.DrawnComponent.createConnectionPt(), com.cafean.client.ui.ConnectingPt.getConnectionData()

Constructor Summary

public	`ConnectionData()` Creates a new instance of ConnectionData
public	`ConnectionData(int index)` Creates a new instance of Connection data with a specific connection point index specified.

Method Summary

Object	`clone()`
boolean	`equals(Object obj)` Determines the equivalence of this ConnectionData with the given object.
int	`getConnectionIndex()` Retrieves this ConnectionData's ConnectingPt index.
void	`setConnectionIndex(int connectionIndex)` 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)` Creates a new instance of ConnectionList
public	`ConnectionList(AbstractComponent component,ConnectionList list)` Creates a new instance of connectionList, copying an existing list.

Method Summary

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

void	removeConnection(Connection con) 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 DummyCompElement

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

All Implemented Interfaces:
ComponentElement

---

public class **DummyCompElement**
extends Object
implements ComponentElement

An empty component element for use in stub methods or in specialized editors for an entire component. This element should be used when an editor does not actually change the value directly and does not fire an event immediately.

Constructor Summary

public	`DummyCompElement(ComponentElement owner)` Creates a new instance of DummyCompElement

Method Summary

boolean	`equals(Object obj)`
AbstractComponent	`getComponent()`
AbstractModel	`getModel()`
ComponentElement	`getOwner()`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DummyCompElement

`public` **`DummyCompElement`**`(ComponentElement owner)`

Creates a new instance of DummyCompElement

Methods

(continued from last page)

equals

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

---

getOwner

```
public ComponentElement getOwner()
```

---

getComponent

```
public AbstractComponent getComponent()
```

---

getModel

```
public AbstractModel getModel()
```

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`	`ElementBean()` Creates a new ElementBean with a DATA_INCOMPLETE data state and named unnamed
`public`	`ElementBean(int componentNumber)` Creates a new ElementBean with a DATA_INCOMPLETE data state and named unnamed

Method Summary

`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.
`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.
`boolean`	`readBlockParams(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.
`boolean`	`readBlockParams(com.apt.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.
`void`	`restoreState(String prefix,Hashtable state)` Restore the state of the bean from an earlier edit.

void	storeState(String prefix,Hashtable state) Store the state of the bean to permit undo.
boolean	writeBlockParams(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.

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, 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

Constructors

ElementBean

public **ElementBean**()

Creates a new ElementBean with a DATA_INCOMPLETE data state and named unnamed

ElementBean

public **ElementBean**(int componentNumber)

Creates a new ElementBean with a DATA_INCOMPLETE data state and named unnamed

Methods

dumpBlockParams

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.

readBlock

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.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.

readBlockParams

```
public boolean readBlockParams(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.

readBlockParams

```
public boolean readBlockParams(com.apt.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()` Creates a new instance of ElementList with a category of `Components`that starts in a sorted state.
public	`ElementList(boolean sorted)` Creates a new instance of ElementList with a category of `Components`.
public	`ElementList(String category,boolean sorted)` Creates a new instance of ElementList with the given category name and the given initial sorted state.

Method Summary

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

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

String	toString() 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 Componentsthat 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

findByIdent

public GenericObject **findByIdent**(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

findByIdent

```
public GenericObject findByIdent(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 resturned.

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()`

Retutrns 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 Indication that this object is complete and error free. Value: **2147483647**
static final int	DATA_ERROR Indication that this object has a fatal error that requires attention. Value: **2**
static final int	DATA_INCOMPLETE Indication that this object is missing required data. Value: **0**
static final int	DATA_WARNING Indication that this object has a non-fatal error that requires attention. Value: **1**

Constructor Summary

public	GenericObject() Creates a new GenericObject with a DATA_INCOMPLETE data state and named unnamed
public	GenericObject(int componentNumber) Creates a new GenericObject with a DATA_INCOMPLETE data state and named unnamed

Method Summary

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

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

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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
void	`reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)` Resets this component's internal ident references to refer to appropriate components in the current AbstractModel.
void	`restoreState(Hashtable state)` Restore the state of the bean from an earlier edit by using copyFrom on the previously stored clone.
void	`restoreState(String prefix,Hashtable state)` Restore the state of the bean from an earlier edit.
void	`setComments(int index,String value)` Sets the comment at the given index.
void	`setComments(String[] comments)` Sets all of the comments inside this object as Strings
void	`setComponentNumber(int i)` Setter for the component number.
void	`setCreationVersion(int major,int minor)` Sets this object's creation version to the given numbers.

void	`setDataState(int ds)` Sets this object's data state.
void	`setDB_ID(int dbid)` Sets this object's DB_ID
void	`setDeleted(boolean del)` Sets this object `deleted`and thus inoperative.
void	`setDescription(String desc)` Sets this object's description to the given desc.
void	`setIdent(int uid)` Sets the unique ident of this object.
void	`setMajorCreationVersion(int version)` Sets this object's creation version to the given version.
void	`setMajorVersion(int major_tag)` Sets this objects current major revision number.
void	`setMinorCreationVersion(int version)` Sets this object's creation version to the given version.
void	`setMinorVersion(int minor_tag)` Sets this objects current minor revision number.
void	`setName(String name)` Sets this object's name to the given name after trimming it.
String	`showComment(int num)` Retrieves the text in the comment at the given index.
void	`storeState(Hashtable state)` Stores the state of the object to permit undo by cloning itself and storing the clone.
void	`storeState(String prefix,Hashtable state)` Store the state of the bean to permit undo.
static void	`trace(String message)` Prints the given message to stderr in TRACE format.
void	`updateVersion()` Updates this object's current version to be greater than or equal to it's model's version.
void	`validate()` Determines this object's general data state by examining it's internal data.
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.

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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
static void	writeSP(java.io.PrintWriter out,String text,int width) 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 it's 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 containging 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.

setIdent

```
public void setIdent(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 propigate 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 valdiate 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 overriden 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 revison 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 ths 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 mesage 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 mesage 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 mesage 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 mesage 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 mesage 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 mesage 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	`reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)` 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 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

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

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.
void	`setModel(AbstractModel model)` 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 extention 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	`SpecialConnectionData()` Creates a new instance of SpecialConnectData
public	`SpecialConnectionData(int index)` Creates a new instance of SpecialConnectData with the given index.

Method Summary

Object	`clone()`
String	`toString()`

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 ValidationOptions

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

All Implemented Interfaces:
StateEditable

---

public class **ValidationOptions**
extends Object
implements StateEditable

This class is used to store all of a model's validation options into MED. This should be a property of each model for a plugin. Model specific options are stored in parallel string arrays.

Constructor Summary

public	`ValidationOptions()` Creates a new instance of ValidationOptions

Method Summary

void	`addOption(String key,String option)` Adds a new option to be stored.
boolean	`dumpBlockParams(java.io.PrintWriter dumpFile)` Simple dumpBlockParams added so that ValidationOptions can be subblocks.
String[]	`getData()` This gets the current data array from this options.
String[]	`getKeys()` This gets the current keys array from this options.
String	`getOption(String key,String def)` Attempts to find an option with the given key.
boolean	`readBlock(com.apt.xdr.PibFile pibFile)` Simple readBlock added so that ValidationOptions can be subblocks.
boolean	`readBlockParams(com.apt.xdr.PibFile pibFile)` Simple readBlockParams added so that ValidationOptions can be subblocks.
boolean	`readBlockParams(com.apt.xdr.PibFile pibFile,int[] blockparm)` Simple readBlockParams added so that ValidationOptions can be subblocks.
void	`restoreState(Hashtable state)`

void	`restoreState(String prefix,Hashtable state)` Loads the current state of this ValidationOptions from a hashtable.
void	`setData(String[] data)` This sets the current data array for this options.
void	`setKeys(String[] keys)` This sets the current keys array for this options.
void	`setPrefix(String prefix)` This method is called by ValidationTest in load and store option.
void	`storeState(Hashtable state)`
void	`storeState(String prefix,Hashtable state)` Stores the current state of this ValidationOptions to a hashtable.
boolean	`writeBlockParams(com.apt.xdr.PibFile pibFile)` Simple writeBlockParams added so that ValidationOptions can be subblocks.

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ValidationOptions

`public` **`ValidationOptions`**`()`

Creates a new instance of ValidationOptions

Methods

addOption

```
public void addOption(String key,
         String option)
```

Adds a new option to be stored. This option will be associated with the given key. If the key already exists, the data in option will overwrite what currently exists.

getOption

```
public String getOption(String key,
         String def)
```

Attempts to find an option with the given key. If no option is found the default is returned. The current prefix will be prepended to the key.

setPrefix

```
public void setPrefix(String prefix)
```

(continued from last page)

This method is called by ValidationTest in load and store option. This sets the current prefix. This prefix is used to prevent data from overwriting between ValidationTests. The prefix used is the name of the current ValidationTest, and is why the ValidationTests for a given plugin must have unique names.

getKeys

```
public String[] getKeys()
```

This gets the current keys array from this options. This should be used only for MED file saving code. Any changes to this array will result in lost data.

setKeys

```
public void setKeys(String[] keys)
```

This sets the current keys array for this options. This should be used only for MED file loading code. Any changes to this array will result in lost data.

getData

```
public String[] getData()
```

This gets the current data array from this options. This should be used only for MED file saving code. Any changes to this array will result in lost data.

setData

```
public void setData(String[] data)
```

This sets the current data array for this options. This should be used only for MED file loading code. Any changes to this array will result in lost data.

readBlock

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

Simple readBlock added so that ValidationOptions can be subblocks.

dumpBlockParams

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

Simple dumpBlockParams added so that ValidationOptions can be subblocks.

writeBlockParams

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

Simple writeBlockParams added so that ValidationOptions can be subblocks.

readBlockParams

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

Simple readBlockParams added so that ValidationOptions can be subblocks.

readBlockParams

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

Simple readBlockParams added so that ValidationOptions can be subblocks.

(continued from last page)

storeState

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

Stores the current state of this ValidationOptions to a hashtable. This ensures that this object is state editable

---

restoreState

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

Loads the current state of this ValidationOptions from a hashtable. This ensures that this object is state editable

---

restoreState

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

---

storeState

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

com.cafean.client.analysis
Class ValidationTest

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

public abstract class **ValidationTest**
extends Object

This class represents a model-level test that is executed when the plugin verifies that a model's data is valid.

Field Summary

static int	VERBOSITY_HIGH Verbose output level.
static int	VERBOSITY_LOW Normal output message level.
static int	VERBOSITY_QUIET Few output messages.

Constructor Summary

public	ValidationTest()

Method Summary

boolean	canRepair() Determines if this validation test can be self correcting.
abstract String	getDisplayName() This returns a short String that is a human readable name for this validation test.
abstract String	getName() Gets the name of the specified Test.
abstract String	getShortDescription() This returns a long, detailed description of this validation test, its user options, and background information.
int	getVerbosity() Gets the verbosity level for this test.
boolean	isEnabled() Determines if this validation test is enabled for the current model.

void	`loadOptions(ValidationOptions options)` Loads the options for a validationTest from the given ValidationOptions object.
void	`repair()` If this validation test can self-correct, this is where it does so.
abstract boolean	`runValidation(boolean printErrors)` This method is the actual validation.
void	`saveOptions(ValidationOptions options)` Stores the options for a validationTest to the given ValidationOptions object.
void	`setEnabled(boolean enabled)` Sets this validation test enabled.
void	`setVerbosity(int verbosity)` Sets the verbosity level for this test.
String	`toString()` Returns a short string representing this test

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

VERBOSITY_QUIET

`public static int` **`VERBOSITY_QUIET`**

Few output messages.

VERBOSITY_LOW

`public static int` **`VERBOSITY_LOW`**

Normal output message level.

VERBOSITY_HIGH

`public static int` **`VERBOSITY_HIGH`**

Verbose output level.

Constructors

ValidationTest

`public` **`ValidationTest`**`()`

Methods

(continued from last page)

getName

`public abstract String getName()`

Gets the name of the specified Test. The test names must be unique for a given plugin.

---

loadOptions

`public void loadOptions(ValidationOptions options)`

Loads the options for a validationTest from the given ValidationOptions object. Options are stored as a String with another String given as the "key" to that option. When this class is overwritten, the override for this method should always call super loadOptions. This will be called right before the validationTest executes.

Parameters:

`options` - the ValidationOptions from which data will be loaded.

---

saveOptions

`public void saveOptions(ValidationOptions options)`

Stores the options for a validationTest to the given ValidationOptions object. Options are stored as a String with another String given as the "key" to that option. When this class is overwritten, the override for this method should always call super saveOptions. This should be called after the options are edited by the user.

Parameters:

`options` - the ValidationOptions where data will be saved.

---

getDisplayName

`public abstract String getDisplayName()`

This returns a short String that is a human readable name for this validation test.

---

getShortDescription

`public abstract String getShortDescription()`

This returns a long, detailed description of this validation test, its user options, and background information.

---

runValidation

`public abstract boolean runValidation(boolean printErrors)`

This method is the actual validation. If this returns false, the model will be considered to have failed.

Parameters:

`printErrors` - a boolean flag that should enable writing error messages to the MainFrame.

Returns:

true if the validation ran and discovered no errors.

---

isEnabled

`public boolean isEnabled()`

Determines if this validation test is enabled for the current model. This should be an option stored in #saveOption and #loadOption.

---

setEnabled

`public void setEnabled(boolean enabled)`

(continued from last page)

Sets this validation test enabled.

getVerbosity

`public int` **`getVerbosity`**`()`

Gets the verbosity level for this test.

setVerbosity

`public void` **`setVerbosity`**`(int verbosity)`

Sets the verbosity level for this test. Acceptable values are: `VERBOSITY_NONE` `VERBOSITY_LOW` `VERBOSITY_HIGH`

canRepair

`public boolean` **`canRepair`**`()`

Determines if this validation test can be self correcting. This is currently not supported. The default is false.

repair

`public void` **`repair`**`()`

If this validation test can self-correct, this is where it does so. Any user interaction should be specified here. This is currently not supported. The default is no operations.

toString

`public String` **`toString`**`()`

Returns a short string representing this test

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() Creates a new instance of ViewComponent
public	ViewComponent(AbstractModel model,int num) Creates a new instance of ViewComponent with a new model and display number

Method Summary

void	addAnnotation(Annotation comp) Adds an `Annotation` to this ViewComponent, or to the `DrawnView` if the view has been opened.
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.

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.
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.
void	`addPibBlock(com.apt.xdr.PibBlock block)` Adds a new Pibblock to this View Component.
void	`buildView()` This initializes a new DrawnView based on a View Component.
boolean	`canConnectTo(AbstractComponent target)` This method checks to see if it is allowable for this AbstractComponent to connect to the given target.
void	`clearViewSelection()` Clears the current selection on the View if it exists
Object	`clone()` this clone method copies all primitive data types.
void	`complete()` When a ViewComponent is created, it immediately opens.
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.
void	`copyFrom(GenericObject o)` Copy the attributes from a clone to this instance.
DrawnComponent	`createDrawnComponent()` {@inheritDoc}
void	`fireComponentChanged()` This calls the component changed function on all of the listeners currently listening to this abstract component
void	`fireComponentDeleted()` This calls the component deleted function on all of the listeners currently listening to this abstract component
byte[]	`getAnnotationArray()` Gets an byte array containing the XML encoded stream of `annotations` from this DrawnView.
static String	`getAttributeGroup(String property)` Retrieves the name of the Attribute Group for the given property name.
static String[]	`getAttributeGroups()` Retrieves the Attribute Group Names for this View Component.

int	getAttributeIndex(String propertyName)
static String[]	getAttributesForGroup(String groupName) Retrieves the Attribute Names for the given group.
java.awt.Color	getBackground() Gets the background color from the DrawnView, or from the local variable if the DrawnView has not been created.
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.
Category	getCategory()
String	getCCnumber()
int[]	getColorArray() This returns the current background in the form of an array of three integers.
Vector	getCustomPopupItems() Creates Custom Menu Items for any popup dialog involving this component
DrawnView	getDrawnView() A getter for the DrawnView owned by this ViewComponent.
byte[]	getEmbedConnArray() Gets an byte array containing the XML encoded stream of `annotations` from this DrawnView.
EmbeddedConnectionMap	getEmbeddedCons()
java.awt.Color	getGridLineColor() Getter for property lineColor.
int	getHorizGridSpacing() Getter for property horizGridSpacing.
int	getHorizSnapSpacing() Getter for property horizSnapSpacing.
ImageIcon	getImage() Gets the ImageIcon for rendering this ViewComponent.
byte[]	getImageData() Gets the icon image for the `DrawnViewComponent` that renders this ViewComponent.
ViewImgDataElem	getImageElement() A stub method used in ViewComponentBeanInfo to edit the image data byte[] contained in this ViewComponent.
java.awt.Point	getLocation() Gets the location of the `DrawnView` or the locally stored location, if the DrawnView has never been opened.

int	`getPixelsPerMeter()` Getter for property pixelsPerMeter.
boolean	`getShowPoints()` Returns whether or not the connection points are drawn on AbstractComponents inside this view.
java.awt.Dimension	`getSize()` Gets the size of the `DrawnView` or the locally stored size, if the DrawnView has never been opened.
float	`getTransparency()` Getter for property transparency.
int	`getVertGridSpacing()` Getter for property vertGridSpacing.
int	`getVertSnapSpacing()` Getter for property vertSnapSpacing.
java.awt.Point	`getViewPosition()` Getter for property viewPosition.
double	`getWidthScaleFactor()` Gets this view's width scale factor.
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.
boolean	`hasBlocks()` Determines whether this view has anything inside it's blocks vector.
boolean	`isGridAbove()` Returns true if this View's grid will be painted on top of its contained elements.
boolean	`isLocked()` 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()` Getter for property showGrid.

boolean	`isSnapToGrid()` Getter for property snapToGrid.
boolean	`isViewVisible()` 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()` Runs the organize function on the view component if it exists.
void	`load(ViewCompRec rec)` Loads this view's data from the given ViewCompRec previously stored with `#store`
static java.awt.Component	`loadComponent(com.apt.xdr.PibBlock block,AbstractModel model)`
void	`popupDataDialog(java.awt.Window parent,boolean modal)`
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.
void	`readEmbedConnArray(byte[] byteArr)` 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)` 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)` Sets the locally stored canvas size.
void	`setColorArray(int[] array)` Sets the current background from an array of three integers.
void	`setEmbeddedCons(EmbeddedConnectionMap cons)`
void	`setGridAbove(boolean above)` If set to true, this View's grid will be painted on top of its contained elements.
void	`setGridLineColor(java.awt.Color lineColor)` Setter for property lineColor.
void	`setHorizGridSpacing(int horizGridSpacing)` Setter for property horizGridSpacing.

void	`setHorizSnapSpacing(int horizSnapSpacing)` Setter for property horizSnapSpacing.
void	`setImageData(byte[] imageData)` Sets the icon image for the `DrawnViewComponent` that renders this ViewComponent.
void	`setImageElement(ViewImgDataElem element)` A stub method used in ViewComponentBeanInfo to edit the image data byte[] contained in this ViewComponent.
void	`setLocation(java.awt.Point location)` Sets the the locally stored location of the DrawnView.
void	`setLocked(boolean locked)` sets this view's locked state
void	`setLockedConstrained(boolean locked)` sets this view's locked state
void	`setPixelsPerMeter(int pixelsPerMeter)` Setter for property pixelsPerMeter.
void	`setShowGrid(boolean showGrid)` Setter for property showGrid.
void	`setShowPoints(boolean showPoints)` Turns on displaying the connection points on AbstractComponents inside this view.
void	`setSize(java.awt.Dimension size)` Sets the size of the `DrawnView` or the locally stored size, if the DrawnView has never been opened.
void	`setSnapToGrid(boolean snapToGrid)` Setter for property snapToGrid.
void	`setStoredBackground(java.awt.Color background)` This just sets the background color of the stored local variable.
void	`setTransparency(float transparency)` Setter for property transparency.
void	`setVertGridSpacing(int vertGridSpacing)` Setter for property vertGridSpacing.
void	`setVertSnapSpacing(int vertSnapSpacing)` Setter for property vertSnapSpacing.
void	`setViewPosition(java.awt.Point viewPosition)` Setter for property viewPosition.
void	`setVisible(boolean value)` Shows or hides this ViewComponent's DrawnView and it's accompanying dialog or panel.

void	`setWidthScaleFactor(double factor)` Sets this view's width scale factor.
void	`setZoomScale(double zoomScale)` Sets the locally stored zoom scale.
boolean	`store(com.apt.xdr.PibFile file)` Writes this view and it's contents to the given PibFile.
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.
String	`toString()` ViewComponents use their name as their to-string, as opposed to including "View CCnumber" before their name.
void	`undockView()` 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, 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

(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.apt.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 Arrangment.

---

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.apt.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 View Component.

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 containg 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.analysis.numerics

com.cafean.client.analysis.numerics
Class FunctionSource

```
java.lang.Object
   |
   +-com.cafean.client.analysis.numerics.FunctionSource
```

public class **FunctionSource**
extends Object

Holds the source code for the user defined function along with a reference to the UDF.

Constructor Summary

public	`FunctionSource()` Creates a new instance of FunctionSource
public	`FunctionSource(ComponentElement owner)` Construct an empty instance.
public	`FunctionSource(String sourceCode,ComponentElement owner)` Construct an empty instance.

Method Summary

ComponentElement	`getOwner()` Getter for property owner.
String	`getSourceCode()` Getter for property sourceCode.
void	`setOwner(ComponentElement owner)` Setter for property owner.
void	`setSourceCode(String sourceCode)` Setter for property sourceCode.

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

FunctionSource

`public` **`FunctionSource`**`()`

Creates a new instance of FunctionSource

FunctionSource

`public FunctionSource(ComponentElement owner)`

Construct an empty instance.

Parameters:

`owner` - the owner of this subelement.

FunctionSource

```
public FunctionSource(String sourceCode,
                      ComponentElement owner)
```

Construct an empty instance.

Parameters:

`owner` - the owner of this subelement.

Methods

getSourceCode

`public String getSourceCode()`

Getter for property sourceCode.

Returns:

Value of property sourceCode.

setSourceCode

`public void setSourceCode(String sourceCode)`

Setter for property sourceCode.

Parameters:

`sourceCode` - New value of property sourceCode.

getOwner

`public ComponentElement getOwner()`

Getter for property owner.

Returns:

Value of property owner.

setOwner

`public void setOwner(ComponentElement owner)`

Setter for property owner.

Parameters:

`owner` - New value of property owner.

com.cafean.client.analysis.numerics
Class ParametricIteration

```
java.lang.Object
   |
   +-com.cafean.client.analysis.numerics.ParametricIteration
```

public class **ParametricIteration**
extends Object

This will perform the iteration used for exporting a series of ASCII runs.

Field Summary

static final int	NAME_NUMBER Indicates that the file name will include the iteration point for each constant used. Value: **0**
static final int	NAME_VALUE Indicates that the file name will include the name and value for each constant used. Value: **1**

Constructor Summary

public	ParametricIteration() Creates a new instance of ParametricIteration
public	ParametricIteration(UserDefinedConstant[] constants,String baseFile,int convention)

Method Summary

boolean	hasNext() Determines if there's more iteration to be done.
String	next() Iterates through the Constants and gets the next file name.
void	resetIteration() Resets this iterator
void	setBaseFile(String baseFile) Sets the basefile name.
void	setConstants(UserDefinedConstant[] constants) Sets the constants.

void	`setModel(AbstractModel model)` Sets the model on this iteration.
void	`setNameConvention(int convention)` Sets the naming convention for determining the name of each file.
String	`writeConstantTable(UserDefinedConstant constant)` Writes out the value table of a tabular parametric constant.
String	`writeDescription()` Writes the export description of this model.
String	`writeExportNote(int count)` Builds an export note for the current state of the parametric export.
String	`writeName(String name)`
String	`writeTableHeader()` Writes the export file table header.
String	`writeTableLine(int count,String filename,String filePath)` Returns a single line of the export file table in HTML format.

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

NAME_NUMBER

`public static final int` **`NAME_NUMBER`**

Indicates that the file name will include the iteration point for each constant used.

NAME_VALUE

`public static final int` **`NAME_VALUE`**

Indicates that the file name will include the name and value for each constant used. This may produce large file names.

Constructors

ParametricIteration

`public` **`ParametricIteration`**`()`

Creates a new instance of ParametricIteration

ParametricIteration

```
public ParametricIteration(UserDefinedConstant[] constants,
                           String baseFile,
                           int convention)
```

(continued from last page)

Methods

setModel

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

Sets the model on this iteration.

setBaseFile

`public void **setBaseFile**(String baseFile)`

Sets the basefile name.

setConstants

`public void **setConstants**(UserDefinedConstant[] constants)`

Sets the constants.

setNameConvention

`public void **setNameConvention**(int convention)`

Sets the naming convention for determining the name of each file.

resetIteration

`public void **resetIteration**()`

Resets this iterator

hasNext

`public boolean **hasNext**()`

Determines if there's more iteration to be done.

next

`public String **next**()`

Iterates through the Constants and gets the next file name.

writeExportNote

`public String **writeExportNote**(int count)`

Builds an export note for the current state of the parametric export. This note is set inside the model before each export.

writeName

`public String **writeName**(String name)`

writeDescription

`public String **writeDescription**()`

Writes the export description of this model.

writeTableHeader

```
public String writeTableHeader()
```

Writes the export file table header. This returns a String in HTML format for defining the table header.

writeTableLine

```
public String writeTableLine(int count,
         String filename,
         String filePath)
```

Returns a single line of the export file table in HTML format.

writeConstantTable

```
public String writeConstantTable(UserDefinedConstant constant)
```

Writes out the value table of a tabular parametric constant. If the passed constant is incremental this will return an empty string.

com.cafean.client.analysis.numerics
Class UserDefinedConstant

```
java.lang.Object
   |-com.cafean.client.analysis.GenericObject
      |-com.cafean.client.analysis.AbstractComponent
         |-com.cafean.client.analysis.numerics.UserDefinedValue
            |-com.cafean.client.analysis.numerics.UserDefinedConstant
```

All Implemented Interfaces:
PropertyController, IdentHolder, StateEditable, Cloneable, Checkable, ComponentElement, Cloneable

---

public class **UserDefinedConstant**
extends UserDefinedValue
implements Cloneable, ComponentElement, Checkable, Cloneable, StateEditable, IdentHolder, PropertyController

The UserDefinedConstant serves as a UserDefinedValue that may be parametric. This may be used as an input variable for a `UserDefinedFunction` and it may be refered to directly by a `Real`. A parametric constant has an array of values, and a default index. When the constant is not the parameter for a parametric ASCII export or a parametric job submission, the current value is always the value at the default index.

Fields inherited from class `com.cafean.client.analysis.GenericObject`
`DATA_COMPLETE`, `DATA_ERROR`, `DATA_INCOMPLETE`, `DATA_WARNING`

Constructor Summary

public	`UserDefinedConstant()` Creates a new UserDefinedConstant
public	`UserDefinedConstant(AbstractModel model,int componentNumber)` Creates a new instance of UserDefinedConstant inside a model with a specified component number.

Method Summary

void	`addParameter(Real parameter)` Appends a new parameter to the end of the parameter list.
Object	`clone()` {@inheritDoc}
void	`complete()`
void	`copyFrom(GenericObject object)` {@inheritDoc}

int	`getAttributeIndex(String property)` Returns a relative index that can be used to order property lists.
Category	`getCategory()` The category for a UserDefinedConstant is always `CAT_CONSTANT`
int	`getCurrentIndex()` This gets the transient value used for determining where inside a parametric iteration this constant is.
boolean	`getCurrentParametric()` Gets whether this constant is the current parameter or not
Real	`getCurrentValue()` If this constant is the current parameter of a parametric run or job submission, this returns the value at the current index.
int	`getDefaultIndex()` Gets the default index inside this constant.
Real	`getDefaultValue()` Gets the default value of this constant.
Real	`getEndValue()` Gets the end value of this constant.
Real	`getIncrement()` Gets the increment of this constant.
Real	`getParameterAt(int index)` Returns the parameter at a given index
Real[]	`getParameters()`
int	`getParametersCount()` Gets the number of parameters in this constant.
String	`getRunValues()` Returns a short string that indicates this constants current values for a parametric export.
Real	`getStartValue()` Gets the start value of this constant.
Real	`getUnits()`
boolean	`hasNext()` Determines if the parameter has processed all of its parameters during parametric export or job submission.
boolean	`isIncremental()` Gets the flag that indicates if this parametric constant is incremental.

boolean	`isParametric()` Determines if this constant is defined as a parametric constant that has multiple values or not.
boolean	`isPropertyActive(String propertyName)` Returns false if this object has a property with the given name that is considered inactive; otherwise true.
boolean	`isPropertyEnabled(String propertyName)` Returns false if this object has a property with the given name that has dependency code that fails;true otherwise
boolean	`isPropertyRequired(String propertyName)` Returns false if this object has a property with the given name that has requirement code that fails; true otherwise.
boolean	`isPropertyResizable(String propertyName)` Returns false if this object has an array property with the given name that should not normally be resizable.
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.
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.
void	`iterate()` Iterates the current index inside a parameter during parametric export or job submission.
String	`label()` {@inheritDoc}
void	`removeParameter(int index)` Removes the parameter at the given index.
void	`resetIteration()` Resets the current index of a parameter during parametric export or job submission.
void	`restoreState(Hashtable state)`
void	`setCurrentParametric(boolean currentParametric)` Sets this constant as the current parameter during a parametric export or job submission.
void	`setDefaultIndex(int defaultIndex)` Sets the default index inside this constant.
void	`setDefaultValue(Real val_)` Sets the default value of this constant.
void	`setEndValue(Real val_)` Sets the end value of this constant.

void	setIncrement(Real val_) Sets the increment of this constant.
void	setIncremental(boolean val_) Sets the flag that indicates if this parametric constant is incremental.
void	setParameter(Real parameter) Setter for the value in a non-parametric constant.
void	setParameterAt(int index,Real parameter) Sets a parametric value at the given index.
void	setParameters(Real[] parameters)
void	setParametric(boolean parametric) Sets this constant to be parametric with multiple values or constant with only one.
void	setStartValue(Real val_) Sets the start value of this constant.
void	setUnits(Real units) This sets all of the values inside this constant to have the same units as the Real passed in.
void	storeState(Hashtable state)
String	toString() {@inheritDoc}

Methods inherited from class com.cafean.client.analysis.numerics.UserDefinedValue

createDrawnComponent, fireComponentDeleted, getCurrentValue, getUnits, popupDataDialog, replaceNumericNames, restoreState, setName, setUnits, setUnits

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, 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, writeMuxLoadArray, writeMuxLoadArray, writeSP, writeSP

Methods inherited from class `java.lang.Object`

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

Constructors

UserDefinedConstant

`public` **`UserDefinedConstant`**`()`

Creates a new UserDefinedConstant

UserDefinedConstant

`public` **`UserDefinedConstant`**`(AbstractModel model, int componentNumber)`

Creates a new instance of UserDefinedConstant inside a model with a specified component number.

Parameters:

`model` - the AbstractModel.

`number` - the component number for this function.

Methods

copyFrom

`public void` **`copyFrom`**`(GenericObject object)`

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!**

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.

getCurrentValue

`public Real getCurrentValue()`

If this constant is the current parameter of a parametric run or job submission, this returns the value at the current index. Otherwise, this returns the value at the default index.

Returns:
Real the current value.

getCurrentIndex

`public int getCurrentIndex()`

This gets the transient value used for determining where inside a parametric iteration this constant is. This is used when exporting a parametric run.

getCategory

`public Category getCategory()`

The category for a UserDefinedConstant is always `CAT_CONSTANT`

Returns:
the Category for this UserDefinedConstant.

toString

`public String toString()`

label

`public String label()`

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

getParametersCount

`public int getParametersCount()`

Gets the number of parameters in this constant.

Returns:
The number of values in the parameter array.

getParameterAt

`public Real getParameterAt(int index)`

Returns the parameter at a given index

Parameters:
`index` - the index of the parameter.

Returns:
the Real value at the given index.

(continued from last page)

setParameters

```
public void setParameters(Real[] parameters)
```

getParameters

```
public Real[] getParameters()
```

setParameterAt

```
public void setParameterAt(int index,
          Real parameter)
```

Sets a parametric value at the given index.

Parameters:

`index` - the index.
`parameter` - the Real parameter to be set.

addParameter

```
public void addParameter(Real parameter)
```

Appends a new parameter to the end of the parameter list.

Parameters:

`parameter` - the Real that will be appended.

removeParameter

```
public void removeParameter(int index)
```

Removes the parameter at the given index. This will not delete the last parameter.

Parameters:

`index` - the index of the parameter to be removed.

iterate

```
public void iterate()
```

Iterates the current index inside a parameter during parametric export or job submission.

resetIteration

```
public void resetIteration()
```

Resets the current index of a parameter during parametric export or job submission.

hasNext

```
public boolean hasNext()
```

Determines if the parameter has processed all of its parameters during parametric export or job submission.

Returns:

true if the current index is less then the number of parameters.

(continued from last page)

getDefaultIndex

`public int` **`getDefaultIndex`**`()`

Gets the default index inside this constant.

Returns:

The index used to determine the value outside of a parametric export.

---

setDefaultIndex

`public void` **`setDefaultIndex`**`(int defaultIndex)`

Sets the default index inside this constant.

Parameters:

`defaultIndex -` The index used to determine the value outside of a parametric export.

---

setCurrentParametric

`public void` **`setCurrentParametric`**`(boolean currentParametric)`

Sets this constant as the current parameter during a parametric export or job submission.

Parameters:

`currentParametric -` whether or not this constant is the current parameter.

---

getCurrentParametric

`public boolean` **`getCurrentParametric`**`()`

Gets whether this constant is the current parameter or not

---

isParametric

`public boolean` **`isParametric`**`()`

Determines if this constant is defined as a parametric constant that has multiple values or not.

Returns:

true if this constant is set parametric.

---

setParametric

`public void` **`setParametric`**`(boolean parametric)`

Sets this constant to be parametric with multiple values or constant with only one.

Parameters:

`parametric -` whether or not this constant has multiple values.

---

isIncremental

`public boolean` **`isIncremental`**`()`

Gets the flag that indicates if this parametric constant is incremental.

---

setIncremental

`public void` **`setIncremental`**`(boolean val_)`

Sets the flag that indicates if this parametric constant is incremental.

getDefaultValue

`public Real getDefaultValue()`

Gets the default value of this constant.

setDefaultValue

`public void setDefaultValue(Real val_)`

Sets the default value of this constant.

getStartValue

`public Real getStartValue()`

Gets the start value of this constant.

setStartValue

`public void setStartValue(Real val_)`

Sets the start value of this constant.

getEndValue

`public Real getEndValue()`

Gets the end value of this constant.

setEndValue

`public void setEndValue(Real val_)`

Sets the end value of this constant.

getIncrement

`public Real getIncrement()`

Gets the increment of this constant.

setIncrement

`public void setIncrement(Real val_)`

Sets the increment of this constant.

getUnits

`public Real getUnits()`

setUnits

`public void setUnits(Real units)`

This sets all of the values inside this constant to have the same units as the Real passed in.

Parameters:

`units` - the Real containing the units to be set.

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

getAttributeIndex

`public int` **`getAttributeIndex`**`(String property)`

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:

`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

isPropertyActive

`public boolean` **`isPropertyActive`**`(String propertyName)`

Returns false if this object has a property with the given name that is considered inactive; otherwise true.

Parameters:

(continued from last page)

`propertyName` - a String containing the name of the property to check

Returns:

false if the property is inactive, true otherwise

setParameter

`public void` **`setParameter`**`(Real parameter)`

Setter for the value in a non-parametric constant.

storeState

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

restoreState

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

complete

`public void` **`complete`**`()`

getRunValues

`public String` **`getRunValues`**`()`

Returns a short string that indicates this constants current values for a parametric export.

com.cafean.client.analysis.numerics
Class UserDefinedFunction

```
java.lang.Object
   |-com.cafean.client.analysis.GenericObject
      |-com.cafean.client.analysis.AbstractComponent
         |-com.cafean.client.analysis.numerics.UserDefinedFunction
```

public class **UserDefinedFunction**
extends AbstractComponent

This is the class that allows the user to write a block of python code that will be executed by the python interpreter. The function can have multiple inputs, that are either UserDefinedConstants or UserDefinedVariables, and multiple outputs that are UserDefinedVariables.

Fields inherited from class com.cafean.client.analysis.GenericObject
DATA_COMPLETE, DATA_ERROR, DATA_INCOMPLETE, DATA_WARNING

Constructor Summary

public	`UserDefinedFunction()` Creates a new instance of UserDefinedFunctions
public	`UserDefinedFunction(AbstractModel model,int number)` Creates a new instance of UserDefinedFunction inside a model with a specified component number.

Method Summary

void	`addInputId(int id)` Adds the UserDefinedValue with the given ident into the input array, if the value is not already in the input array.
void	`addOutputId(int id)` Adds an output variable's ident to the output array if it doesn't already exist.
void	`clearInputs()` Removes all the entries in the input array
void	`clearOutputs()` This removes all of the variables from the output array.
Object	`clone()`
void	`complete()`

void	copyFrom(GenericObject object)
DrawnComponent	createDrawnComponent() UserDefinedFunctions are not visual components.
boolean	execute() Executes the python code in this function.
static UserDefinedConstant	findConstant(AbstractModel model,String name) Finds the UserDefinedConstant in the given model with the given name.
static UserDefinedFunction	findFunction(AbstractModel model,int ident) Finds the UserDefinedFunction in the given model with the given name.
static UserDefinedVariable	findVariable(AbstractModel model,String name) Finds the UserDefinedVariable in the given model with the given name.
Category	getCategory() The category for a UserDefinedFunction is always CAT_FUNCTION
Vector	getCustomPopupItems()
FunctionSource	getFunction() Returns the current python code for this function.
UserDefinedValue	getInputAt(int index) Returns the UserDefinedValue at the given index in the input array.
int	getInputCount() Gets the number of input variables.
int	getInputIdAt(int index) Gets the ident of the input varible at the given index.
double	getInputValue(int index) Returns the current value of the input at a given index.
double	getInputValue(String name) Returns the current value of the input variable with the given name.
UserDefinedVariable	getOutputAt(int index) Gets the UserDefinedVariable from the output array at a given index.
int	getOutputCount() Returns the number of variables this function outputs to.
int	getOutputIdAt(int index) Returns the ident stored at a given index in the output array.
String	getPythonHeader() Returns the header that is prepended to the python code before execution.

static String	getPythonInit() Gets the python initialization string.
String	label()
void	popupDataDialog(java.awt.Window parent,boolean modal)
void	reconnectIdentReferences(boolean preserveUnresolved,boolean useDbId)
void	removeInputAt(int index) Removes the input at the given index.
void	removeOutputAt(int index) Removes a the variable at the given index from the output array.
void	setFunction(FunctionSource function) Sets the python code for this function.
void	setInputIdAt(int index,int id) Sets the ident of the UserDefinedValue to be stored inside the input array at a given index.
void	setOutputIdAt(int index,int id) Sets the ident of an output at the given index.
void	setOutputValue(int index,double value) Sets the SI value given on the output variable at the given index.
void	setOutputValue(String name,double value) Sets the SI value on on the output with the given name.
String	toString()
void	validate()

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, 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`

Constructors

UserDefinedFunction

```
public UserDefinedFunction()
```

Creates a new instance of UserDefinedFunctions

UserDefinedFunction

```
public UserDefinedFunction(AbstractModel model,
                           int number)
```

Creates a new instance of UserDefinedFunction inside a model with a specified component number.

Parameters:

`model` - the AbstractModel.
`number` - the component number for this function.

Methods

popupDataDialog

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

clone

```
public Object clone()
```

copyFrom

```
public void copyFrom(GenericObject object)
```

reconnectIdentReferences

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

getCustomPopupItems

```
public Vector getCustomPopupItems()
```

getCategory

```
public Category getCategory()
```

The category for a UserDefinedFunction is always `CAT_FUNCTION`

Returns:

the Category for this UserDefinedFunction.

createDrawnComponent

```
public DrawnComponent createDrawnComponent()
```

UserDefinedFunctions are not visual components. Therefore they always return null for when creating a DrawnComponent.

Returns:

null

getOutputCount

```
public int getOutputCount()
```

Returns the number of variables this function outputs to.

Returns:

the integer size of the output array.

getOutputIdAt

```
public int getOutputIdAt(int index)
```

Returns the ident stored at a given index in the output array.

Parameters:

`index` - the index into the output array.

Returns:

the ident stored at the given index.

setOutputIdAt

```
public void setOutputIdAt(int index,
          int id)
```

Sets the ident of an output at the given index.

Parameters:

`index` - the index into the output array.

(continued from last page)

`id` - the ident of the output variable.

addOutputId

`public void addOutputId(int id)`

Adds an output variable's ident to the output array if it doesn't already exist.

Parameters:

`id` - the ident of the output variable to be added.

clearOutputs

`public void clearOutputs()`

This removes all of the variables from the output array.

removeOutputAt

`public void removeOutputAt(int index)`

Removes a the variable at the given index from the output array.

Parameters:

`index` - the index into the output array.

setOutputValue

```
public void setOutputValue(int index,
                           double value)
```

Sets the SI value given on the output variable at the given index.

Parameters:

`index` - the integer index into the output array.
`value` - the SI value being set on the output variable.

getOutputAt

`public UserDefinedVariable getOutputAt(int index)`

Gets the UserDefinedVariable from the output array at a given index.

Parameters:

`index` - the integer index into the output array.
`the` - UserDefinedVariable stored at the given index.

setOutputValue

```
public void setOutputValue(String name,
                           double value)
```

Sets the SI value on on the output with the given name.

Parameters:

`name` - the name of the output variable.
`value` - the SI value to be set.

getInputCount

`public int getInputCount()`

Gets the number of input variables.

(continued from last page)

Returns:
the size of the input array.

getInputIdAt

`public int getInputIdAt(int index)`

Gets the ident of the input varible at the given index.

Parameters:
`index` - the integer index into the input array.
`the` - ident of the UserDefinedValue in the input array.

setInputIdAt

`public void setInputIdAt(int index, int id)`

Sets the ident of the UserDefinedValue to be stored inside the input array at a given index.

Parameters:
`index` - the integer index into the input array.
`id` - of the UserDefinedVariable to be set in the array.

addInputId

`public void addInputId(int id)`

Adds the UserDefinedValue with the given ident into the input array, if the value is not already in the input array.

Parameters:
`id` - the ident of the UserDefinedValue to be added.

clearInputs

`public void clearInputs()`

Removes all the entries in the input array

removeInputAt

`public void removeInputAt(int index)`

Removes the input at the given index.

Parameters:
`index` - the index into the input array.

getInputValue

`public double getInputValue(int index)`

Returns the current value of the input at a given index.

Parameters:
`index` - the integer index into the input array.

getInputAt

`public UserDefinedValue getInputAt(int index)`

Returns the UserDefinedValue at the given index in the input array.

(continued from last page)

Parameters:
`index` - the integer index into the input array.

Returns:
the UserDefinedValue at the index.

getInputValue

`public double` **`getInputValue`**`(String name)`

Returns the current value of the input variable with the given name. If no input variable with that name is found, this returns `Real#Unknown`

Parameters:
`name` - the String containing the name of the input variable.

getFunction

`public` `FunctionSource` **`getFunction`**`()`

Returns the current python code for this function.

Returns:
the String containing the python code.

setFunction

`public void` **`setFunction`**`(FunctionSource function)`

Sets the python code for this function.

Parameters:
`function` - the String containing the python code.

toString

`public String` **`toString`**`()`

label

`public String` **`label`**`()`

execute

`public boolean` **`execute`**`()`

Executes the python code in this function.

Returns:
the success or failure of this execution.

findConstant

`public static` `UserDefinedConstant` **`findConstant`**`(``AbstractModel` `model, String name)`

Finds the UserDefinedConstant in the given model with the given name.

(continued from last page)

Parameters:

`model` - the AbstractModel that contains the constant.
`the` - String name of the constant.

findVariable

```
public static UserDefinedVariable findVariable(AbstractModel model,
        String name)
```

Finds the UserDefinedVariable in the given model with the given name.

Parameters:

`model` - the AbstractModel that contains the variable.
`the` - String name of the variable.

findFunction

```
public static UserDefinedFunction findFunction(AbstractModel model,
        int ident)
```

Finds the UserDefinedFunction in the given model with the given name.

Parameters:

`model` - the AbstractModel that contains the function.
`the` - String name of the function.

getPythonInit

```
public static String getPythonInit()
```

Gets the python initialization string.

Returns:

the String containing the results of the python initialziation.

getPythonHeader

```
public String getPythonHeader()
```

Returns the header that is prepended to the python code before execution.

Returns:

the Sttring that preceeds the user code

validate

```
public void validate()
```

complete

```
public void complete()
```

com.cafean.client.analysis.numerics
Class UserDefinedValue

```
java.lang.Object
   |-com.cafean.client.analysis.GenericObject
      |-com.cafean.client.analysis.AbstractComponent
         |-com.cafean.client.analysis.numerics.UserDefinedValue
```

Direct Known Subclasses:
UserDefinedConstant, UserDefinedVariable

---

public abstract class **UserDefinedValue**
extends AbstractComponent

A UserDefinedValue is a numeric component that may be referenced by a Real value inside an AbstractModel. UserDefinedValues are rendered by DrawnUserValues, and have to contain functions for getting the current value, and setting the units.

Fields inherited from class com.cafean.client.analysis.GenericObject
DATA_COMPLETE, DATA_ERROR, DATA_INCOMPLETE, DATA_WARNING

Constructor Summary

public	`UserDefinedValue()` Creates a new instance of UserDefinedValue
public	`UserDefinedValue(AbstractModel model,int number)` Creates a new instance of a UserDefinedValue inside a model with a specified component number.

Method Summary

DrawnComponent	`createDrawnComponent()` This creates a new DrawnUserValue to render this UserDefinedValue inside a view.
void	`fireComponentDeleted()`
abstract Real	`getCurrentValue()` This gets the current value for this user defined value.
Real	`getUnits()` Returns a clone of the current value.
void	`popupDataDialog(java.awt.Window parent,boolean modal)`

static String	replaceNumericNames(String token,AbstractModel model) Replaces named references to user defined constants in the given token with the values of those constants.
void	restoreState(Hashtable table)
void	setName(String val_)
abstract void	setUnits(Real units) Sets the value inside this numeric to use the same units as the given Real.
void	setUnits(String units) Sets the units on this numeric based on the SI units given

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, 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, writeMuxLoadArray, writeMuxLoadArray, writeSP, writeSP

Methods inherited from class java.lang.Object

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

Constructors

(continued from last page)

UserDefinedValue

`public` **`UserDefinedValue`**`()`

Creates a new instance of UserDefinedValue

UserDefinedValue

`public` **`UserDefinedValue`**`(AbstractModel model, int number)`

Creates a new instance of a UserDefinedValue inside a model with a specified component number.

Parameters:

`model` - the AbstractModel.
`number` - the component number for this function.

Methods

getCurrentValue

`public abstract Real` **`getCurrentValue`**`()`

This gets the current value for this user defined value.

Returns:

the Real with the current value.

---

createDrawnComponent

`public DrawnComponent` **`createDrawnComponent`**`()`

This creates a new DrawnUserValue to render this UserDefinedValue inside a view.

Returns:

the DrawnUserValue for this numeric.

---

setUnits

`public abstract void` **`setUnits`**`(Real units)`

Sets the value inside this numeric to use the same units as the given Real.

Parameters:

`the` - Real containing the new units.

---

setUnits

`public void` **`setUnits`**`(String units)`

Sets the units on this numeric based on the SI units given

Parameters:

`units` - the SI units to be set.

---

getUnits

`public Real` **`getUnits`**`()`

Returns a clone of the current value.

popupDataDialog

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

fireComponentDeleted

```
public void fireComponentDeleted()
```

restoreState

```
public void restoreState(Hashtable table)
```

setName

```
public void setName(String val_)
```

replaceNumericNames

```
public static String replaceNumericNames(String token,
         AbstractModel model)
```

Replaces named references to user defined constants in the given token with the values of those constants.

Parameters:

`token` - a String containing a single line to have numerics replaced in
`model` - the AbstractModel containing the numerics to check

com.cafean.client.analysis.numerics
Class UserDefinedVariable

```
java.lang.Object
   |-com.cafean.client.analysis.GenericObject
      |-com.cafean.client.analysis.AbstractComponent
         |-com.cafean.client.analysis.numerics.UserDefinedValue
            |-com.cafean.client.analysis.numerics.UserDefinedVariable
```

All Implemented Interfaces:
PropertyController, IdentHolder, StateEditable, Cloneable, Checkable, ComponentElement, Cloneable

---

public class **UserDefinedVariable**
extends UserDefinedValue
implements Cloneable, ComponentElement, Checkable, Cloneable, StateEditable, IdentHolder, PropertyController

The UserDefinedVariable is a UserDefinedValue that can have it's current value set as the output of a {@link UserDefinedFunction}. As a UserDefinedValue it can be used as input for `functions`, or as the value of a {@link Real}

Fields inherited from class `com.cafean.client.analysis.GenericObject`
`DATA_COMPLETE`, `DATA_ERROR`, `DATA_INCOMPLETE`, `DATA_WARNING`

Constructor Summary

`public`	`UserDefinedVariable()` Creates a new instance of UserDefinedVariable.
`public`	`UserDefinedVariable(AbstractModel model,int componentNumber)` Creates a new instance of UserDefinedVariable inside a model with a specified component number.

Method Summary

`Object`	`clone()`
`void`	`complete()`
`void`	`copyFrom(GenericObject object)`
`int`	`getAttributeIndex(String property)` Returns a relative index that can be used to order property lists.
`Category`	`getCategory()` The category for a UserDefinedVariable is always `CAT_VARIABLE`

Real	getCurrentValue()
Real	getUnits()
boolean	isPropertyActive(String propertyName) Returns false if this object has a property with the given name that is considered inactive; otherwise true.
boolean	isPropertyEnabled(String propertyName) Returns false if this object has a property with the given name that has dependency code that fails;true otherwise
boolean	isPropertyRequired(String propertyName) Returns false if this object has a property with the given name that has requirement code that fails; true otherwise.
boolean	isPropertyResizable(String propertyName) Returns false if this object has an array property with the given name that should not normally be resizable.
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.
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.
String	label()
void	restoreState(Hashtable state)
void	setCurrentValue(Real value) Sets the current value inside this variable.
void	setCurrentValueDouble(double siValue) Sets the current value inside this variable by setting its SI value.
void	setUnits(Real units)
void	storeState(Hashtable state)
String	toString()

Methods inherited from class com.cafean.client.analysis.numerics.UserDefinedValue

createDrawnComponent, fireComponentDeleted, getCurrentValue, getUnits, popupDataDialog, replaceNumericNames, restoreState, setName, setUnits, setUnits

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, 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

Constructors

UserDefinedVariable

```
public UserDefinedVariable()
```

Creates a new instance of UserDefinedVariable.

UserDefinedVariable

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

Creates a new instance of UserDefinedVariable inside a model with a specified component number.

Parameters:

`model` - the AbstractModel.
`number` - the component number for this function.

Methods

(continued from last page)

copyFrom

`public void copyFrom(GenericObject object)`

clone

`public Object clone()`

getCurrentValue

`public Real getCurrentValue()`

setCurrentValue

`public void setCurrentValue(Real value)`

Sets the current value inside this variable.

Parameters:

`value` - The Real containing the new value for this variable.

setCurrentValueDouble

`public void setCurrentValueDouble(double siValue)`

Sets the current value inside this variable by setting its SI value.

Parameters:

`siValue` - the double SI value.

getCategory

`public Category getCategory()`

The category for a UserDefinedVariable is always `CAT_VARIABLE`

Returns:

the Category for this UserDefinedVariable.

toString

`public String toString()`

label

`public String label()`

setUnits

`public void setUnits(Real units)`

getUnits

`public Real getUnits()`

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.

Parameters:

`propertyName` - a String containing the name of the property to check

Returns:

false if the property is inactive, true otherwise

getAttributeIndex

`public int getAttributeIndex(String property)`

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

storeState

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

restoreState

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

complete

`public void` **`complete`**`()`

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)` Creates a new array changed event for the given component with the given old and new array references.

Method Summary

`Object[]`	`getNewArray()` Retrieves the new array reference.
`Object[]`	`getOldArray()` 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)` Creates a new instance of ComponentChangedEvent

Method Summary

`AbstractComponent`	`getComponent()` 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

```
java.lang.Object
   |
   +-java.util.EventObject
      |
      +-com.cafean.client.event.ComponentChangedEvent
         |
         +-com.cafean.client.event.ArrayChangedEvent
            |
            +-com.cafean.client.event.ElementsAddedEvent
```

public class **ElementsAddedEvent**
extends ArrayChangedEvent

An event object describing a change to an array contained in a component.

Constructor Summary

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.

Method Summary

int[]	`getIndexes()` 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)

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	`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.

Method Summary

`int[]`	`getIndexes()` 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 UserDefinedConstant	`loadUserConstant(UserConstantRec rec,AbstractModel model)` Loads the given UserConstantRec into a UserDefinedConstant object created in the given model.
static UserDefinedFunction	`loadUserFunction(UserFunctionRec rec,AbstractModel model)` Loads the given UserFunctionRec into a UserDefinedFunction object created in the given model.
static UserDefinedVariable	`loadUserVariable(UserVariableRec rec,AbstractModel model)` Loads the given UserVariableRec into a UserDefinedVariable object created in the given model.
static void	`loadVisualComponents(Vector drawingBlocks,Vector viewBlocks,AbstractModel model)` Loads the ViewComponent, DrawnComponent and DrawnAnnotation records from the given Vectors.
static com.apt.xdr.PibBlock	`readDrawingBlock(com.apt.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.
static UserConstantRec	`storeUserConstant(UserDefinedConstant var)` Stores the given UserDefinedConstant into a UserConstantRec object for use in saving it into a PIB formatted file.
static UserFunctionRec	`storeUserFunction(UserDefinedFunction var)` Stores the given UserDefinedFunction into a UserFunctionRec object for use in saving it into a PIB formatted file.
static UserVariableRec	`storeUserVariable(UserDefinedVariable var)` 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.apt.xdr.PibBlock readDrawingBlock(com.apt.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 Writeable

public interface **Writeable**

Interface implemented by components that know how to write themselves.

Method Summary

void	`write(java.io.PrintWriter out)` 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)` 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)`

Writeable	getWriteable() 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() Updates this AsciiViewer's data
void	setTargetComponent(AbstractComponent comp) 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, 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()` Disables and deactivates this Updater.
`void`	`run()` Continuously updates and sleeps while enabled to update the AsciiViewer by calling `writeComponent`.
`void`	`update()` Requests that an update of the AsciiViewer be performed on a background thread.
`void`	`Updater()` 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)` Constructs a new BeanBox.

Method Summary

`java.awt.Component`	`add(java.awt.Component comp)` Adds the specified component to this container.
void	`addBoxSelectionListener(BoxSelectionListener listener)` Adds the given listener to the list that is notified when the BeanBox selection set may have changed.

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.
void	addNotify() Notifies this component that it now has a parent.
void	addSelectedComponent(Object selected) Add a single object to the selection.
void	addSelectedComponents(Object[] sel) Add an array of beans to the current selection.
java.awt.Component	addToFront(java.awt.Component comp) Adds the specified component to this container.
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.
void	calcSelectionBounds() Calculate the bounds of selected beans.
boolean	canAlign() The align operation can only be performed if two or more components are selected.
boolean	canCopy() The Copy or cut command can only be performed if there is at least one object selected.
boolean	canDelete() The delete operation can only be performed if a copy opperation could be performed.
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.
void	clearSelection() Clear the current selection.
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.
boolean	copy() Serialize the current selection to the system clipboard.
JPopupMenu	createPopupMenu(java.awt.event.MouseEvent evt) Creates a popup menu suitable for this GlassPanel and the model it is viewing.
JMenu	createZoomMenu() Creates a Zoom menu suitable for this view.

void	`cut()` Serialize the current selection to the system clipboard, and then remove it from the BeanBox.
void	`delete()` Remove the current selection from the model.
java.awt.Component[]	`findComponentsInside(int x1,int y1,int x2,int y2)` Retreive the list of components located within a rectangular region bounded by the points (x1,y1) and (x2,y2).
Vector	`getAnnotations()` This gets fills a vector with all of the `annotations` it contains.
Vector	`getComponents(boolean includeConnections)` Retrieves a Vector of all the Components in this BeanBox.
DrawnComponent	`getDrawnComponent(AbstractComponent comp)` Retrieves the DrawnComponent rendering the given component in this BeanBox.
DrawnComponent	`getDrawnComponent(int compIdent)` Retrieves the DrawnComponent rendering the given component in this BeanBox.
Vector	`getDrawnComponents()` Retrieves a Vector of all the Components in this BeanBox, including `Annotations`, `DrawnComponents` and `DrawnConnections`.
java.awt.Dimension	`getMaximumSize()` Return the maximum size of this component.
java.awt.Dimension	`getMinimumSize()` Return the minimum size of this component.
AbstractModel	`getModel()` Getter for the `AbstractModel` that contains the `com.cafean.client.analysis.ViewComponent` this BeanBox represents.
int	`getNumSelected()` Return the number of selected components.
java.awt.Dimension	`getPanelSize()` Return the panelSize parameter from the ZoomablePanel parent class.
java.awt.Dimension	`getPreferredSize()` Return the preferred size of this component.
Double	`getScale()` Return the scale parameter from the ZoomablePanel parent class.
java.awt.Component	`getSelected(int i)` Return a selected bean.
java.awt.Component[]	`getSelection()` Return the array of selected beans.

java.awt.Component[]	getSelection(Vector selComps) Retreive the list of selected components in the drawing order.
java.awt.Rectangle	getSelectionBounds() Return the bounds of selected beans.
ViewComponent	getViewComponent() Retrieves the ViewComponent that corresponds with the DrawnView that contains this BeanBox.
double	getWidthScaleFactor() Gets this view's width scale factor.
boolean	isSelected(java.awt.Component component) Returns true if the given component is part of the current selection.
static java.awt.Component[]	loadComponents(AbstractModel model,com.apt.xdr.PibBlock[] blocks) Loads annotations and drawn components from the given array of PibBlock's.
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.
void	minimizeView() Minimizes the current view to the minimum possible with the current components.
void	organizeSelection() Organize the selected components.
void	organizeView(Vector drawnComponents,boolean relative) Calls AbstractModel.layoutComponents with the given components.
void	paint(java.awt.Graphics g)
void	paintImmediately(int x,int y,int w,int h)
void	paintImmediately(JComponent component) Immediately paints the region of the (scaled) BeanBox that the given component is located.
void	paste()
void	print(java.awt.Graphics g)
void	redrawSelection() Redraw the selected components, or all components in this view if none are selected
void	refresh(java.awt.Rectangle r) This refreshses a given Rectangle that indicates a dirty region.
void	remove(java.awt.Component comp) Removes the specified component from this container.

void	`remove(int index)` Removes the component, specified by `index`, from this container.
void	`removeBoxSelectionListener(BoxSelectionListener listener)` Removes the given listener from the list that is notified when the BeanBox selection set may have changed.
void	`removeSelectedComponent(Object sel)` This removes the Object from the list of selected components.
void	`renumberSelectedComponents()` This takes the selected `components` and has the model `renumber` their component numbers.
void	`repaint(long tm,int x,int y,int width,int height)`
void	`resetConnections()` This resets all the connections that connect to the selected components
void	`restoreState(Hashtable state)` Restore the state of the bean from an earlier edit.
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.
void	`setPaintEnabled(boolean b)` Enables or disables the repainting of this BeanBox.
void	`setScale(Double scale)` Set the scale parameter.
void	`setSelectedComponent(java.awt.Component focus)` Clear the current selection and add a single object to the selection.
void	`showAllConnections()` This goes through all of the components currently in the view, and tries to render their connections.
void	`showAllConnections(boolean undoable)` This goes through all of the components currently in the view, and tries to render their connections.
void	`storeState(Hashtable state)` Store the state of the bean to permit undo.
void	`updateComponentList(Vector components,boolean organize)` Synchronizes the component list in this BeanBox to the given list of components.
void	`updateSelection()` Updates this BeaBox'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, 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

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 prefered 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 refreshses 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 opperation 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)
```

Retreive 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)`

Retreive 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 minmum 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	`boxSelectionChanged(BeanBox box)` 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	`ClientPluginLoader(Vector plugins)` Creates a new instance of ClientPluginLoader to load client plugins into the given Vectors.

Method Summary

void	`pluginException(Exception ex)` Called when there is an exception thrown reading in a plugin.
void	`pluginLoaded(Object plugin)` 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 ComponentSelectionDialog

```
java.lang.Object
   |-java.awt.Component
      |-java.awt.Container
         |-java.awt.Window
            |-java.awt.Dialog
               |-javax.swing.JDialog
                  |-com.cafean.client.ui.ComponentSelectionDialog
```

All Implemented Interfaces:
ListSelectionListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, RootPaneContainer, javax.accessibility.Accessible, WindowConstants

---

public class **ComponentSelectionDialog**
extends JDialog
implements WindowConstants, javax.accessibility.Accessible, RootPaneContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, ListSelectionListener

This dialog allows for the selection of a component that is stored by ident. A Category may be specified before the dialog is set visible which will limit the selection scope to the components in that Category. Optionally, the dialog may be set to enable the "Clear" button which clears the current selection. Additionally, the ident of a currently selected component can be set.

Fields inherited from class `java.awt.Component`
`BOTTOM_ALIGNMENT`, `CENTER_ALIGNMENT`, `LEFT_ALIGNMENT`, `RIGHT_ALIGNMENT`, `TOP_ALIGNMENT`

Constructor Summary

`public`	`ComponentSelectionDialog(java.awt.Frame parent)` Creates new form ComponentSelectionDialog.
`public`	`ComponentSelectionDialog(java.awt.Dialog parent)` Creates new form ComponentSelectionDialog.

Method Summary

`AbstractComponent`	`getSelection()` Returns the component selected by the dialog, or NULL if it was cleared
`boolean`	`isCancelled()` Returns true unless the dialog was exited using the OK button

void	`setAllowEmpty(boolean allowEmpty)` Sets whether or not this dialog can allow no selection
void	`setAvailableComponents(AbstractComponent[] components)` Initializes the array of available components.
void	`setCategory(Category category)` Sets the Category that will be used to fill this dialog.
void	`setCreator(ComponentCreator creator)` Sets the custom component creator that this dialog will execute if the user presses the create button.
void	`setModel(AbstractModel model)` Sets the AbstractModel for this component
void	`setSelection(AbstractComponent comp)` Sets the initially selected component
void	`setSelection(int selection)` Sets the initial selection ident
void	`setShowCategory(boolean show)` If set to true, the category number column will be shown.
void	`setShowComponentNumber(boolean show)` If set to true, the component number column will be shown.
void	`setTitle(String title)` Sets the default title.
void	`setVisible(boolean visible)` {@inheritDoc} This is where the dialog fills the list of components by Category, or by reading in all the components in the model.
void	`valueChanged(ListSelectionEvent e)` {@inheritDoc}

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

ComponentSelectionDialog

`public` **`ComponentSelectionDialog`**`(java.awt.Frame parent)`

Creates new form ComponentSelectionDialog. This initializes the table and sets the window location.

ComponentSelectionDialog

`public` **`ComponentSelectionDialog`**`(java.awt.Dialog parent)`

Creates new form ComponentSelectionDialog. This initializes the table and sets the window location.

Methods

(continued from last page)

setCategory

`public void setCategory(Category category)`

Sets the Category that will be used to fill this dialog.

setAllowEmpty

`public void setAllowEmpty(boolean allowEmpty)`

Sets whether or not this dialog can allow no selection

setSelection

`public void setSelection(int selection)`

Sets the initial selection ident

setSelection

`public void setSelection(AbstractComponent comp)`

Sets the initially selected component

setModel

`public void setModel(AbstractModel model)`

Sets the AbstractModel for this component

setCreator

`public void setCreator(ComponentCreator creator)`

Sets the custom component creator that this dialog will execute if the user presses the create button. Calling this with a value other then null before the dialog is set visible makes the create button appear.

setShowComponentNumber

`public void setShowComponentNumber(boolean show)`

If set to true, the component number column will be shown.

setShowCategory

`public void setShowCategory(boolean show)`

If set to true, the category number column will be shown.

setTitle

`public void setTitle(String title)`

Sets the default title.

setVisible

`public void setVisible(boolean visible)`

This is where the dialog fills the list of components by Category, or by reading in all the components in the model.

(continued from last page)

setAvailableComponents

`public void **setAvailableComponents**(AbstractComponent[] components)`

Initializes the array of available components. This allows for a selection from a group of components defined by the editor.

isCancelled

`public boolean **isCancelled**()`

Returns true unless the dialog was exited using the OK button

getSelection

`public AbstractComponent **getSelection**()`

Returns the component selected by the dialog, or NULL if it was cleared

valueChanged

`public void **valueChanged**(ListSelectionEvent e)`

{@inheritDoc}

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)` Creates new form CompomentSelector
public	`ComponentSelector(JDialog parent,boolean modal,AbstractComponent[] comps)` Creates new form CompomentSelector
public	`ComponentSelector(JDialog parent,boolean modal,Vector comps)` Creates new form CompomentSelector
public	`ComponentSelector(java.awt.Frame parent,boolean modal,Vector comps)` Creates new form CompomentSelector

Method Summary

AbstractComponent	`getSelection()` Return the selected component or null if there no selection has been made.
void	`hideComponentNumber()` Hides the component number field of this ComponentSelector.

void	hideComponentType() Hides the component type field of this ComponentSelector.
boolean	isCanceled() 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) 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 CompomentSelector

Parameters:

`parent` - the parent `java.awt.Frame` of this dialog.
`modal` - true if this dialog shoud 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 CompomentSelector

Parameters:

`parent` - the parent `javax.swing.JDialog` of this dialog.
`modal` - true if this dialog shoud be modal.
`comps` - the AbstractComponent[] list to choose from.

ComponentSelector

```
public ComponentSelector(JDialog parent,
                         boolean modal,
                         Vector comps)
```

Creates new form CompomentSelector

Parameters:

`parent` - the parent `java.awt.Frame` of this dialog.
`modal` - true if this dialog shoud be modal.
`comps` - the Vector of `components` to choose from.

ComponentSelector

```
public ComponentSelector(java.awt.Frame parent,
                         boolean modal,
                         Vector comps)
```

Creates new form CompomentSelector

Parameters:

`parent` - the parent `javax.swing.JDialog` of this dialog.
`modal` - true if this dialog shoud 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 betweeen two components using a {@link com.cafean.client.ui.DrawnConnection}

Field Summary

static final int	`CROSSFLOW_DROP_ZONE` A connection target for a crossflow connect Value: **2**
static final int	`INLET_DROP_ZONE` Connection target that points into the component. Value: **0**
static final int	`INLET_MOVEABLE_CONNECTOR` A connection source that points into the component. Value: **3**
static final int	`INVISIBLE_CONNECTOR` A connection point that does not draw on the drawn component Value: **6**
static final int	`OUTLET_DROP_ZONE` A Connection target that points out from the component Value: **1**
static final int	`OUTLET_MOVEABLE_CONNECTOR` A connection source that points out of the component Value: **4**
static final int	`STATIC_CONNECTOR` A connection source that cannot be moved Value: **5**

Constructor Summary

public	`ConnectingPt(int type,int linkNum)` The default constructor for a ConnectingPt.
public	`ConnectingPt(int type,int linkNum,ConnectionData data)` The default constructor for a ConnectingPt.

Method Summary

Object	`clone()`
java.awt.Color[]	`getBadgeColors()` Gets the colors of the Badges for this ConnectingPt
java.awt.Shape[]	`getBadges()` Gets the Badges for this ConnectingPt
ConnectionData	`getConnectionData()` Gets the ConnectionData that defines this ConnectingPt's position.
java.awt.Shape	`getHandle()`
double	`getHandleSize()` Gets the size of the handle.
boolean	`getNeedsTarget()` Returns true if connections started from this ConnectingPt will allow connections to whole components.
Pad	`getPad()` This returns the actual drawn location on the panel where a connection can be initiated or completed with the `connect` tool.
static java.awt.geom.Point2D.Double	`getRotatedLocation(double theta,java.awt.geom.Point2D.Double pt,java.awt.geom.Point2D.Double ctr)` 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)` computes the new location of a point after it has been rotated.
int	`getType()` Returns the type of this ConnectingPt.
boolean	`isConnected()` Returns true if there is a `Connection` to the `AbstractComponent` that woulc be drawn to this ConnectingP existst, even if there is no `DrawnConnection` for that `Connection` on this `DrawnView`
boolean	`isVisible()` Returns true if this ConnectingPt is a connectinPt that can be visible.
void	`move(double diffx,double diffy)` Moves a ConnectingPt by the distance in each dimension.

void	`rotate(double theta,double x,double y)` Rotates a ConnectingPt clockwize by the angle theta, around the point specified by x and y.
void	`setBadgeColors(java.awt.Color[] badgeColors)` Sets the colors of the Badges for this ConnectingPt
void	`setBadges(java.awt.Shape[] theBadges)` Sets the Badges for this ConnectingPt
void	`setConnected()` Should be called if there is a `Connection` to the `AbstractComponent` that woulc be drawn to this ConnectingPt exists, even if there is no `DrawnConnection` for that `Connection` on this `DrawnView`.
void	`setConnectPtType(int type)` Sets the type of this ConnectingPt.
void	`setDisconnected()` Should be called if there is a `Connection` to the `AbstractComponent` that woulc be drawn to this ConnectingPt has been disconnected.This specifically gets called by the DrawnComponent when a `Connection` has been disconnected.
void	`setHandle(java.awt.Shape theHandle)`
void	`setHandleSize(double hsHeight)` Sets the size of the handle.
void	`setNeedsTarget(boolean val_)` Sets whether connections started from this ConnectingPt will allow connections to whole components.
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.
String	`toString()` Constructs an informative label for this ConnectingPt, including type, and pad coordinates
boolean	`typeIsConnector()` This determines if this ConnectingPt is of a type that can initiate a connection.
boolean	`typeIsCrossflow()` This determines if this ConnectingPt is the used to represent the outlet of a component.
boolean	`typeIsDropZone()` This determines if this ConnectingPt is of a type that can complete a connection.
boolean	`typeIsInlet()` This determines if this ConnectingPt is the used to represent the inlet of a component.
boolean	`typeIsOutlet()` 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 woulc 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 woulc 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 woulc 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 clockwize 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()` Creates new form ConnectionsPanel
public	`ConnectionSetPanel(AbstractComponent component,Connection[] connections)` Creates new form ConnectionsPanel

Method Summary

void	`addConnectionSelectionListener(ConnectionSelectionListener listener)` Adds a selection listener to the list of listeners in this panel
void	`clearSelection()` Clears the currently selected connections

void	`init(AbstractComponent component,Connection[] connections)` Initializes this panel for use in displaying the given connections
void	`refresh(Connection[] connections)` Refreshes this panel with the given Connection list.
void	`removeConnectionSelectionListener(ConnectionSelectionListener listener)` 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
Class CreateViewsDialog

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-java.awt.Window
            ┗-java.awt.Dialog
               ┗-javax.swing.JDialog
                  ┗-com.cafean.client.ui.CreateViewsDialog
```

public class **CreateViewsDialog**
extends JDialog

A dialog intended for creating views for a newly imported model. This dialog assumes that the model contains no existing views.

Fields inherited from class `java.awt.Component`
`BOTTOM_ALIGNMENT`, `CENTER_ALIGNMENT`, `LEFT_ALIGNMENT`, `RIGHT_ALIGNMENT`, `TOP_ALIGNMENT`

Constructor Summary

`public`	`CreateViewsDialog(java.awt.Frame parent,AbstractModel model)` Creates new CreateViewsDialog to allow the user to select which component categories to create views for.

Method Summary

`void`	`setVisible(boolean b)`

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

CreateViewsDialog

```
public CreateViewsDialog(java.awt.Frame parent,
                         AbstractModel model)
```

Creates new CreateViewsDialog to allow the user to select which component categories to create views for.

Methods

setVisible

```
public void setVisible(boolean b)
```

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	performExport(boolean exportCheck) 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` Oriented towards the bottom of the screen Value: **2**
static final int	`CENTER` Align in the direct center. Value: **5**
static final int	`CENTER_H` align in the center horizontally. Value: **7**
static final int	`CENTER_V` Align in the center vertically. Value: **6**

static final int	CIRCLE Indicates the Pad for the ConnectingPt should be a Circle Value: **0**
static final int	CROSSHATCH Indicates the Pad for the ConnectingPt should be a Crosshatch Value: **4**
static final int	DIAMOND Indicates the Pad for the ConnectingPt should be a Diamond Value: **3**
static final int	DOWN Oriented towards the bottom of the screen Value: **2**
static final int	LEFT Oriented to the left hand of the screen Value: **0**
static final int	max_positions The number of different orientation enumerations. Value: **4**
static final int	NONE No Pad type is needed for the ConnectingPt Value: **-1**
static final double	PIXELS_P_METER This determines how many pixels are needed to display a single meter in length Value: **25.0**
static final int	RIGHT Oriented to the right-hand of the screen Value: **1**
static final int	SEGMENT_BOTH Indicates a segment has both an inlet and outlet ConnectingPt. Value: **3**
static final int	SEGMENT_INLET Indicates a segment has only an inlet ConnectingPt. Value: **1**
static final int	SEGMENT_NONE Indicates a segment has neither an inlet nor an outlet. Value: **0**
static final int	SEGMENT_OUTLET Indicates a segment has only an outlet ConnectingPt Value: **2**

static final int	SEGMENT_SPECIAL Indicates a segment a special meaning. Value: **4**
static final int	SQUARE Indicates the Pad for the ConnectingPt should be a Crosshatch Value: **5**
static final int	TOP Oriented towards the top of the screen Value: **3**
static final int	TRIANGLE Indicates the Pad for the ConnectingPt should be a Triangle Value: **2**
static final int	UP Oriented towards the top of the screen Value: **3**

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) 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()
boolean	canBeResized() All components can be resized as a default.
void	Clear() Clears the current DrawnComponent's connections and labels
void	clearLinks() **Deprecated.** *DrawnLinks have been converted to DrawnConnections.*
Object	clone(AbstractComponent cb) Creates a clone of a DrawnComponent, including its links and connections

void	`componentChanged(ComponentChangedEvent evt)` When a DrawnComponent's component changes, the drawn component needs to call initDrawing on itself {@inheritDoc}
void	`componentConnected(Connection con)` This should make sure that the ZoomablePanel tries to add the drawnLink to the view when a connection is established.
void	`componentDeleted()` When a DrawnComponent's component is deleted, the Drawn component needs to remove itself from the view.
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.
void	`connectLinks()` **Deprecated.** *DrawnLinks are now DrawnConnection objects*
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.
boolean	`contains(int x,int y)` {@inheritDoc}
boolean	`contains(java.awt.Point p)` {@inheritDoc}
java.awt.Shape	`createBorderRegion(java.awt.Shape shape,int face,double shift)` Creates a border shape on the specified face of the given shape
java.awt.Shape	`createCenterShape(java.awt.Shape shape,int face,double shift)` Creates a stripe down the center of a given shape
static void	`createConnectionPrototypes()` Creates the various archtypes of Connection Shapes to be used at connection points
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
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.
JPopupMenu	`createPopupMenu()` Creates a custom popup context menu for this DrawnComponent that includes an optional Actions menu from sub components.
TemplateEntry	`createTemplateEntry()` Creates a new TemplateEntry for this DrawnComponent that stores all of the location and state data for this DranwComponent.

void	`disconnectAllMyLinks()` **Deprecated.** *DrawnLinks have been converted to DrawnConnections*
void	`draw(java.awt.Graphics2D g,boolean selected)` Tints the current DrawnComponent according to its state.
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.
void	`flip()` Flips the orientation of a DrawnComponent to the next one in the given rotation.
BeanBox	`getBeanBox()` Retrieves the BeanBox that contains this DrawnComponent or null if this DrawnComponent exists outside of a BeanBox.
int	`getClockwiseFace(int face)` Retrieves the face value that is 90 degrees clockwise from the given face.
AbstractComponent	`getComponent()` Each DrawnComponent is a rendering object for a specific `AbstractComponent`.
int	`getComponentID()` Retrieves the ident of this drawn component's component
java.awt.Point	`getConnectingLocation(ConnectingPt point)` Returns the absolute screen location of the given ConnectingPt.
ConnectingPt	`getConnectingPt(ConnectionData data)` Retrieves the ConnectingPt for the given ConnectionData or null of none is found.
ConnectingPt	`getConnectingPt(int i)` Returns a ConnectingPt for a given int.
ConnectingPt	`getConnectingPtAt(double x,double y)` Retrieves the connector at the given coordinates
int	`getConnectSize()` Retrieves the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent.
int	`getCounterFace(int face)` Retrieves the face value that is 90 degrees counter-clockwise from the given face.
int	`getCrossflowIndex(ConnectingPt cp)` Gets the number of ConnectingPt components that are crossflow Points that occur before the given ConnectingPt.
Action[]	`getCustomPopupActions()` Retrieves the curstom popup actions from this drawn component's `AbstractComponent`.
Vector	`getCustomPopupItems()` Returns the custom popup items of this DrawnComponent's target component

float	`getDefaultDrawLength()` The Default Draw Length for a DrawnComponent is 50 pixels.
float	`getDefaultDrawWidth()` The Default Draw Width for a DrawnComponent is 50 pixels.
double	`getDrawAngle()` Gets the angle, in radians describing the angle of this DrawnComponent
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.
java.awt.Shape	`getDrawingObject()` Gets the Shape that should be drawn when this DrawnComponent is painted.
int	`getFaceByAngle(double ang,boolean invert)` Determines the appropriate orientation for a given angle.
java.awt.Color	`getFillColor()` Returns the color to be used for filling this DrawnComponent's normal shape.
GlassPanel	`getGlassPane()` Retrieves the GlassPanel from the ZoomablePanel that contains this DrawnComponent.
int	`getHandleSize()` This gets the user defined size of the handles from the user preferences.
static java.awt.Color	`getIndicatorColor()` Gets the user preference for the Connection Color.
double	`getLength()` The length of a drawn component is the same as it's height.
double	`getLenScaleFactor()` This gets the Length Scale Factor from the AbstractModel.
float	`getMaxHeight()` This gets the maximum height for this DrawnComponent.
float	`getMaxWidth()` This gets the maximum width for this DrawnComponent.
float	`getMinWidth()` This gets the minimum width for this DrawnComponent.
java.awt.Shape	`getMirrorImageShape(java.awt.Shape shape,int mirrorCode)` Creates a mirror image of a given shape s
java.awt.Shape	`getNormalObj()` Gets the Shape that is used to drawn the component in the DrawnPanel

int	`getNumberConnections()` Retrieves the number of ConnectingPts the user specifies to be within the Connections vector of the current DrawnComponent.
int	`getOppositeFace(int face)` Retrieves the face value that is the exact opposite of the given face.
int	`getOrientation()` Returns the orientation of the DrawnComponent
JMenu	`getOrientationMenu()` Creates a menu appropriate for selecting the desired orientation for this drawn component.
static String	`getOrientationName(int orientation)` Retrieves the name of the given orientation.
java.awt.Container	`getParent()` This either gets the actual parent of the DrawnComponent, if it exists inside a Container.
java.awt.Dimension	`getPreferredSize()` Provides the preferred size for this component.
ConnectingPt	`getSelectedConnector(double x,double y)` Retrieves the connector at the given coordinates
ConnectingPt	`getSelectedDropZone(double x,double y)` Retrieves the drop zone at the given coordinates
DrawnSubComponent	`getSubComponentAt(int x,int y)` Returns the sub-component at the given coordinates, or null of no sub-component exists at that location.
String	`getToolTipText()`
String	`getToolTipText(int x,int y)`
double	`getWidthScaleFactor()` This gets the width scale factor from the AbstractModel.
double	`getX_Pos()` Gets the X coordinate of the center point of this DrawnComponent.
double	`getXDistBetweenCPs()` This distance is used in the layout algorithm to allow for drawn objects which are not square, such as segmented pipes.
double	`getXDistBetweenXflowCPs()` This distance is used in the layout algorithm to allow for drawn objects which are not square, such as segmented pipes.
double	`getY_Pos()` Gets the Y coordinate of the center point of this DrawnComponent.

ZoomablePanel	`getZoomablePanel()` Retrieves the ZoomablePanel that contains this DrawnComponent or null if this DrawnComponent exists outside of a ZoomablePanel.
boolean	`hasSubComponents()` Returns true if this DrawnComponent has DrawnSubComponents contained within it
void	`InitDrawing()` Initializes and scales this component to prepare it for painting.
boolean	`isAutoScale()` Getter for property autoScale that determines whether scaleIt calls scale this Drawn Component.
boolean	`isDrawBadges()` Getter for property drawBadges.
boolean	`isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)` Determines if an object is inside the given rectangle
boolean	`isPlenumShaped()` Plenums are traditionally shaped differently from other components.
boolean	`isPosnSet()` Determines if the DrawnComponent's position has been initialized.
boolean	`isScalable()` All components can be scaled as a default.
static boolean	`isSegmentSet(int test,int segment)` Returns true if the given segment bit set is included in the given test value.
boolean	`isSelected()` Getter for property selected.
boolean	`isValveShaped()` Valves are traditionally shaped differently from other components.
void	`loadDrawnComponent(com.apt.xdr.PibBlock block)` This function loads a DrawnComponent from a DrawnComponentRec.
void	`mouseClicked(java.awt.event.MouseEvent e)`
void	`mouseDragged(java.awt.event.MouseEvent evt)`
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)`

void	`mouseReleased(java.awt.event.MouseEvent e)`
void	`moveRel(double x,double y,boolean last)` Move the drawing relative to its previous position
void	`moveTo(double x,double y,boolean last)` Move the drawing so the center is the specified position
void	`paint(java.awt.Graphics g)` Paints this component.
void	`paintComponent(java.awt.Graphics g)` Tints the current DrawnComponent according to its state.
void	`print(java.awt.Graphics g)` Prints this component.
void	`readTemplateEntry(TemplateEntry entry)` Sets the data on this DrawnComponent from a TemplateEntry read in from a view template file.
void	`removeNotify()`
void	`repositionLinks()` Adjusts the position of all links coming out of an object
void	`repositionLinks(boolean last)` **Deprecated.** *DrawnLinks have been converted to DrawConnections.*
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.
void	`restoreState(Hashtable state)` 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)` 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)` Rotate the specified shape around a given point by the given angle.
boolean	`scaleIt()` Scale the drawing so that it reflects the true relative size
void	`setAutoScale(boolean autoScale)` Setter for property autoScale that determines whether scaleIt calls scale this Drawn Component.
void	`setBackupComponent(AbstractComponent backup)` When a DrawnComponent is rendering a cloneed object for displaying changes, the backup component is the source of external data for that component.

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.
void	`setComponent(AbstractComponent comp)` This sets the `AbstractComponent` that is being rendered by this DrawnComponent.
void	`setDrawAngle()` sets the drawAngle value depending on the DrawnComponent's current orientation
void	`setDrawBadges(boolean drawBadges)` Setter for property drawBadges.
void	`setDrawHeight(double l)` Sets the drawing height of the DrawnComponent.
void	`setDrawWidth(double w)` Sets the drawing width of the DrawnComponent.
void	`setEqualTo(DrawnComponent dc)` Sets the position and orientation of the current DrawnComponent to be the same as the DrawnComponent provided as an argument.
void	`setLabelString(String str,int index)` Adds or replaces a String in the labels array.
void	`setLenScaleFactor(double factor)`
void	`setOrientation(int orientation)` Sets the orientation of the current DrawnComponent to the specified orientation.
void	`setOrientationByAngle(double a)` Sets the orientation by an angle measure, in radians
void	`setParent(java.awt.Container parent)` Setter for the parent value of this DrawnComponent.
void	`setSelected(boolean selected)` Setter for property selected.
boolean	`setSizeTo(double len,double wid)` Resets the drawing size to the given dimension and then reinitializes the DrawnComponent.
void	`setWidthScaleFactor(double factor)`
void	`setX_Pos(double position)` Sets the location of this DrawnComponent's center to the given position in the X dimension.
void	`setY_Pos(double position)` Sets the location of this DrawnComponent's center to the given position in the Y dimension.

boolean	showConnections()
com.apt.xdr.PibBlock	store(int viewNum) This function returns a PibBlock for a drawn component.
void	storeState(Hashtable state) Store the state of the bean to permit undo.
String	toString() The String produced is based on the toString of the component.
java.awt.Point	translateConnectionToScreen(ConnectingPt pt) This converts the center of a passed connecting Point into the coordinates on the zoomable panel.
java.awt.Point	translatePointToScreen(java.awt.Point point) 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, 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`

Fields

NONE

`public static final int` **`NONE`**

No Pad type is needed for the ConnectingPt

CIRCLE

`public static final int` **`CIRCLE`**

Indicates the Pad for the ConnectingPt should be a Circle

(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 cloneed 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 appripropriately in relation to one another. Note: The base implementation returns null.

Returns:

a javax.swing.JComponent[] array or null of 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 InitDrawning.

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 coordiinate 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 oposite 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-clockwize 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 of 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 of 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 curstom 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 ceter 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 of 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 DranwComponent. 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 renderer for a `Connection`between two components that draws a segmented line between the `DrawnComponent`renderers 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 Value: **60.0**
static final double	HALFWIDTH Value: **7.0**

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)` Creates a renderer for the given Connection object.
public	`DrawnConnection(Connection connection,DrawnComponent source,DrawnComponent target)` Creates a renderer for the given Connection object and initializes its' source and target DrawnComponent references.

Method Summary

boolean	`addPoint(java.awt.Point p)` Adds a path point at the given point, if the point falls on an existing line segment.
boolean	`canAddPoint(java.awt.Point p)` Return true if a point can be added to the path.
boolean	`canRemovePoint(java.awt.Point p)` Return true if a point can be removed from the path.
void	`componentChanged(ComponentChangedEvent evt)`
void	`componentRemoved(DrawnComponent comp)` This should be called by the view when a drawn component has been removed.
void	`componentReshaped(DrawnComponent comp)` 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 DrawnConnection	`createDrawnConnection(Connection connection,DrawnComponent source,DrawnComponent target)`
TemplateEntry	`createTemplateEntry()` 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)` Draws indicator colored lines between each of this DrawnConnection's plotted points.

java.awt.Point[]	getPath() Retrieves a copy of the plotted path points that this DrawnConnection is painting.
DrawnComponent	getSource()
DrawnComponent	getTarget()
String	getToolTipText()
java.awt.Rectangle	getUsedBounds()
void	InitDrawing()
boolean	isGeneratePoints() Getter for property generatePoints.
boolean	isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)
void	loadDrawnComponent(com.apt.xdr.PibBlock block) This function loads a DrawnComponent from a DrawnComponentRec.
void	mouseClicked(java.awt.event.MouseEvent e)
void	mouseDragged(java.awt.event.MouseEvent e) 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) Handle mousePressed events to support segment and point manipulation
void	mouseReleased(java.awt.event.MouseEvent e) Handle mouseReleased events to support segment and point manipulation
void	readTemplateEntry(TemplateEntry entry) Sets the data on this DrawnComponent from a TemplateEntry read in from a view template file.
void	removeClosestPoint(java.awt.Point p) Attempts to remove the closest point in this DrawnConnection's set of points.
boolean	removePoint(java.awt.Point p) Attempts to remove the path point at the given location.
void	repaint()

void	restoreState(Hashtable state) Restore the state of the bean from an earlier edit.
void	setGeneratePoints(boolean generatePoints) Setter for property generatePoints.
void	setLocation(int x,int y)
void	setPath(java.awt.Point[] path) Sets the path points that this DrawnConnection will use for connecting.
void	setSelected(boolean selected)
com.apt.xdr.PibBlock	store(int viewNum) This function returns a PibBlock for a drawn component.
void	storeState(Hashtable state) 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, 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, 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

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.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.

---

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 DranwComponent. 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

```
java.lang.Object
   |
   +-java.awt.Component
      |
      +-java.awt.Container
         |
         +-javax.swing.JComponent
            |
            +-com.cafean.client.ui.DrawnComponent
               |
               +-com.cafean.client.ui.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 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	DrawnUserValue(UserDefinedValue value) Creates a new instance of DrawnUserValue

Method Summary

void	`actionPerformed(java.awt.event.ActionEvent evt)` 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()` Getter for property label.
JPanel	`getPanel()` Getter for property panel.
java.awt.Dimension	`getPreferredSize()`
JLabel	`getUnits()` Getter for property units.
JLabel	`getValueLabel()` Getter for property valueLabel.
RealTextField	`getValueText()` Getter for property valueText.
void	`InitDrawing()` Updates the values inside this component, and updates it's bounds
boolean	`isEditable()` Getter for property editable.
boolean	`isShowLabel()` Getter for property showLabel.
boolean	`isShowUnits()` Getter for property showUnits.
void	`loadDrawnComponent(com.apt.xdr.PibBlock block)`
void	`popupEditor()` Opens a dialog for editing the properties of this drawing.
void	`readByteArray(byte[] byteArr)` Reads this DrawnUserValue's data from the given byte[] of XML Encoded information.
void	`restoreState(Hashtable state)` Restore the state of the bean from an earlier edit.
void	`setBackground(java.awt.Color color)`
void	`setEditable(boolean editable)` Setter for property editable.
void	`setForeground(java.awt.Color color)`

void	setLabel(JLabel label) Setter for property label.
void	setPanel(JPanel panel) Setter for property panel.
void	setShowLabel(boolean showLabel) Setter for property showLabel.
void	setShowUnits(boolean showUnits) Setter for property showUnits.
void	setUnits(JLabel units) Setter for property units.
void	setValueLabel(JLabel valueLabel) Setter for property valueLabel.
void	setValueText(RealTextField valueText) Setter for property valueText.
com.apt.xdr.PibBlock	store(int viewNum)
void	storeState(Hashtable state) 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, 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, 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

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.

setValueLabel

`public void setValueLabel(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 corrisponds 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)` Creates a new instance of DrawnView.

Method Summary

void	`addAnnotation(Annotation comp)` This adds an {@link Annotation} to the center of the `panel's` viewport.
void	`addAnnotation(Annotation comp,boolean tofront)` This adds an {@link Annotation} to the center of the `panel's` viewport.

void	addComponents(Iterator itr) Adds a list of components by an iterator on a list of components.
void	addComponents(Iterator itr,boolean select) Adds a list of components by an iterator on a list of components.
DrawnComponent	addDrawnComponent(AbstractComponent comp,boolean select) This adds a specific AbstractComponent to a drawn view.
void	addDrawnComponent(DrawnComponent comp,boolean select) This adds a specific DrawnComponent to this DrawnView.
void	addDrawnComponent(DrawnComponent comp,boolean select,boolean withCons) This adds a specific DrawnComponent to this DrawnView.
void	addMenuItem(JMenuItem item) This allows plugins to add {@link JMenuItem menu items} to the "Tools" menu for this DrawnView.
void	addOverlapPanel(OverlapPanel panel) Adds an overlapping internal panel to this view for use as a user input sub-dialog.
void	addToolbar(JToolBar toolbar,String prefix,boolean visible) Adds the given toolbar to this DrawnView.
void	clearSelection() Clears the current selection from the zoom panel's bean box
Vector	getAnnotations() Gets all of the annotations in the panel for this Drawn View.
BeanBox	getBeanBox() An accessor for the BeanBox from the ZoomablePanel
java.awt.Dimension	getCanvasSize() Getter for the size of the ZoomablePanel
Vector	getComponents(boolean includeConnections) Returns all of the components stored inside the beanbox in an array.
DrawnComponent	getDrawnComponentAt(int i) Returns the DrawnComponent at a given index inside the bean Box.
int	getDrawnComponentCount() Returns the number of DrawnComponents that exist inside this drawn view.
DrawnComponent[]	getDrawnComponents() Returns all of the components stored inside the beanbox in an array.
JMenuBar	getMenubar()
AbstractModel	getModel() Gets the AbstractModel for this DrawnView.

DrawnComponent[]	getSelectedComponents() Returns all of the components that are selected inside the beanbox in an array.
String	getTitle() Returns the title of the dialog containing this view or the name of the ViewComponent.
Toolbox	getToolbox() Retrieves this DrawnView's Toolbox.
ViewComponent	getViewComponent() Gets the ViewComponent for this DrawnView.
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.
java.awt.Dimension	getViewSize() Returns the dimensions of the viewport in the main scroll pane.
ZoomablePanel	getZoomablePanel() An accessor for the ZoomablePanel
double	getZoomScale() Getter for the scale of the {@link ZoomablePanel}
void	layoutView() This organizes all of the components within the view
void	menuCanceled(MenuEvent e) This implements the functionality of the MenuListener interface.
void	menuDeselected(MenuEvent e) This implements the functionality of the MenuListener interface.
void	menuSelected(MenuEvent e) This implements the functionality of the MenuListener interface.
boolean	printView(boolean usePerspective) Prints this view
boolean	printView(org.apache.batik.transcoder.print.PrintTranscoder transcoder,boolean usePerspective) Prints this view into the given transcoder.
void	resetZoomMenu() Reset the zoom menu to ensure the current zoom factor is selected or none if the zoom is between selections.
void	setCanvasSize(java.awt.Dimension size) Setter for the size of the ZoomablePanel
void	setLocked(boolean locked) Locks this View (and updates the ViewComponent).

void	`setTitle(String title)` Sets the title of the dialog containing this view or does nothing if no dialog exists
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.
void	`setVisible(boolean b)`
void	`setZoomScale(double scale)` Setter for the `scale` of the {@link ZoomablePanel}
java.io.File	`snapshotView()` Creates an image snapshot of this view for use in the DrawnViewComponent.
void	`toFront()` Selects this view in the tabbed pane or toFront's the dialog that contains it.
void	`updateToolbars()` 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, 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

(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`**`(boolean usePerspective)`

Prints this view

Returns:
true if successful

printView

`public boolean` **`printView`**`(org.apache.batik.transcoder.print.PrintTranscoder transcoder, boolean usePerspective)`

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

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-javax.swing.JComponent
            ┗-com.cafean.client.ui.DrawnComponent
               ┗-com.cafean.client.ui.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

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	DrawnViewComponent(AbstractComponent c) Creates a new instance of DrawnComponentView

Method Summary

void	draw(java.awt.Graphics2D g,boolean selected) Draw a simple X'd box if there is no image.
DrawnEmbeddedConnection	findEmbedded(DrawnEmbeddedConnection[] array,int comp,ConnectionData data)

ConnectingPt	getConnectingPt(int compIdent,ConnectionData data)
Vector	getCustomPopupItems()
java.awt.Dimension	getMinimumSize()
DrawnSubComponent	getSubComponentAt(int x,int y)
boolean	hasSubComponents()
void	InitDrawing() Creates a draw view component and puts it in the drawing vectors.
boolean	isRepresenting(int compid)
void	loadDrawnComponent(com.apt.xdr.PibBlock block)
void	mouseClicked(java.awt.event.MouseEvent e)
void	mouseMoved(java.awt.event.MouseEvent e)
void	restoreState(Hashtable state) Restore the state of the bean from an earlier edit.
void	setBounds(java.awt.Rectangle r)
com.apt.xdr.PibBlock	store(int viewNum)
void	storeState(Hashtable state) 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, 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, 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

(continued from last page)

DrawnViewComponent

```
public DrawnViewComponent(AbstractComponent c)
```

Creates a new instance of DrawnComponentView

Methods

store

```
public com.apt.xdr.PibBlock store(int viewNum)
```

loadDrawnComponent

```
public void loadDrawnComponent(com.apt.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()` Retrieves the bounds actually used by this FullScreenDrawing.
void	`translate(int dx,int dy)` 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	`GlassPanel()` The constructor for a new GlassPanel.

Method Summary

void	`clearRubberBoxes()` Clears all of the current rubberband selection boxes.
void	`deleteRubberBox()` Erases the existing rubberband selection box.

void	`deleteRubberBoxes()` Erase the current rubberband selection boxes.
void	`deleteRubberLine(java.awt.Point begin,java.awt.Point end)` Erases an existing line between two points.
void	`deleteRubberLines(Vector points,java.awt.Point last)` Erase the current rubberband connection lines.
void	`drawRubberBox(java.awt.Rectangle box)` Draws a rubberband box on the GlassPane that corrisponds to the given Rectangle coordinates and size.
void	`drawRubberBoxes(java.awt.Rectangle[] boxes)` Drawns all the rubberband selection boxes.
void	`drawRubberLine(java.awt.Point begin,java.awt.Point end,java.awt.Point oldPoint)` Draws the rubberband connection line between two given points.
void	`drawRubberLines(Vector points,java.awt.Point last,java.awt.Point oldPoint)` Draw the rubberband connection lines
void	`forwardMouseEvent(java.awt.event.MouseEvent e)` Forward a mouse event to the current mouse target, setting it if necessary.
AbstractComponent	`getConnectionSource()` Gets the AbstractComponent that started the connection.
java.awt.Dimension	`getMaximumSize()` Return the maximum size of this panel, which is one pixel larger then the `BeanBox` below it.
String	`getToolTipText(java.awt.event.MouseEvent event)` Returns the string to be used as the tooltip for event.
java.awt.Component	`locateComponentAt(int xpos,int ypos)` Locate a component at a specific position in the display.
java.awt.Component	`locateComponentAt(int xpos,int ypos,boolean returnChild)` Locate a component at a specific position in the display.
void	`mouseClicked(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler
void	`mouseDragged(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler
void	`mouseEntered(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler
void	`mouseExited(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler

void	`mouseMoved(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler
void	`mousePressed(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler
void	`mouseReleased(java.awt.event.MouseEvent orig_evt)` Proxies the given MouseEvent to the current MouseHandler
void	`paintComponent(java.awt.Graphics g)` Repaint the glasspane.
void	`removeNotify()` {@inheritDoc} Overriden here to unregister from the tooltip manager
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
boolean	`requiresTarget()` 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, 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

(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 corrisponds 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)
```

Drawns 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`	`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.

Method Summary

`void`	`setInitialValues(int projectId,int modelId,int restartId)` Sets the initial project, model and restart id's.
`void`	`setRunOptions(com.cafean.CalcServer.command.RunOptions[] options)` Sets the Run Options passed to LaunchCalc when a job is sumitted.
`void`	`setVisible(boolean visible)`

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

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 sumitted.

---

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

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-java.awt.Window
            ┗-java.awt.Frame
               ┗-javax.swing.JFrame
                  ┗-com.cafean.client.ui.MainFrame
```

All Implemented Interfaces:
SDITopFrame, java.awt.datatransfer.ClipboardOwner, BatchProcessor, 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, SplashListener, BatchProcessor, java.awt.datatransfer.ClipboardOwner, SDITopFrame

The MainFrame is the central main class for the ModelEditor.

Field Summary

static boolean	debug Display all debug information to the standard output
static MainFrame	instance This is a static reference to the current instance of the ModelEditor.
static final int	MODE_CLEAR_CONSTANTS THis is clears constant references in clone Value: **3**
static final int	MODE_EXPORT_ASCII This is exporting an ASCII file mode Value: **2**
static final int	MODE_IMPORT This clips all values less then -1e30 to -1e29 Value: **4**
static final int	MODE_NORMAL This is normal operational mode Value: **0**

static final int	MODE_SAVE_MED This is saving a MED file mode Value: **1**
static final int	SELECTOR_OPEN A file selector mode for openFileSelector that indicates an Open type Value: **1**
static final int	SELECTOR_SAVE A file selector mode for openFileSelector that indicates a Save type Value: **2**
SnapCorba	snapCorba 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) This constructs a new instance of the ModelEditor.

Method Summary

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.
void	activateUndoButton(String name,boolean set) This activates the Undo button, which will allow the user to undo a change inside the ModelEditor.
static void	addAbstractModel(AbstractModel model) Adds a new AbstractModel to the ModelEditor.
static void	addAbstractModel(AbstractModel model,String label) Adds a new AbstractModel to the model editor.

static void	`addContextHelp(java.awt.Component comp,String label)` Adds a component to the context sensitive help, with the given label.
void	`addDrawnViewTab(DrawnView view)` Adds the given DrawnView to the tabbed pane holding all DrawnViews in Single Window mode.
static void	`addExportItem(JMenuItem item)` This inserts a new export option into the export menu.
static void	`addImportItem(JMenuItem item)` This inserts a new import option into the import menu.
static void	`addMenuItem(JMenuItem item,String name)` This is used to add a new menu item to the model editor main frame menu bar.
static int	`addMessage(String text)` Adds a new message to the MessageWindow.
static void	`addMessage(String text,AbstractComponent comp)` 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)` 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)` Adds the given text as a message to the message window associated with the given GenericObject.
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.
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.
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.
void	`addPlaybackPanel(com.cafean.client.anim.PlaybackButtonPanel panel)` Adds the given playback panel to the list of panels updated in `#updatePlaybackButtons`
static void	`addRegisteredDialog(java.awt.Dialog dialog,AbstractModel model)` Adds the given dialog to the MainFrame's list of registered child dialogs.
static void	`addSound(int soundType)` This plays a sound, depending on the enumuerated sound type.

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.
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.
void	closeAbstractModel(AbstractModel model) Closes the given AbsractModel.
void	closeMessageWindow() Closes the MessageWindow from an external source.
void	createModel(MECodePlugin plugin) Creates a new AbstractModel for the given plugin.
static String	dateStamp() Gets a string that contains the current date.
static void	deactivateRedoButtons() disables the undoButton and redoButton
static void	disableMainFrame() Disables all mouse events for this MainFrame.
static void	enableMainFrame() Deactivates this MainFrame's glass pane thus allowing mouse events to pass through to the underlying components.
static ClientCodePlugin	findClientPlugin(String id) Finds a client plugin by its ID.
static AbstractModel	findModelByIdent(int ident) This finds the AbstractModel with the given ident.
static MEPlugin	findPlugin(String id) Finds a plugin by its ID.
static AbstractModel	getAbstractModelAt(int index) Returns the AbstractModel at a specified index.
static int	getAbstractModelCount() Returns the number of models currently inside the calculation project.
static ClientCodePlugin	getClientPluginAt(int index) Returns the plugin at the specified index into the client plugins vector.
static int	getClientPluginCount() Returns the number of client plugins currently loaded.

static java.awt.datatransfer.Clipboard	getClipboard()
static MECodePlugin[]	getCodePlugins() This returns all of the plugins currently loaded that extend MECodePlugin.
static AbstractModel	getCurrentModel() This returns the AbstractModel that, of the list of currenly open models, is currently being worked on.
static MEFeaturePlugin[]	getFeaturePlugins() Returns an array of all the MEFeaturePlugins that are currently loaded.
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.
static FileChooser	getFileChooser() Gives access to a FileChooser for selecting a file.
static int	getMode() Getter for property mode.
static AbstractModel	getModel(String label) This finds the AbstractModel from the list of open models that has been stored under a given label.
static int	getNumberOfErrorsFound() Retrieves the number of error messages in the message window.
static int	getNumberOfWarningsFound() Retrieves the number of warning messages in the message window.
org.omg.CORBA.ORB	getOrb() Gets the current CORBA orb.
PDFViewer	getPdfViewer() Getter for property pdfViewer.
static MEPlugin	getPluginAt(int index) Returns the plugin at the specified index into the plugins vector.
static int	getPluginCount() Retursn the number of plugins currently loaded.
SnapPreferences	getPrefs() Getter for the SnapPreferences object.
static AbstractModel	getReferenceModel(AbstractModel model) Retrieves the reference model for the given model if one has been loaded.
static Iterator	getRegisteredDialogs() Returns an unmodifiable List of the registered dialogs.

static Iterator	getRegisteredDialogs(AbstractModel model) Returns an unmodifiable List of the registered dialogs for a given model
static java.io.File	getSnapHomeDirectory() Returns the installation directory for CAFEAN.
int	getSplashStatus()
String[]	getSplashTitle()
SnapUndoManager	getUndoManager() Accessor for the SnapUndoManager that controls the undo-stack for the ModelEditor.
UndoableEditSupport	getUndoSupport() Accessor for the UndoableEditSupport for the ModelEditor.
String	getUserName() Gets the current user id.
static String	getValidLabel(String label) Returns a valid key string for the user defined key passed in.
Version	getVersion() This gets the Version information for this instance of the ModelEditor.
void	init() Reads the ModelEditor preferences from the user's preferences file.
static void	loadReferenceModel(AbstractModel model,java.io.File file) Loads the reference model for the given model from the given file.
void	lostOwnership(java.awt.datatransfer.Clipboard clipboard,java.awt.datatransfer.Transferable contents) The ClipboardOwner inferface routine.
static void	main(String[] args) This is the main function for the ModelEditor.
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.
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.
static void	openMessageArea() 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() Refreshes the loaded steam tables.

static void	`removeAbstractModel(AbstractModel model)` Removes a AbstractModel from the model editor.
void	`removeDrawnViewTab(DrawnView view)` Removes the given DrawnView from the tabbed pane holding all DrawnViews in Single Window mode.
void	`removePlaybackPanel(com.cafean.client.anim.PlaybackButtonPanel panel)` Removes the given playback panel from the list of panels updated in `#updatePlaybackButtons`
static void	`removeRegisteredDialog(java.awt.Dialog dialog,AbstractModel model)` Removes the given dialog from the MainFrame's list of registered child dialogs.
boolean	`requestSaveFileName(AbstractModel model)` Opens a file selection dialog for specifying the target location of a MED file.
static void	`resetAllUnits(int units)` Resets all the renderers for units inside the model editor.
static void	`resetCursor()` Sets the cursor to the `default` cursor when the process requiring the waiting is complete.
void	`resetWindowingMode()` Updates and resets all Window Arrangement based on the current status.
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.
void	`setCurrentModel(AbstractModel m)` This sets the AbstractModel from the list of open models that currently is being worked on.
static void	`setFileChooserLocation(JFileChooser dlg)` Sets the given JFileChooser's location to be the center of the main frame's current screen.
static void	`setMode(int mode_)` Setter for property mode.
static void	`setPlaybackTime(String mess)` Sets the currently display status message
void	`setSplashStatus(int stat)`
void	`setUserName(String name)` Sets the current user id.
void	`setVisible(boolean visible)` Shows or hides this MainFrame.
static void	`setWaitCursor()` Sets the cursor to a `cursor` indicating the user should wait for a process before continuing.

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.
static void	`showCreateDialog()` Attempts to create and add a new model by prompting the user with the list of available plugins.
static void	`showOpenDialog()` This event is called when the user wants to open a localy stored SAM file.
void	`toFrontDrawnViewTab(DrawnView view)` Sets the given view as the selected tab in tabbed pane holding all DrawnViews in single window mode.
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.

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

Fields

(continued from last page)

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

Constructors

(continued from last page)

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)`

(continued from last page)

getSplashStatus

`public int` **`getSplashStatus`**`()`

---

getSplashTitle

`public String[]` **`getSplashTitle`**`()`

---

getVersion

`public` `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 AbsractModel. 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 currenly 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.

---

getModel

`public static` `AbstractModel` **`getModel`**`(String label)`

(continued from last page)

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 creats 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 localy stored SAM file.

Parameters:

`event` - the ActionEvent that fired this action.

requestSaveFileName

`public boolean requestSaveFileName(AbstractModel model)`

(continued from last page)

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 cancells 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 enumuerated 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()
```

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
`comp` - the AbstractComponent.
`severityCode` - the severity code chosen from those in MessageWindow.

See Also:

(continued from last page)

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()
```

(continued from last page)

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.

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.

(continued from last page)

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` indicating the user should wait for a process before continuing.

See Also:
`Cursor`

(continued from last page)

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.

(continued from last page)

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 inferface routine. Implemented as a noop here.

Parameters:
`clipboard` - the Clipboard.
`contents` - the Transferable contents on the clipboard.

init

`public void` **`init`**`()`

Reads the ModelEditor preferences from the user's preferences file.

getPrefs

`public` `SnapPreferences` **`getPrefs`**`()`

Getter for the SnapPreferences object.

Returns:
prefs.

setUserName

`public void` **`setUserName`**`(String name)`

Sets the current user id.

Parameters:

(continued from last page)

`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.

(continued from last page)

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 of 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 of there isn't one.

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`**`()`

Retursn 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.

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:

(continued from last page)

the File object containing the absolute path to the current working component.

getFileChooser

`public static 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()`

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.

(continued from last page)

Parameters:
`mode` - New value of property mode.

getPdfViewer

`public 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` Message type for non-error messages that the user should be alerted to. Value: **0**
static final int	`AlertSound` Sound type for an alert event Value: **0**
static final int	`ConnectSound` Sound type for a connection event Value: **5**
static final int	`DisconnectSound` Sound type for a disconnection event Value: **6**
static final int	`InfoMsg` Message type for normal messages. Value: **1**
static final int	`InfoSound` Sound type for an information event Value: **1**
static final int	`InternalErrMsg` Message type for errors that are not user created or caused. Value: **4**
static final int	`InternalErrSound` Sound type for an internal error event Value: **4**

static final int	NoSound Sound type for the default case Value: **-1**
static final int	UserErrorMsg Message type for errors that are considered user created or caused Value: **3**
static final int	UserErrorSound Sound type for a user error event Value: **3**
static final int	WarningMsg Message type for non-fatal errors or warnings Value: **2**
static final int	WarningSound Sound type for a warning event Value: **2**

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	MessageWindow() Creates new form MessageDialog.

Method Summary

int	addMessage(String message) Adds the given message as type InfoMsg.
void	addMessage(String text,AbstractComponent comp) Adds the given message as type InfoMsg with the attached component.
void	addMessage(String text,AbstractComponent comp,int severityCode) Adds the given message as type severityCodewith the attached component.
void	addMessage(String text,GenericObject object,int severityCode) Adds the given message as type severityCodewith the attached component.

int	`addMessage(String text,int severityCode)` Adds the given message with the given severity code
void	`addMessage(String text,Vector comps,int severityCode)` Adds the given message as type `severityCode`with the attached components.
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.
void	`addSound(int soundType)` Attemps 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()` 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()` 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()` Creates a formatted date stamp of the current time/date.
int	`getNumberOfErrorsFound()` Retrieves the number of error messages since the last clearWarningCount or clearCheckErrors call.
java.io.PrintStream	`getOutputStream()` Retrieves the PrintStream that this MessageWindow sends it's alternate output stream to.
int	`getWarningCount()` Retrieves the number of warning messages since the last clearWarningCount or clearCheckErrors call.
void	`selectComp(AbstractComponent comp,boolean clear)` **Deprecated.**
void	`setOutputStream(java.io.PrintStream out)` Sets the PrintStream that this MessageWindow sends it's alternate output stream to.
void	`setupContextHelp()` 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, 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

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)
```

Attemps 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)` Creates a new NamedValueSelector that is modal over the given parent.
public	`NamedValueSelector(JDialog parent,Object[] values,Object[] names)` Creates a new NamedValueSelector that is modal over the given parent.
public	`NamedValueSelector(java.awt.Frame parent,Vector values,Vector names)` Creates a new NamedValueSelector that is modal over the given parent.
public	`NamedValueSelector(JDialog parent,Vector values,Vector names)` Creates a new NamedValueSelector that is modal over the given parent.

Method Summary

int	`getSelectedIndex()` Returns the index of the object selected by the user.
TableCellEditor	`getTableCellEditor()` Provides a table cell editor of a property editor panel and the popup named value selection dialog

boolean	isCancelled() 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, 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

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 then 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 Pad

```
java.lang.Object
   |
   +-com.cafean.client.ui.Pad
```

All Implemented Interfaces:
Cloneable

---

public class **Pad**
extends Object
implements Cloneable

A small storage class for the location and direction of a connecting point in a drawn component.

Constructor Summary

public	`Pad(int direction,java.awt.geom.Point2D.Double pt)` Creates a new Pad with the given direction and location.
public	`Pad(int direction,double xd,double yd)` Creates a new Pad with the given direction and location.

Method Summary

int	`getFace()` Sets this pad's face.
java.awt.Point	`getLocation()` Retrieves this Pad's location
java.awt.geom.Point2D.Double	`getPadXY()` Retrieves this Pad's location
void	`setFace(int face)` Sets this pad's face.
void	`setPadXY(double xd,double yd)` Sets this pad's location
void	`setPadXY(int x,int y)` Sets this pad's location
void	`setPadXY(java.awt.Point pt)` Sets this pad's location
void	`setPadXY(java.awt.geom.Point2D.Double pt)` Sets this pad's location

void	setPadXY(java.awt.geom.Point2D pt) Sets this pad's location

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Pad

```
public Pad(int direction,
           java.awt.geom.Point2D.Double pt)
```

Creates a new Pad with the given direction and location.

Pad

```
public Pad(int direction,
           double xd,
           double yd)
```

Creates a new Pad with the given direction and location.

Methods

setPadXY

```
public void setPadXY(int x,
         int y)
```

Sets this pad's location

setPadXY

```
public void setPadXY(double xd,
         double yd)
```

Sets this pad's location

setPadXY

```
public void setPadXY(java.awt.geom.Point2D.Double pt)
```

Sets this pad's location

setPadXY

```
public void setPadXY(java.awt.geom.Point2D pt)
```

Sets this pad's location

setPadXY

```
public void setPadXY(java.awt.Point pt)
```

Sets this pad's location

getPadXY

`public java.awt.geom.Point2D.Double` **`getPadXY`**`()`

Retrieves this Pad's location

getLocation

`public java.awt.Point` **`getLocation`**`()`

Retrieves this Pad's location

setFace

`public void` **`setFace`**`(int face)`

Sets this pad's face. This value is also referred to as direction.

getFace

`public int` **`getFace`**`()`

Sets this pad's face. This value is also referred to as direction.

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)` Creates new form RealArrayDialog
public	`RealArrayDialog(java.awt.Frame parent,Real[] array,boolean fixedDimension)` 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)` Creates a new RealArrayDialog for editing the given array, setting whether the array's length can be modified.
public	`RealArrayDialog(JDialog parent,Real[] array)` Creates new form 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.
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.

Method Summary

Real[]	getArray() Gets the array that has been modified by this dialog.
boolean	isCancelled() Determins 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

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

RealArrayDialog

```
public RealArrayDialog(java.awt.Frame parent,
                       Real[] array)
```

Creates new form RealArrayDialog

RealArrayDialog

```
public RealArrayDialog(java.awt.Frame parent,
                       Real[] array,
                       boolean fixedDimension)
```

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()
```

Determins 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()` This constructor for a new RealEditor will create its own RealTextField.
public	`RealEditor(RealTextField field)` This constructor for a new RealEditor

Method Summary

boolean	`doMultiEdit()`
Object	`getCellEditorValue()` This gets the value from the RealTextField.
AbstractModel	`getModel()`
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.
void	`setModel(AbstractModel model)`

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.
static void	setUpRealEditor(JTable table) This fills all the columns in a JTable with RealEditors.
static void	setUpRealEditor(JTable table,AbstractModel model) This fills all the columns in a JTable with RealEditors.

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 clumn 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 RealRenderer

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-javax.swing.JComponent
            ┗-javax.swing.JLabel
               ┗-com.cafean.client.ui.RealRenderer
```

All Implemented Interfaces:
TableCellRenderer, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, SwingConstants

---

public class **RealRenderer**
extends JLabel
implements SwingConstants, javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, TableCellRenderer

A renderer for use in rendering a Real value in a JTable. This renderer should be paired with `RealEditor` if the value is editable.

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	`RealRenderer()` creates a new renderer
public	`RealRenderer(int alignment)` creates a new renderer with a default horizontal alignment already set.

Method Summary

`java.awt.Component`	`getTableCellRendererComponent(JTable table,Object value,boolean isSelected,boolean hasFocus,int row,int column)` Returns the component used for drawing the cell.

Methods inherited from class `javax.swing.JLabel`

getAccessibleContext, getDisabledIcon, getDisplayedMnemonic, getDisplayedMnemonicIndex, getHorizontalAlignment, getHorizontalTextPosition, getIcon, getIconTextGap, getLabelFor, getText, getUI, getUIClassID, getVerticalAlignment, getVerticalTextPosition, imageUpdate, setDisabledIcon, setDisplayedMnemonic, setDisplayedMnemonic, setDisplayedMnemonicIndex, setHorizontalAlignment, setHorizontalTextPosition, setIcon, setIconTextGap, setLabelFor, setText, setUI, setVerticalAlignment, setVerticalTextPosition, 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

RealRenderer

`public` **`RealRenderer`**`()`

creates a new renderer

RealRenderer

`public` **`RealRenderer`**`(int alignment)`

creates a new renderer with a default horizontal alignment already set.

Methods

(continued from last page)

getTableCellRendererComponent

```
public java.awt.Component getTableCellRendererComponent(JTable table,
          Object value,
          boolean isSelected,
          boolean hasFocus,
          int row,
          int column)
```

Returns the component used for drawing the cell. This method is used to configure the renderer appropriately before drawing.

Parameters:

`table` - the `JTable`that is asking the renderer to draw; can be `null`

`value` - the value of the cell to be rendered. It is up to the specific renderer to interpret and draw the value. For example, if `value`is the string "true", it could be rendered as a string or it could be rendered as a check box that is checked. `null`is a valid value

`isSelected` - true if the cell is to be rendered with the selection highlighted; otherwise false

`hasFocus` - if true, render cell appropriately. For example, put a special border on the cell, if the cell can be edited, render in the color used to indicate editing

`row` - the row index of the cell being drawn. When drawing the header, the value of `row`is -1

`column` - the column index of the cell being drawn

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	`RealTextField()` Creates a default textfield with a null string, 0 columns, and a null DecimalFormat.
public	`RealTextField(int alignment)` Creates a default textfield with a null string, 0 columns, and a null DecimalFormat.
public	`RealTextField(String text,int columns,java.text.DecimalFormat format)`
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.
public	`RealTextField(int columns,java.text.DecimalFormat format)` Creates a textfield with an inital null String as text, and the specified number of columns and format.
public	`RealTextField(String text)` Creates a textfield with the specified text, 0 columns and a null DecimalFormat
public	`RealTextField(String text,int columns)` Creates a textfield with the specified text, the specified number of columns, and a null DecimalFormat

Method Summary

void	`formatChanged()` This function is called when the document changes format
Double	`getDoubleValue()` This attempts to return the value inside this RealTextField as a Double.
java.text.DecimalFormat	`getFormat()` Getter for the DecimalFormat of this RealTextField.
Long	`getLongValue()` This attempts to return the value inside this RealTextField as a Long.
AbstractModel	`getModel()`
Number	`getNumberValue()` This attempts to return the value inside this RealTextField as a Number.
Real	`getValue()` This attempts to return the value inside this RealTextField as a Real.
void	`insertFailed(NumericPlainDocument doc,int offset,String str,AttributeSet a)` This function is overridden to hanadle an error that occurs during insertion.
boolean	`isUncertain()` Determines if the current value in the RealTextField is uncertain.

boolean	`isUnknown()` 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()` Formats the String in the RealTextField to the specified DecimalFormat.
void	`setFormat(java.text.DecimalFormat format)` Sets the DecimalFormat on this RealTextField.
void	`setModel(AbstractModel model)`
void	`setText(String text)`
void	`setValue(double d)` Sets the value of the current RealTextField to the specified double
void	`setValue(long l)` Sets the value of the current RealTextField to the specified long
void	`setValue(Number number)` Sets the value of the current RealTextField to the specified Number
void	`setValue(Real r)` Sets the value of the current RealTextField to the specified Real value.
void	`setValueUncertain()` If the RealTextField represents multiple sources of data that differ, the value is considered uncertain.
void	`setValueUnknown()` Sets the value of the current RealTextField to Unknown
String	`toString()` 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, 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

RealTextField

`public` **`RealTextField`**`()`

Creates a default textfield with a null string, 0 columns, and a null DecimalFormat.

RealTextField

`public` **`RealTextField`**`(int alignment)`

Creates a default textfield with a null string, 0 columns, and a null DecimalFormat. This also sets the default horizontal alignment.

(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 inital 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`	`refresh()` Refreshes and redisplays this dialog's data
`void`	`unitsChanged()` 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 redisplays this dialog's data

com.cafean.client.ui
Class SnapPreferences

```
java.lang.Object
   |-com.cafean.utils.Configurator
      |-com.cafean.client.ui.SnapPreferences
```

All Implemented Interfaces:
PropertyController, ComponentElement

---

public class **SnapPreferences**
extends Configurator
implements ComponentElement, PropertyController

Field Summary

static final int	DCB_MINI_NAV a behavior enumeration that indicates that double-clicking a DrawnComponent will open a MiniNavigator for that component Value: **2**
static final int	DCB_NAV_SELECT a behavior enumeration that indicates that double-clicking a DrawnComponent will select that component in the Navigator. Value: **1**
static final int	Num_Databases Value: **5**
static final int	RECENT_START Value: **11**
static final int	SOUND_BEEP Value: **1**
static final int	SOUND_OFF Value: **0**
static final int	SOUND_ON Value: **2**
static final int	WINDOW_MULTIPLE the enumerated windowing arrangement mode that indicates that each major UI portion will have it's own Frame. Value: **1**
static final String[]	WINDOW_NAMES

static final int	WINDOW_SINGLE the enumerated windowing arrangement mode that indicates that all major UI portions use a single Frame. Value: **2**
static final int[]	WINDOW_VALUES

Constructor Summary

public	SnapPreferences()

Method Summary

boolean	areComponentsScaled()
String	getAlertSound() Getter for property alertSound.
static String	getAttributeGroup(String property) Retrieves the name of the Attribute Group for the given property name.
static String[]	getAttributeGroups() Retrieves the Attribute Group Names for this preferences object.
int	getAttributeIndex(String propertyName)
static String[]	getAttributesForGroup(String groupName) Retrieves the Attribute Names for the given group.
java.awt.Color	getBaseErrorColor()
java.awt.Color	getBaseHeatedColor()
java.awt.Color	getBaseIncompleteColor()
java.awt.Color	getBaseNormalColor()
java.awt.Color	getBaseNormCsColor()
java.awt.Color	getBasePoweredColor() Getter for property poweredColorBase.
java.awt.Color	getBaseSelectedColor()
java.awt.Color	getBaseSpecialColor()
java.awt.Color	getBaseWarningColor()

int	GetCalcServeritorPort()
java.awt.Font	getCanvasFont()
java.awt.Font	getCanvasFontItalic()
java.awt.FontMetrics	getCanvasFontItalicMetrics()
java.awt.FontMetrics	getCanvasFontMetrics()
AbstractComponent	getComponent()
java.awt.Color	getConnectionColor()
int	getConnectionSize()
String	getConnectSound() Getter for property connectSound.
String	GetDatabaseHostName()
String	GetDatabaseHostName(int num)
String	GetDatabaseName()
int	getDefaultUnits() Getter for the default model units.
String	getDisconnectSound() Getter for property disconnectSound.
int	getDoubleClickBehavior() Retrieves the double-click behavior as either #DCB_NAV_SELECT or #DCB_MINI_NAV
int	GetEnaPort()
String	GetEnaServerName()
java.awt.Color	getErrorColor()
String	GetExeServer(int i)
int	GetFileArcPortNumber()
int	getHandleSize()
java.awt.Color	getHeatedColor()
java.awt.Color	getIncompleteColor()

String	`getInfoSound()` Getter for property infoSound.
String	`getInternalSound()` Getter for property internalSound.
java.io.File	`getLastPath(String name)`
int	`getLayoutIterations()` Getter for property layoutIterations.
java.awt.Font	`getMessageFont()`
AbstractModel	`getModel()`
java.awt.Color	`getNormalColor()`
java.awt.Color	`getNormCsColor()`
int	`GetNumExecServers()`
ComponentElement	`getOwner()`
int	`GetPortNumber()`
java.awt.Color	`getPoweredColor()` Getter for property poweredColor.
int	`getPropertiesOrientation()` Getter for property propertiesOrientation.
String	`getRecentFileName(int i)` Getter for property recentFile.
boolean	`getScaleComponents()`
java.awt.Color	`getSelectedColor()`
boolean	`getShowCreateViewDialog()` Returns true if the Create Views dialog should be shown when a model is imported.
boolean	`getShowWelcomeDialog()` Returns true if the welcome dialog should be shown when a the ModelEditor is opened.
int	`getSound()`
java.awt.Color	`getSpecialColor()`
String	`getUserSound()` Getter for property userSound.

java.awt.Color	getWarningColor()
String	getWarningSound() Getter for property warningSound.
int	getWindowingMode() retrieves the current windowing arrangement mode as one of the above WINDOW_* enumerations.
String	getXpdfPath() sets the path to the XPDF executable used by the PDFViewer
boolean	isBeepOn()
boolean	isDatabaseConfigured() Returns true if the SNAP Database server is configured.
boolean	isJeditAvailable() Returns true if the "jEdit Executable" property is defined and refers to an existing file.
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	isSingleFrame() Returns true if the current windowing mode is #WINDOW_SINGLE
boolean	isSoundOn()
boolean	load()
static java.awt.TexturePaint	makeBumpyTexture(java.awt.Color drawColor)
void	resetModule()
boolean	save()
void	setAlertSound(String alertSound) Setter for property alertSound.
void	setBaseErrorColor(java.awt.Color errorColor)

void	setBaseHeatedColor(java.awt.Color heatedColor)
void	setBaseIncompleteColor(java.awt.Color incompleteColor)
void	setBaseNormalColor(java.awt.Color normalColor)
void	setBaseNormCsColor(java.awt.Color normCsColor)
void	setBasePoweredColor(java.awt.Color poweredColor)
void	setBaseSelectedColor(java.awt.Color selectedColor)
void	setBaseSpecialColor(java.awt.Color specialColor)
void	setBaseWarningColor(java.awt.Color warningColor)
void	setConnectionColor(java.awt.Color c)
void	setConnectionSize(int i)
void	setConnectSound(String connectSound) Setter for property connectSound.
void	setDefaultUnits(int val_) Setter for the default model units.
void	setDisconnectSound(String disconnectSound) Setter for property disconnectSound.
void	setDoubleClickBehavior(int doubleClickBehavior) Sets the double-click behavior as either #DCB_NAV_SELECT or #DCB_MINI_NAV
void	setErrorColor(java.awt.Color c)
void	setHandleSize(int i)
void	setHeatedColor(java.awt.Color c)
void	setIncompleteColor(java.awt.Color c)
void	setInfoSound(String infoSound) Setter for property infoSound.
void	setInternalSound(String internalSound) Setter for property internalSound.
void	setLastPath(String name,java.io.File last)
void	setLayoutIterations(int layoutIterations) Setter for property layoutIterations.

void	setNormalColor(java.awt.Color c)
void	setNormCsColor(java.awt.Color c)
void	setPoweredColor(java.awt.Color c)
void	setPropertiesOrientation(int propertiesOrientation) Setter for property propertiesOrientation.
void	setRecentFileName(int i,String recent) Sets the recent file name at the given index.
void	setScaleComponents(boolean flag)
void	setSelectedColor(java.awt.Color c)
void	setShowCreateViewDialog(boolean show) Set to true if the Create Views dialog should be shown when a model is imported.
void	setShowHS_connections(boolean flag)
void	setShowWelcomeDialog(boolean show) Set to true if the welcome dialog should be shown when a the ModelEditor is opened.
void	setSound(int val_)
void	setSpecialColor(java.awt.Color c)
void	setUserSound(String userSound) Setter for property userSound.
void	setWarningColor(java.awt.Color c)
void	setWarningSound(String warningSound) Setter for property warningSound.
void	setWindowingMode(int mode) sets the current windowing arrangement mode as one of the above WINDOW_* enumerations.
void	setXpdfPath(String path) gets the path to the XPDF executable used by the PDFViewer
boolean	ShowHS_connections()
boolean	UseExecServer()

Methods inherited from class com.cafean.utils.Configurator

addCodeInfos, addServer, destroy, getCodeInfos, getColor, getColor, getConfigFileDir, getConfigFileName, getDividerLocation, getFont, getFont, getGeometry, getGeometry, getGeometry, getGeometry, getGeometry, getNrcDBServer, getProperty, getProperty, getServers, getServers, getSnapDBServer, getWebSupport, reDirectStdErrOut, removeAllCodeInfos, removeAllServers, removeServer, save, setColor, setDBServer, setDividerLocation, setFont, setGeometry, setGeometry, setLookAndFeel, setModule, setNrcDBServer, setProperty, setWebSupport, updateAvailable

Methods inherited from class `java.lang.Object`

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

Fields

WINDOW_MULTIPLE

`public static final int` **`WINDOW_MULTIPLE`**

the enumerated windowing arrangement mode that indicates that each major UI portion will have it's own Frame.

WINDOW_SINGLE

`public static final int` **`WINDOW_SINGLE`**

the enumerated windowing arrangement mode that indicates that all major UI portions use a single Frame.

WINDOW_VALUES

`public static final int` **`WINDOW_VALUES`**

WINDOW_NAMES

`public static final java.lang.String` **`WINDOW_NAMES`**

Num_Databases

`public static final int` **`Num_Databases`**

SOUND_OFF

`public static final int` **`SOUND_OFF`**

SOUND_BEEP

`public static final int` **`SOUND_BEEP`**

SOUND_ON

`public static final int` **`SOUND_ON`**

(continued from last page)

RECENT_START

`public static final int` **`RECENT_START`**

DCB_NAV_SELECT

`public static final int` **`DCB_NAV_SELECT`**

a behavior enumeration that indicates that double-clicking a DrawnComponent will select that component in the Navigator.

DCB_MINI_NAV

`public static final int` **`DCB_MINI_NAV`**

a behavior enumeration that indicates that double-clicking a DrawnComponent will open a MiniNavigator for that component

Constructors

SnapPreferences

`public` **`SnapPreferences`**`()`

Methods

resetModule

`public void` **`resetModule`**`()`

load

`public boolean` **`load`**`()`

save

`public boolean` **`save`**`()`

isJeditAvailable

`public boolean` **`isJeditAvailable`**`()`

Returns true if the "jEdit Executable" property is defined and refers to an existing file.

isDatabaseConfigured

`public boolean` **`isDatabaseConfigured`**`()`

Returns true if the SNAP Database server is configured.

(continued from last page)

getDoubleClickBehavior

`public int getDoubleClickBehavior()`

Retrieves the double-click behavior as either #DCB_NAV_SELECT or #DCB_MINI_NAV

setDoubleClickBehavior

`public void setDoubleClickBehavior(int doubleClickBehavior)`

Sets the double-click behavior as either #DCB_NAV_SELECT or #DCB_MINI_NAV

GetDatabaseHostName

`public String GetDatabaseHostName(int num)`

GetDatabaseHostName

`public String GetDatabaseHostName()`

GetDatabaseName

`public String GetDatabaseName()`

GetEnaServerName

`public String GetEnaServerName()`

GetPortNumber

`public int GetPortNumber()`

GetFileArcPortNumber

`public int GetFileArcPortNumber()`

GetCalcServeritorPort

`public int GetCalcServeritorPort()`

GetEnaPort

`public int GetEnaPort()`

GetNumExecServers

`public int GetNumExecServers()`

GetExeServer

public String **GetExeServer**(int i)

UseExecServer

public boolean **UseExecServer**()

areComponentsScaled

public boolean **areComponentsScaled**()

setScaleComponents

public void **setScaleComponents**(boolean flag)

getScaleComponents

public boolean **getScaleComponents**()

isSoundOn

public boolean **isSoundOn**()

isBeepOn

public boolean **isBeepOn**()

getSound

public int **getSound**()

setSound

public void **setSound**(int val_)

ShowHS_connections

public boolean **ShowHS_connections**()

(continued from last page)

setShowHS_connections

```
public void setShowHS_connections(boolean flag)
```

getHandleSize

```
public int getHandleSize()
```

setHandleSize

```
public void setHandleSize(int i)
```

getConnectionSize

```
public int getConnectionSize()
```

setConnectionSize

```
public void setConnectionSize(int i)
```

getNormalColor

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

getNormCsColor

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

getWarningColor

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

getErrorColor

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

getIncompleteColor

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

getSelectedColor

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

getSpecialColor

public java.awt.Color **getSpecialColor**()

getHeatedColor

public java.awt.Color **getHeatedColor**()

getConnectionColor

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

getBaseNormalColor

public java.awt.Color **getBaseNormalColor**()

getBaseNormCsColor

public java.awt.Color **getBaseNormCsColor**()

getBaseWarningColor

public java.awt.Color **getBaseWarningColor**()

getBaseErrorColor

public java.awt.Color **getBaseErrorColor**()

getBaseIncompleteColor

public java.awt.Color **getBaseIncompleteColor**()

getBaseSelectedColor

public java.awt.Color **getBaseSelectedColor**()

getBaseSpecialColor

public java.awt.Color **getBaseSpecialColor**()

(continued from last page)

getBaseHeatedColor

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

setBaseNormalColor

```
public void setBaseNormalColor(java.awt.Color normalColor)
```

setBaseNormCsColor

```
public void setBaseNormCsColor(java.awt.Color normCsColor)
```

setBaseWarningColor

```
public void setBaseWarningColor(java.awt.Color warningColor)
```

setBaseErrorColor

```
public void setBaseErrorColor(java.awt.Color errorColor)
```

setBaseIncompleteColor

```
public void setBaseIncompleteColor(java.awt.Color incompleteColor)
```

setBaseSelectedColor

```
public void setBaseSelectedColor(java.awt.Color selectedColor)
```

setBaseSpecialColor

```
public void setBaseSpecialColor(java.awt.Color specialColor)
```

setBaseHeatedColor

```
public void setBaseHeatedColor(java.awt.Color heatedColor)
```

setNormalColor

```
public void setNormalColor(java.awt.Color c)
```

setNormCsColor

```
public void setNormCsColor(java.awt.Color c)
```

setWarningColor

public void **setWarningColor**(java.awt.Color c)

setErrorColor

public void **setErrorColor**(java.awt.Color c)

setIncompleteColor

public void **setIncompleteColor**(java.awt.Color c)

setSelectedColor

public void **setSelectedColor**(java.awt.Color c)

setSpecialColor

public void **setSpecialColor**(java.awt.Color c)

setHeatedColor

public void **setHeatedColor**(java.awt.Color c)

setConnectionColor

public void **setConnectionColor**(java.awt.Color c)

getCanvasFont

public java.awt.Font **getCanvasFont**()

getCanvasFontItalic

public java.awt.Font **getCanvasFontItalic**()

getCanvasFontItalicMetrics

public java.awt.FontMetrics **getCanvasFontItalicMetrics**()

(continued from last page)

getCanvasFontMetrics

```
public java.awt.FontMetrics getCanvasFontMetrics()
```

makeBumpyTexture

```
public static java.awt.TexturePaint makeBumpyTexture(java.awt.Color drawColor)
```

getMessageFont

```
public java.awt.Font getMessageFont()
```

getRecentFileName

```
public String getRecentFileName(int i)
```

Getter for property recentFile.

Returns:

Value of property recentFile.

setRecentFileName

```
public void setRecentFileName(int i,
          String recent)
```

Sets the recent file name at the given index.

getLastPath

```
public java.io.File getLastPath(String name)
```

setLastPath

```
public void setLastPath(String name,
          java.io.File last)
```

getPoweredColor

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

Getter for property poweredColor.

Returns:

Value of property poweredColor.

setPoweredColor

```
public void setPoweredColor(java.awt.Color c)
```

(continued from last page)

getBasePoweredColor

`public java.awt.Color` **`getBasePoweredColor`**`()`

Getter for property poweredColorBase.

Returns:

Value of property poweredColorBase.

---

setBasePoweredColor

`public void` **`setBasePoweredColor`**`(java.awt.Color poweredColor)`

---

getPropertiesOrientation

`public int` **`getPropertiesOrientation`**`()`

Getter for property propertiesOrientation.

Returns:

Value of property propertiesOrientation.

---

setPropertiesOrientation

`public void` **`setPropertiesOrientation`**`(int propertiesOrientation)`

Setter for property propertiesOrientation.

Parameters:

`propertiesOrientation` - New value of property propertiesOrientation.

---

setWindowingMode

`public void` **`setWindowingMode`**`(int mode)`

sets the current windowing arrangement mode as one of the above WINDOW_* enumerations.

---

getWindowingMode

`public int` **`getWindowingMode`**`()`

retrieves the current windowing arrangement mode as one of the above WINDOW_* enumerations.

---

isSingleFrame

`public boolean` **`isSingleFrame`**`()`

Returns true if the current windowing mode is #WINDOW_SINGLE

---

getComponent

`public` AbstractComponent **`getComponent`**`()`

---

getModel

`public` AbstractModel **`getModel`**`()`

getOwner

`public ComponentElement getOwner()`

getAttributeGroups

`public static String[] getAttributeGroups()`

Retrieves the Attribute Group Names for this preferences object.

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 containg 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.

getLayoutIterations

`public int getLayoutIterations()`

Getter for property layoutIterations.

Returns:

Value of property layoutIterations.

setLayoutIterations

`public void setLayoutIterations(int layoutIterations)`

Setter for property layoutIterations.

Parameters:

`layoutIterations -` New value of property layoutIterations.

getAlertSound

`public String getAlertSound()`

Getter for property alertSound.

Returns:

(continued from last page)

Value of property alertSound.

setAlertSound

`public void setAlertSound(String alertSound)`

Setter for property alertSound.

Parameters:

`alertSound` - New value of property alertSound.

getInfoSound

`public String getInfoSound()`

Getter for property infoSound.

Returns:

Value of property infoSound.

setInfoSound

`public void setInfoSound(String infoSound)`

Setter for property infoSound.

Parameters:

`infoSound` - New value of property infoSound.

getWarningSound

`public String getWarningSound()`

Getter for property warningSound.

Returns:

Value of property warningSound.

setWarningSound

`public void setWarningSound(String warningSound)`

Setter for property warningSound.

Parameters:

`warningSound` - New value of property warningSound.

getUserSound

`public String getUserSound()`

Getter for property userSound.

Returns:

Value of property userSound.

setUserSound

`public void setUserSound(String userSound)`

Setter for property userSound.

(continued from last page)

Parameters:
`userSound` - New value of property userSound.

getInternalSound

`public String getInternalSound()`

Getter for property internalSound.

Returns:
Value of property internalSound.

setInternalSound

`public void setInternalSound(String internalSound)`

Setter for property internalSound.

Parameters:
`internalSound` - New value of property internalSound.

getConnectSound

`public String getConnectSound()`

Getter for property connectSound.

Returns:
Value of property connectSound.

setConnectSound

`public void setConnectSound(String connectSound)`

Setter for property connectSound.

Parameters:
`connectSound` - New value of property connectSound.

getDisconnectSound

`public String getDisconnectSound()`

Getter for property disconnectSound.

Returns:
Value of property disconnectSound.

setDisconnectSound

`public void setDisconnectSound(String disconnectSound)`

Setter for property disconnectSound.

Parameters:
`disconnectSound` - New value of property disconnectSound.

getDefaultUnits

`public int getDefaultUnits()`

Getter for the default model units.

setDefaultUnits

`public void setDefaultUnits(int val_)`

Setter for the default model units.

getShowCreateViewDialog

`public boolean getShowCreateViewDialog()`

Returns true if the Create Views dialog should be shown when a model is imported.

setShowCreateViewDialog

`public void setShowCreateViewDialog(boolean show)`

Set to true if the Create Views dialog should be shown when a model is imported.

getShowWelcomeDialog

`public boolean getShowWelcomeDialog()`

Returns true if the welcome dialog should be shown when a the ModelEditor is opened.

setShowWelcomeDialog

`public void setShowWelcomeDialog(boolean show)`

Set to true if the welcome dialog should be shown when a the ModelEditor is opened.

isPropertyEnabled

`public boolean isPropertyEnabled(String propertyName)`

getXpdfPath

`public String getXpdfPath()`

sets the path to the XPDF executable used by the PDFViewer

setXpdfPath

`public void setXpdfPath(String path)`

gets the path to the XPDF executable used by the PDFViewer

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)
```

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 algorthm 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

See Also:
#addMouseListenerToHeaderInTable

Constructor Summary

public	`TableSorter()` Creates a new instance of Table Sorter
public	`TableSorter(TableModel model)` Creates an instance of table sorter given a table model

Method Summary

void	`addMouseListenerToHeaderInTable(JTable table)` Adds a mouse litener to the table header.
void	`checkModel()` Verifies the current model has the same size as the data stored by the table sorter.
int	`compare(int row1,int row2)` Compares the values stored in two rows of the table.

int	`compareRowsByColumn(int row1,int row2,int column)` Compares the value stored in two rows of a given column.
Class	`getColumnClass(int aColumn)` Returns the Class of the given column.
int	`getColumnCount()` Returns the number of columns in the tablemodel.
String	`getColumnName(int aColumn)` Returns the name of the given column.
int	`getIndexForRow(int aRow)` Finds the actual index given a specific table row.
TableModel	`getModel()` Accessor for the TableModel of the JTable this is sorting
int	`getRowCount()` Returns the number of rows in the tablemodel.
Object	`getValueAt(int aRow,int aColumn)` Passes through the value stored in the sorted locations.
boolean	`isCellEditable(int row,int column)` Determines if the cell specified by row and column is editable.
void	`setModel(TableModel model)` Setter for the TableModel of the JTable this is sorting
void	`setValueAt(Object aValue,int aRow,int aColumn)` Sets the value stored in the sorted locations.
void	`sort(Object sender)` Arranges the rows of a table so that they are sorted.
void	`sortByColumn(int column)` Sorts the table by the given column in ascending order.
void	`sortByColumn(int column,boolean ascending)` Sorts the table by the given column, in either ascending or decending order.
void	`tableChanged(TableModelEvent e)` 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 java.awt.Cursor	CURSOR_MANIP_P 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 The enumeration for the Connect tool Value: **5**
static final int	TOOL_INSERT The enumeration for the Insert tool Value: **4**
static final int	TOOL_INTERACTIVE The enumeration for the Interactive tool Value: **6**

static final int	TOOL_NONE No tool has been selected Value: **0**
static final int	TOOL_PAN The enumeration for the Pan tool Value: **2**
static final int	TOOL_SELECT The enumeration for the select Tool Value: **1**
static final int	TOOL_ZOOM The enumeration for the Zoom tool Value: **3**

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	Toolbox(DrawnView parent) Creates new Toolbox

Method Summary

void	addToolChangeListener(ToolChangeListener listener) Adds the given listener to the the list to notify when the current tool has changed.
void	createBeanSelectionMenu(JMenu menu) Creates a select menu appropriate for selecting AbstractDisplayBean instances within the DrawnView parent of this Toolbox.
JToolBar[]	createCategoryToolBars() Creates a set of JToolBar instances appropriate for allowing quick selection of types for the Insertion Tool.
static java.awt.Cursor	createCursor(String resource) Creates a Cursor from the image referred to by the given resource name.
JComponent[]	createDisplayBeans(int pixelsPerMeter,double widthScaleFactor,DrawnComponent dc) Attempts to create a set of AbstractDisplayBeans from the given DrawnComponent using DrawnComponent#createDisplayBeans.

JComponent[]	createDisplayBeans(int pixelsPerMeter,DrawnComponent dc) Attempts to create a set of AbstractDisplayBeans from the given DrawnComponent using DrawnComponent#createDisplayBeans.
JComponent	createSelected() 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) Retrieves the Class with the given class name in the list of Display Bean classes available.
Category	getComponentCategory() Retrieves the Category of the currently selected component type in the palette of the Toolbox.
java.awt.Cursor	getCurrentCursor() Retrieves the cursor for the views based off of the currently selected tool.
java.awt.Cursor	getCurrentCursor(java.awt.event.MouseEvent evt) Retrieves the cursor for the views based off of the currently selected tool.
int	getCurrentTool() Retrieves the enumerated type of the current tool.
boolean	isCompActionSelected() Returns true if the selected toolbox action will create an AbstractComponent.
static com.cafean.client.anim.AbstractDisplayBean	loadDisplayBean(DrawnDisplayBeanRec rec) Loads a display bean from the given display bean rec.
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.
boolean	loadVedaExport(java.io.File exportedFile) Attempts to import a VEDA mask dump file.
void	removeToolChangeListener(ToolChangeListener listener) Adds the given listener to the the list to notify when the current tool has changed.
void	setCurrentAction(ToolboxAction action) Sets the current ToolboxAction used for creating new drawings with the Insertion Tool.
void	setCurrentTool(int tool) switches the toolbox to the given tool.
void	setLocked(boolean locked) Updates the state of this Toolbox to correspond with the given locked state.
static byte[]	storeDisplayBean(com.cafean.client.anim.AbstractDisplayBean bean) XML Encodes the given bean and returns the byte[] result of the encoding.

static boolean	verifyBeansAvailable(Vector records) 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, 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

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 `AbstractDisplayBean`s stored in the given Vector as `DrawnDisplayBeanRec`s 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 UndoableConnect

```
java.lang.Object
   |
   +-javax.swing.undo.AbstractUndoableEdit
      |
      +-com.cafean.client.ui.UndoableConnect
```

public class **UndoableConnect**
extends AbstractUndoableEdit

This class is an undoable connect

Constructor Summary

public	`UndoableConnect(AbstractComponent left,AbstractComponent right)` creates a new undoable connection to allow the connection of the given left component to the given right component to be undone.

Method Summary

boolean	`canRedo()`
boolean	`canUndo()`
void	`end()` Gets the post-edit state of the StateEditable object and ends the edit.
String	`getPresentationName()`
void	`redo()`
void	`setConnection(Connection connection)` This method stores a reference to the newly created connection object for use in undoing and redoing the connection
void	`undo()`

Methods inherited from class `javax.swing.undo.AbstractUndoableEdit`

`addEdit, canRedo, canUndo, die, getPresentationName, getRedoPresentationName, getUndoPresentationName, isSignificant, redo, replaceEdit, toString, undo`

Methods inherited from class `java.lang.Object`

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

Constructors

UndoableConnect

```
public UndoableConnect(AbstractComponent left,
                       AbstractComponent right)
```

creates a new undoable connection to allow the connection of the given left component to the given right component to be undone. This object should be created before the connection is established and setConnection called afterward.

Methods

setConnection

```
public void setConnection(Connection connection)
```

This method stores a reference to the newly created connection object for use in undoing and redoing the connection

---

end

```
public void end()
```

Gets the post-edit state of the StateEditable object and ends the edit.

---

canRedo

```
public boolean canRedo()
```

---

canUndo

```
public boolean canUndo()
```

---

undo

```
public void undo()
  throws CannotUndoException
```

---

redo

```
public void redo()
  throws CannotRedoException
```

---

getPresentationName

```
public String getPresentationName()
```

com.cafean.client.ui
Class UndoableDelete

```
java.lang.Object
   |-javax.swing.undo.AbstractUndoableEdit
      |-com.cafean.client.ui.UndoableDelete
```

public class **UndoableDelete**
extends AbstractUndoableEdit

This class is an UndoableAction which deletes components.

Constructor Summary

public	`UndoableDelete(Vector deleted)` Creates an undoable object for use in undoing the deletion of the givne components.
public	`UndoableDelete(AbstractComponent[] deleted)` Creates an undoable object for use in undoing the deletion of the givne components.

Method Summary

boolean	`canRedo()` returns true
boolean	`canUndo()` returns true
String	`getPresentationName()`
void	`redo()` Redoes the deletion of this object's components by deleting them again
void	`undo()` Undoes the deletion of this object's components by reinserting them into the model.

Methods inherited from class `javax.swing.undo.AbstractUndoableEdit`

`addEdit, canRedo, canUndo, die, getPresentationName, getRedoPresentationName, getUndoPresentationName, isSignificant, redo, replaceEdit, toString, undo`

Methods inherited from class `java.lang.Object`

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

(continued from last page)

Constructors

UndoableDelete

`public` **`UndoableDelete`**`(Vector deleted)`

Creates an undoable object for use in undoing the deletion of the givne components.

Parameters:

`components` - a Vector of AbstractComponent objects to be deleted.

UndoableDelete

`public` **`UndoableDelete`**`(AbstractComponent[] deleted)`

Creates an undoable object for use in undoing the deletion of the givne components.

Parameters:

`components` - an AbstractComponent[] of components to be deleted.

Methods

canRedo

`public boolean` **`canRedo`**`()`

returns true

---

canUndo

`public boolean` **`canUndo`**`()`

returns true

---

undo

`public void` **`undo`**`()`
`throws CannotUndoException`

Undoes the deletion of this object's components by reinserting them into the model.

---

redo

`public void` **`redo`**`()`
`throws CannotRedoException`

Redoes the deletion of this object's components by deleting them again

---

getPresentationName

`public String` **`getPresentationName`**`()`

com.cafean.client.ui
Class UndoableDisconnect

```
java.lang.Object
   |
   +-javax.swing.undo.AbstractUndoableEdit
      |
      +-com.cafean.client.ui.UndoableDisconnect
```

public class **UndoableDisconnect**
extends AbstractUndoableEdit

This class is an undoable disconnect

Constructor Summary

public	`UndoableDisconnect(AbstractComponent left,AbstractComponent right)` creates a new undoable connection to allow the disconnection of the given left component to the given right component to be undone.

Method Summary

boolean	`canRedo()` returns true
boolean	`canUndo()` returns true
void	`end()` Gets the post-edit state of the StateEditable object and ends the edit.
String	`getPresentationName()` returns "Disconnect"
void	`redo()` Redoes this disconnection by disconnecting
void	`setConnection(Connection connection)`
void	`undo()` Undoes this disconnection by re-establishing the conneciton

Methods inherited from class `javax.swing.undo.AbstractUndoableEdit`

`addEdit, canRedo, canUndo, die, getPresentationName, getRedoPresentationName, getUndoPresentationName, isSignificant, redo, replaceEdit, toString, undo`

Methods inherited from class `java.lang.Object`

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

Constructors

UndoableDisconnect

```
public UndoableDisconnect(AbstractComponent left,
                          AbstractComponent right)
```

creates a new undoable connection to allow the disconnection of the given left component to the given right component to be undone. This object should be created before the connection disconnected and setConnection called afterward.

Methods

setConnection

```
public void setConnection(Connection connection)
```

end

```
public void end()
```

Gets the post-edit state of the StateEditable object and ends the edit.

canRedo

```
public boolean canRedo()
```

returns true

canUndo

```
public boolean canUndo()
```

returns true

undo

```
public void undo()
  throws CannotUndoException
```

Undoes this disconnection by re-establishing the conneciton

redo

```
public void redo()
  throws CannotRedoException
```

Redoes this disconnection by disconnecting

getPresentationName

```
public String getPresentationName()
```

returns "Disconnect"

com.cafean.client.ui
Class UndoableInsert

```
java.lang.Object
   |-javax.swing.undo.AbstractUndoableEdit
      |-com.cafean.client.ui.UndoableInsert
```

public class **UndoableInsert**
extends AbstractUndoableEdit

A simple UndoableEdit for use in undo-redo of a component insertion.

Constructor Summary

public	`UndoableInsert(AbstractComponent component)` Creates a new undo object for use in undoing the insert of the given component.

Method Summary

boolean	`canRedo()` Returns true
boolean	`canUndo()` Returns true
String	`getPresentationName()` returns "Insert Component"
void	`redo()` redoes the insertion of the component by re-inserting it
void	`undo()` Undoes the insertion of the component by deleting it

Methods inherited from class `javax.swing.undo.AbstractUndoableEdit`

`addEdit, canRedo, canUndo, die, getPresentationName, getRedoPresentationName, getUndoPresentationName, isSignificant, redo, replaceEdit, toString, undo`

Methods inherited from class `java.lang.Object`

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

Constructors

(continued from last page)

UndoableInsert

`public` **`UndoableInsert`**`(`[`AbstractComponent`]` component)`

Creates a new undo object for use in undoing the insert of the given component.

Methods

canRedo

`public boolean` **`canRedo`**`()`

Returns true

---

canUndo

`public boolean` **`canUndo`**`()`

Returns true

---

undo

`public void` **`undo`**`()`
`  throws CannotUndoException`

Undoes the insertion of the component by deleting it

---

redo

`public void` **`redo`**`()`
`  throws CannotRedoException`

redoes the insertion of the component by re-inserting it

---

getPresentationName

`public String` **`getPresentationName`**`()`

returns "Insert Component"

com.cafean.client.ui
Class ZoomablePanel

```
java.lang.Object
   |
   +-java.awt.Component
      |
      +-java.awt.Container
         |
         +-javax.swing.JComponent
            |
            +-javax.swing.JPanel
               |
               +-com.cafean.client.ui.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)` Constructs a new Zoomable Panel in a DrawnView.

Method Summary

AbstractButton	`addMouseHandler(MouseHandler handler)` Adds the given handler to the list of zoomable panel handlers and Toolbox handlers.
void	`addNotify()` Initializes this panel in response to its addition to another component.

BeanBox	`getBeanBox()` The BeanBox contains all the components that are visible inside the DrawnView.
DrawnView	`getDrawnView()` Each Zoomable Panel appears inside a scroll panel in a DrawnView.
GlassPanel	`getGlassPane()` Access routine for the glass pane.
java.awt.Dimension	`getMaximumSize()` Return the maximum size of this component.
java.awt.Dimension	`getMinimumSize()` Return the minimum size of this component.
MouseHandler	`getMouseHandler()` Gets the currently active MouseHandler based on the current tool.
java.awt.Dimension	`getPanelSize()` The PanelSize is the size of the canvas below this zoomable panel.
java.awt.Dimension	`getPreferredSize()` Return the preferred size of this component.
double	`getScale()` The scale factor is the factor by which the current view is multiplied when zooming in or out.
JScrollPane	`getScrollPane()` Each Zoomable Panel appears inside a scroll panel in a DrawnView.
java.awt.Point	`getViewportCenter()` Calculates the center of the currently visible position of the scrollpane containing this ZoomablePanel.
java.awt.Dimension	`getZoomSize()` Return the size of scaled components.
java.awt.Point	`inverseTransformPoint(java.awt.Point p1)` The reverse transform of a point backwards throught the scaling factor.
void	`refresh()` Forces a repaint of all children.
void	`refreshCompBounds(java.awt.Rectangle componentBounds)` Repaints the given component-space bounds.
void	`removeMouseHandler(MouseHandler handler)` Removes the given handler from the list of zoomable panel handlers and Toolbox handlers.
void	`revalidate()`
void	`setPanelSize(java.awt.Dimension size)` Sets the size of the canvas in the view.

void	`setScale(double scale)` Set the scaling factor property.
void	`setViewCenter(java.awt.Point center)` This sets the viewports center to the given Point.
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.
java.awt.Point	`transformPoint(java.awt.Point p1)` Translate the coordinates of a point to account for the scaling factor.
void	`zoomToFit(java.awt.Rectangle rect)` 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, 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

(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 MouseHandeler 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 toobox.

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 prefered 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 prefered 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 throught 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 JScrollPanel, 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()` Creates a new instance of Annotation

Method Summary

`void`	`addPopupMenuItems(JPopupMenu menu,java.awt.event.MouseEvent evt)`
`DrawnView`	`getDrawnView()` 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) Move the Annotation so the center is the specified position
abstract void	popupEditor() This is used for Annotations to generate their editing dialogs.
void	refresh(java.awt.Rectangle r) This refreshses a given Rectangle that indicates a dirty region.
abstract void	resetSize() This resets the current size of the annotation.
void	restoreState(Hashtable state) 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) 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, 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

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 it's 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 refreshses 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

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-javax.swing.JComponent
            ┗-javax.swing.JPanel
               ┗-com.cafean.client.ui.annotation.Annotation
                  ┗-com.cafean.client.ui.annotation.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.
`AbstractInsertHandler`	`getNewInsertHandler(ZoomablePanel parent)`
`boolean`	`isFilled()`

void	paintComponent(java.awt.Graphics g) {@inheritDoc}
void	popupEditor() {@inheritDoc}
void	resetSize() {@inheritDoc}
void	restoreState(Hashtable state) Restore the state of the bean from an earlier edit.
void	setEqualTo(EllipticalAnnotation annotation) Sets this annotation equal to the given annotation
void	setFilled(boolean filled)
void	setLineThickness(int lineThickness) 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, 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

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

```
java.lang.Object
   |-java.awt.Component
      |-java.awt.Container
         |-javax.swing.JComponent
            |-javax.swing.JPanel
               |-com.cafean.client.ui.annotation.Annotation
                  |-com.cafean.client.ui.annotation.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() Retrieves this annotation's image data.
java.awt.Dimension	getMinimumSize()
void	paintComponent(java.awt.Graphics g) Draw a simple X'd box if there is no image.
void	popupEditor()
void	reset() Creates a draw view component and puts it in the drawing vectors.
void	resetSize()
void	restoreState(Hashtable state) Restore the state of the bean from an earlier edit.
void	setImageData(byte[] imageData) Sets this annotation's image data and resizes to the size of the image in imageData.
void	setImageData(byte[] imageData,boolean resize) Sets this annotation's image data and optionally resizes to the size of the image in imageData.
void	storeState(Hashtable state) 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, 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

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

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-javax.swing.JComponent
            ┗-javax.swing.JPanel
               ┗-com.cafean.client.ui.annotation.Annotation
                  ┗-com.cafean.client.ui.annotation.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 Value: **1**
static final int	ARROW_HOLLOW Value: **2**
static final int	ARROW_LINE Value: **4**
static final String[]	ARROW_NAMES
static final int	ARROW_NONE 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

`public`	`LineAnnotation()` Creates a new instance of LineAnnotation

Method Summary

`boolean`	`addPoint(java.awt.Point p)` Adds a path point at the given point, if the point falls on an existing line segment.
`void`	`addPopupMenuItems(JPopupMenu menu,java.awt.event.MouseEvent evt)`
`boolean`	`canAddPoint(java.awt.Point p)` Return true if a point can be added to the path.
`boolean`	`canRemovePoint(java.awt.Point p)` Return true if a point can be removed from the path.
`boolean`	`contains(double x,double y)`
`boolean`	`contains(int x,int y)`
`boolean`	`contains(java.awt.Point p)`
`int[]`	`getArrows()` Getter for property arrows.
`float`	`getArrowSize()` Getter for property arrowSize.
`static String`	`getArrowTypeName(int type)`
`int`	`getHead1()`
`int`	`getHead2()`
`int`	`getLineThickness()` Getter for property lineThickness.
`AbstractInsertHandler`	`getNewInsertHandler(ZoomablePanel parent)`
`java.awt.Point[]`	`getPath()` Retrieves a copy of the plotted path points that this DrawnConnection is painting.

String	`getToolTipText()` {@inheritDoc}
java.awt.Rectangle	`getUsedBounds()`
boolean	`isDashed()` Getter for property dashed.
boolean	`isObjectInsideBounds(java.awt.geom.Rectangle2D.Double rect)` {@inheritDoc}
void	`mouseClicked(java.awt.event.MouseEvent e)`
void	`mouseDragged(java.awt.event.MouseEvent e)` 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)` {@inheritDoc}
void	`mousePressed(java.awt.event.MouseEvent e)` Handle mousePressed events to support segment and point manipulation
void	`mouseReleased(java.awt.event.MouseEvent e)` Handle mouseReleased events to support segment and point manipulation
void	`moveTo(double x,double y,boolean last)` Move the Annotation so the center is the specified position
void	`paintComponent(java.awt.Graphics g)`
void	`popupEditor()`
void	`removeClosestPoint(java.awt.Point p)` Attempts to remove the closest point in this DrawnConnection's set of points.
boolean	`removePoint(java.awt.Point p)` Attempts to remove the path point at the given location.
void	`repaint()`
void	`resetSize()`
void	`restoreState(Hashtable state)` Restore the state of the bean from an earlier edit.
void	`revalidate()`

void	setArrows(int[] arrows) Setter for property arrows.
void	setArrowSize(float arrowSize) Setter for property arrowSize.
void	setDashed(boolean dashed) Setter for property dashed.
void	setEqualTo(LineAnnotation annotation) Sets this annotation equal to the given annotation
void	setHead1(int head1)
void	setHead2(int head2)
void	setLineThickness(int lineThickness) Setter for property lineThickness.
void	setPath(java.awt.Point[] path) Sets the path points that this DrawnConnection will use for connecting.
void	storeState(Hashtable state) 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, 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

Fields

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

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-javax.swing.JComponent
            ┗-javax.swing.JPanel
               ┗-com.cafean.client.ui.annotation.Annotation
                  ┗-com.cafean.client.ui.annotation.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()` Getter for property lineThickness.
AbstractInsertHandler	`getNewInsertHandler(ZoomablePanel parent)`
boolean	`isFilled()`
boolean	`isOpaque()`
boolean	`isRounded()` Returns true if this annotation has rounded corners.
void	`paintComponent(java.awt.Graphics g)`
void	`popupEditor()`
void	`resetSize()` {@inheritDoc}
void	`setArcHeight(double arcHeight)` Setter for property arcHeight.
void	`setArcWidth(double arcWidth)` Setter for property arcWidth.
void	`setEqualTo(RectangularAnnotation annotation)` Sets this annotation equal to the given annotation
void	`setFilled(boolean filled)`
void	`setLineThickness(int lineThickness)` Setter for property lineThickness.
void	`setRounded(boolean rounded)` 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, 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

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

```
java.lang.Object
   |-java.awt.Component
      |-java.awt.Container
         |-javax.swing.JComponent
            |-javax.swing.JPanel
               |-com.cafean.client.ui.annotation.Annotation
                  |-com.cafean.client.ui.annotation.TextAnnotation
```

public class **TextAnnotation**
extends Annotation

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	`TextAnnotation()` Creates a new instance of TextAnnotation

Method Summary

void	`ensureVersion21()` 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()` Getter for property label.

String	getText()
void	popupEditor()
void	resetSize() Resets the size of the label and the text annotation to be slightly larger then the prefered size of the label.
void	restoreState(Hashtable state) 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) Setter for property label.
void	setSize(java.awt.Dimension size)
void	setText(String text)
void	storeState(Hashtable state) 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, 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

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 prefered 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 ths 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()` Creates a new instance of CellComponentSelector

Method Summary

`String`	`getAsText()`
`java.awt.Component`	`getCustomEditor()`
`String`	`getJavaInitializationString()` Does nothing, and returns null.

String[]	getTags() 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, 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

(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()` Constructor.

Method Summary

void	`editorChangeValue(Border newBorder)` Update the user interface to reflect the edited border properties.
String	`getAsText()` The property value as a human editable string.
int	`getBorderType(Border bord)` Determine the border type for a given border.
java.awt.Component	`getCustomEditor()` A component that will allow direct editing of the current property value.
String	`getJavaInitializationString()` This method is intended for use when generating Java code to set the value of the property.
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.
Object	`getValue()` Access routine for the channel name.
boolean	`isPaintable()` True if the class will honor the paintValue method.

void	`itemStateChanged(java.awt.event.ItemEvent evt)` Invoked when the combobox's state has been changed.
void	`paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)` Paint a representation of the value into a given area of screen real estate.
void	`setAsText(String s)` Set the property value by parsing a given String.
void	`setValue(Object o)` Set (or change) the object that is to be edited.
boolean	`supportsCustomEditor()` True if the propertyEditor 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)` Creates a new instance of CellComponentSelector
public	`ComponentSelectionEditor()` Creates a new instance of CellComponentSelector

Method Summary

`String`	`getAsText()`
`Category`	`getCategory()` Gets the Category for this ComponentSelectionEditor.
`java.awt.Component`	`getCustomEditor()`
`String`	`getJavaInitializationString()` Does nothing, and returns null.
`AbstractModel`	`getModel()`
`TableCellEditor`	`getTableCellEditor()` Method allows this panel to be used in JTable as an editor
`String[]`	`getTags()` Does nothing, and returns null.
`Object`	`getValue()`

boolean	isCancelled()
boolean	isPaintable()
void	paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)
void	setAsText(String text)
void	setAvailable(AbstractComponent[] components)
void	setCategory(Category category) Sets the Category for this ComponentSelectionEditor.
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
void	setEditable(boolean e)
void	setFont(java.awt.Font font) 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

Methods

setAvailable

`public void **setAvailable**(AbstractComponent[] components)`

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:

TableCellEdtior value to be used to edit JTable cells;

com.cafean.client.ui.beans
Class IntNonMultiEdit

```
java.lang.Object
   ┣-java.awt.Component
      ┣-java.awt.Container
         ┣-javax.swing.JComponent
            ┣-javax.swing.JPanel
               ┣-com.cafean.client.ui.beans.IntNonMultiEdit
```

All Implemented Interfaces:
java.beans.PropertyEditor, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

---

public class **IntNonMultiEdit**
extends JPanel
implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.beans.PropertyEditor

This editor allows the user edit a integer that can't be multi-edited. This includes component numbers.

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	`IntNonMultiEdit()` Creates new form IntNonMultiEdit

Method Summary

String	`getAsText()`
java.awt.Component	`getCustomEditor()`
String	`getJavaInitializationString()`
String[]	`getTags()`

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, 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

IntNonMultiEdit

`public` **`IntNonMultiEdit`**`()`

Creates new form IntNonMultiEdit

Methods

getAsText

`public String` **`getAsText`**`()`

getCustomEditor

```
public java.awt.Component getCustomEditor()
```

isPaintable

```
public boolean isPaintable()
```

getJavaInitializationString

```
public String getJavaInitializationString()
```

getTags

```
public String[] getTags()
```

paintValue

```
public void paintValue(java.awt.Graphics gfx,
          java.awt.Rectangle box)
```

getValue

```
public Object getValue()
```

setAsText

```
public void setAsText(String text)
  throws IllegalArgumentException
```

setValue

```
public void setValue(Object value)
```

supportsCustomEditor

```
public boolean supportsCustomEditor()
```

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	`init(java.beans.PropertyDescriptor descriptor)` 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 IntSpinnerEditor

```
java.lang.Object
   |-java.awt.Component
      |-java.awt.Container
         |-javax.swing.JComponent
            |-javax.swing.JPanel
               |-com.cafean.client.ui.beans.IntSpinnerEditor
```

All Implemented Interfaces:
java.beans.PropertyEditor, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

---

public class **IntSpinnerEditor**
extends JPanel
implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.beans.PropertyEditor

This simple editor allows the user to select within a range of integer values. The range is defined either with the constructor or with setter methods.

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	`IntSpinnerEditor()` Creates new form IntSpinnerEditor
public	`IntSpinnerEditor(int minimum,int maximum,int increment)` This creates a new IntSpinnerEditor with the given values.
public	`IntSpinnerEditor(int minimum,int maximum,int increment,boolean showSize)` This creates a new IntSpinnerEditor with the given values.

Method Summary

String	`getAsText()`

java.awt.Component	getCustomEditor()
String	getJavaInitializationString()
String[]	getTags()
Object	getValue()
boolean	isPaintable()
void	paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)
void	setAsText(String text)
void	setIncrement(int increment) Sets the step size of the selection range.
void	setMaximum(int maximum) Sets the maximum value for the selection range.
void	setMinimum(int minimum) Sets the minimum value for the selection range.
void	setShowSize(boolean showSize) This flag hides or shows the label displaying the maximum allowable value for the spinner.
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, 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

IntSpinnerEditor

```
public IntSpinnerEditor()
```

Creates new form IntSpinnerEditor

IntSpinnerEditor

```
public IntSpinnerEditor(int minimum,
                        int maximum,
                        int increment)
```

This creates a new IntSpinnerEditor with the given values. Note that PropertyEditor instances are created using the default no-arg constructor. To use this constructor a sub-class must be created and used as the editor.

Parameters:

(continued from last page)

`minimum` - the minimum value for the selection.
`maximum` - the maximum value for the selectin.
`increment` - the step size for the selection.

IntSpinnerEditor

```
public IntSpinnerEditor(int minimum,
                        int maximum,
                        int increment,
                        boolean showSize)
```

This creates a new IntSpinnerEditor with the given values. Note that PropertyEditor instances are created using the default no-arg constructor. To use this constructor a sub-class must be created and used as the editor.

Parameters:

`minimum` - the minimum value for the selection.
`maximum` - the maximum value for the selectin.
`increment` - the step size for the selection.
`showSize` - sets whether the label displaying the current size is visible.

Methods

setMinimum

```
public void setMinimum(int minimum)
```

Sets the minimum value for the selection range.

---

setMaximum

```
public void setMaximum(int maximum)
```

Sets the maximum value for the selection range. This also updates the size label to reflect the maximum value. If the maximum is set to Integer.MAX_VALUE the size label's text is set empty.

---

setShowSize

```
public void setShowSize(boolean showSize)
```

This flag hides or shows the label displaying the maximum allowable value for the spinner.

---

setIncrement

```
public void setIncrement(int increment)
```

Sets the step size of the selection range.

---

getValue

```
public Object getValue()
```

---

setValue

```
public void setValue(Object value)
```

---

getAsText

```
public String getAsText()
```

(continued from last page)

setAsText

```
public void setAsText(String text)
  throws IllegalArgumentException
```

getJavaInitializationString

```
public String getJavaInitializationString()
```

getTags

```
public String[] getTags()
```

isPaintable

```
public boolean isPaintable()
```

paintValue

```
public void paintValue(java.awt.Graphics gfx,
         java.awt.Rectangle box)
```

getCustomEditor

```
public java.awt.Component getCustomEditor()
```

supportsCustomEditor

```
public boolean supportsCustomEditor()
```

com.cafean.client.ui.beans
Class LimitedLongStringEditor

```
java.lang.Object
   |
   +-java.awt.Component
      |
      +-java.awt.Container
         |
         +-javax.swing.JComponent
            |
            +-javax.swing.JPanel
               |
               +-com.cafean.client.ui.beans.PropertyEditorPanelSupport
                  |
                  +-com.cafean.client.ui.beans.LimitedLongStringEditor
```

public class **LimitedLongStringEditor**
extends PropertyEditorPanelSupport

This bean editor contains a JPanel with a label and a button. This allows the user to edit long strings in a text area, and displays the first 80 characters in the label.

This editor is considered limited in that it includes a maximum line length. Lines longer than this length will be wrapped.

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	`LimitedLongStringEditor(String title,int linelength)` Creates a new instance of LimitedLongStringEditor with the given title and line length Note that PropertyEditor instances are created using the default no-arg constructor.
public	`LimitedLongStringEditor()` Creates a new instance of LimitedLongStringEditor with no title and a line length of 80

Method Summary

void	`actionPerformed(java.awt.event.ActionEvent e)`
void	`updateLabel()`

Methods inherited from class `com.cafean.client.ui.beans.PropertyEditorPanelSupport`

getAsText, getCustomEditor, getJavaInitializationString, getTags, getValue, isPaintable, paintValue, setAsText, setForeground, setTextLabelString, setValue, supportsCustomEditor, updateLabel, updateValue

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

LimitedLongStringEditor

```
public LimitedLongStringEditor(String title,
                               int linelength)
```

Creates a new instance of LimitedLongStringEditor with the given title and line length Note that PropertyEditor instances are created using the default no-arg constructor. To use this constructor a sub-class must be created and used as the editor.

LimitedLongStringEditor

```
public LimitedLongStringEditor()
```

Creates a new instance of LimitedLongStringEditor with no title and a line length of 80

(continued from last page)

Methods

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent e)
```

---

updateLabel

```
public void updateLabel()
```

com.cafean.client.ui.beans
Class LongStringEditor

```
java.lang.Object
   |
   +-java.beans.PropertyEditorSupport
      |
      +-com.cafean.client.ui.beans.LongStringEditor
```

public class **LongStringEditor**
extends java.beans.PropertyEditorSupport

This bean editor contains a JPanel with a label and a button. This allows the user to edit long strings in a text area, and displays the first 80 characters in the label.

Constructor Summary

public	`LongStringEditor(String title)` Creates a new instance of LongStringEditor with the given title
public	`LongStringEditor()` Creates a new instance of LongStringEditor with no title

Method Summary

void	`addPropertyChangeListener(java.beans.PropertyChangeListener listener)`
java.awt.Component	`getCustomEditor()`
String	`getJavaInitializationString()`
String[]	`getTags()`
Object	`getValue()`
boolean	`isPaintable()`
void	`paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)`
void	`removePropertyChangeListener(java.beans.PropertyChangeListener listener)`
void	`setAsText(String text)`
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

LongStringEditor

`public` **`LongStringEditor`**`(String title)`

Creates a new instance of LongStringEditor with the given title

LongStringEditor

`public` **`LongStringEditor`**`()`

Creates a new instance of LongStringEditor with no title

Methods

addPropertyChangeListener

`public void` **`addPropertyChangeListener`**`(java.beans.PropertyChangeListener listener)`

getCustomEditor

`public java.awt.Component` **`getCustomEditor`**`()`

getJavaInitializationString

`public String` **`getJavaInitializationString`**`()`

getTags

`public String[]` **`getTags`**`()`

getValue

`public Object` **`getValue`**`()`

(continued from last page)

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)
```

setValue

```
public void setValue(Object value)
```

supportsCustomEditor

```
public boolean supportsCustomEditor()
```

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 Value: **-2147483647**
static final int	INACTIVE Value: **-2147483648**
static final int	UNKNOWN Value: **2147483647**

Constructor Summary

public	NamedIntEditor() Creates a new instance of 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.

Method Summary

`String`	`getAsText()`
`java.awt.Component`	`getCustomEditor()`
`String`	`getJavaInitializationString()` {@inheritDoc} Does nothing, and returns null.
`String[]`	`getTags()` {@inheritDoc} Does nothing, and returns null.
`Object`	`getValue()`
`boolean`	`isPaintable()`
`boolean`	`isShowNumbers()` 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)` Sets the owner and the description of the property currently being edited by this editor.
`void`	`setShowNumbers(boolean showNumbers)` 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

```
java.lang.Object
   |-java.awt.Component
      |-java.awt.Container
         |-javax.swing.JComponent
            |-javax.swing.JPanel
               |-com.cafean.client.ui.beans.BooleanEditor
                  |-com.cafean.client.ui.beans.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) 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()` Creates a new NamelistIntEditor

Method Summary

void	`itemStateChanged(java.awt.event.ItemEvent e)`
void	`setContext(PropertyController owner,java.beans.PropertyDescriptor descriptor)` 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) 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()` Creates a new NamelistIntEditor

Method Summary

void	`itemStateChanged(java.awt.event.ItemEvent e)`
void	`setContext(PropertyController owner,java.beans.PropertyDescriptor descriptor)` 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 *Active*state 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)` 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 *Active*state 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`	`NamelistRealEditor()` Creates a new instance of DefaultableRealEditor

Method Summary

`void`	`itemStateChanged(java.awt.event.ItemEvent e)`
`void`	`setContext(PropertyController owner,java.beans.PropertyDescriptor descriptor)` 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:
SnapPreferences, UserDefinedConstant, UserDefinedVariable, 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

static final int	ALL the group containing all propertied, be they disabled required or optional. Value: **7**
static final java.awt.Color	COLOR_OPTIONAL the foreground color for optional properties
static final int	DISABLED the group containing only disabled properties. Value: **4**
static final int	NONE the group containing no properties. Value: **0**
static final int	OPTIONAL the group containing only optional properties. Value: **2**
static final int	REQUIRED the group containing only required properties. Value: **1**

Method Summary

int	getAttributeIndex(String propertyName) Returns a relative index that can be used to order property lists.
boolean	isPropertyActive(String propertyName) Returns false if this object has a property with the given name that is considered inactive; otherwise true.

boolean	`isPropertyEnabled(String propertyName)` Returns false if this object has a property with the given name that has dependency code that fails;true otherwise
boolean	`isPropertyRequired(String propertyName)` Returns false if this object has a property with the given name that has requirement code that fails; true otherwise.
boolean	`isPropertyResizable(String propertyName)` Returns false if this object has an array property with the given name that should not normally be resizable.
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.
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.

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 propertied, 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 PropertyEditorPanelSupport

```
java.lang.Object
   ┗-java.awt.Component
      ┗-java.awt.Container
         ┗-javax.swing.JComponent
            ┗-javax.swing.JPanel
               ┗-com.cafean.client.ui.beans.PropertyEditorPanelSupport
```

All Implemented Interfaces:
java.awt.event.ActionListener, java.beans.PropertyEditor, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible

Direct Known Subclasses:
LimitedLongStringEditor

---

public abstract class **PropertyEditorPanelSupport**
extends JPanel
implements javax.accessibility.Accessible, java.io.Serializable, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.beans.PropertyEditor, java.awt.event.ActionListener

Abstract Bean property editor containing a JPanel with a label and a button.

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`	`PropertyEditorPanelSupport()` Creates new form PropertyEditorPanelSupport

Method Summary

`String`	`getAsText()` Gets the property value as a string suitable for presentation to a human to edit.
`java.awt.Component`	`getCustomEditor()` A PropertyEditor may chose to make available a full custom Component that edits its property value.

String	`getJavaInitializationString()` This method is intended for use when generating Java code to set the value of the property.
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 tag values.
Object	`getValue()` Gets the value of the property.
boolean	`isPaintable()` Determines whether the class will honor the painValue method.
void	`paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)` Paint a representation of the value into a given area of screen real estate.
void	`setAsText(String text)` Sets the property value by parsing a given String.
void	`setForeground(java.awt.Color fg)`
void	`setTextLabelString(String text)` Set the labels text.
void	`setValue(Object value)` Set (or change) the object that is to be edited.
boolean	`supportsCustomEditor()` Determines whether the propertyEditor can provide a custom editor.
abstract void	`updateLabel()` Override this to customize the label text.
void	`updateValue(Object value)` Non-overriden version of setValue().

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

PropertyEditorPanelSupport

`public` **`PropertyEditorPanelSupport`**`()`

Creates new form PropertyEditorPanelSupport

Methods

updateLabel

`public abstract void` **`updateLabel`**`()`

Override this to customize the label text.

(continued from last page)

setTextLabelString

```
public void setTextLabelString(String text)
```

Set the labels text.

Parameters:

`text` - The text to use.

---

setForeground

```
public void setForeground(java.awt.Color fg)
```

---

updateValue

```
public void updateValue(Object value)
```

Non-overriden version of setValue().

---

setValue

```
public void setValue(Object value)
```

Set (or change) the object that is to be edited.

Parameters:

`value` - The new target object to be edited. Note that this object should not be modified by the PropertyEditor, rather the PropertyEditor should create a new object to hold any modified value.

---

getValue

```
public Object getValue()
```

Gets the value of the property.

Returns:

The value of the property.

---

isPaintable

```
public boolean isPaintable()
```

Determines whether the class will honor the painValue method.

Returns:

True if the class will honor the paintValue method.

---

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. Note that the propertyEditor is responsible for doing its own clipping so that it fits into the given rectangle. If the PropertyEditor doesn't honor paint requests (see isPaintable) this method should be a silent noop.

Parameters:

`gfx` - Graphics object to paint into.

`box` - Rectangle within graphics object into which we should paint.

(continued from last page)

getJavaInitializationString

`public String **getJavaInitializationString**()`

This method is intended for use when generating Java code to set the value of the property. It should return a fragment of Java code that can be used to initialize a variable with the current property value. Example results are "2", "new Color(127,127,34)", "Color.orange", etc.

Returns:

A fragment of Java code representing an initializer for the current value.

---

getAsText

`public String **getAsText**()`

Gets the property value as a string suitable for presentation to a human to edit.

Returns:

The property value as a string suitable for presentation to a human to edit.

Returns "null" is the value can't be expressed as a string.

If a non-null value is returned, then the PropertyEditor should be prepared to parse that string back in setAsText().

---

setAsText

```
public void setAsText(String text)
  throws IllegalArgumentException
```

Sets the property value by parsing a given String. May raise java.lang.IllegalArgumentException if either the String is badly formatted or if this kind of property can't be expressed as text.

Parameters:

`text` - The string to be parsed.

---

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 tag values. This can be used to represent (for example) enum values. If a PropertyEditor supports tags, then it should support the use of setAsText with a tag value as a way of setting the value.

Returns:

The tag values for this property. May be null if this property cannot be represented as a tagged value.

---

getCustomEditor

`public java.awt.Component **getCustomEditor**()`

A PropertyEditor may chose to make available a full custom Component that edits its property value. It is the responsibility of the PropertyEditor to hook itself up to its editor Component itself and to report property value changes by firing a PropertyChange event. The higher-level code that calls getCustomEditor may either embed the Component in some larger property sheet, or it may put it in its own individual dialog, or ...

Returns:

A java.awt.Component that will allow a human to directly edit the current property value. May be null if this is not supported.

---

supportsCustomEditor

`public boolean **supportsCustomEditor**()`

Determines whether the propertyEditor can provide a custom editor.

(continued from last page)

Returns:

True if the propertyEditor can provide a custom editor.

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 lanuches a `RealArrayDialog` that actually performs the editing.

Constructor Summary

public	`RealArrayEditor(boolean fixedDimension,int dimension)` This creates a new instance of RealArrayEditor that has a fixed dimension
public	`RealArrayEditor()` 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()` Does nothing, and returns null.
AbstractModel	`getModel()`
String[]	`getTags()` Does nothing, and returns null.
Object	`getValue()`
void	`init(java.beans.PropertyDescriptor descriptor)`
boolean	`isPaintable()`

void	paintValue(java.awt.Graphics gfx,java.awt.Rectangle box)
void	removePropertyChangeListener(java.beans.PropertyChangeListener listener)
void	setAsText(String text)
void	setDimension(int dimension) Sets the dimension on a given array.
void	setModel(AbstractModel model)
void	setValue(Object value) If this array is of fixed width.
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

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()` Creates new form RealBeanEditor

Method Summary

`void`	`firePropertyChange()` Report that we have been modified to any interested listeners.
`String`	`getAsText()`
`java.awt.Component`	`getCustomEditor()`

String	getJavaInitializationString()
AbstractModel	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, 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

(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)` Creates a new action for the given toolbox.

Method Summary

abstract JComponent	`createComponent()` 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	`BeanAction(Toolbox toolbox,Class beanClass)` Creates a new action for creating the given bean.

Method Summary

JComponent	`createComponent()` 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	CategoryAction(Toolbox toolbox,Category category)

Method Summary

Category	getCategory()

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
   |-com.cafean.client.ui.tools.MouseHandler
      |-com.cafean.client.ui.tools.ConnectMouseHandler
```

public class **ConnectMouseHandler**
extends MouseHandler

A MouseHandler for Connection events on the ZoomablePanel

Field Summary

static final java.awt.Cursor	CURSOR_CONNECT The cursor for the connect tool
static final java.awt.Cursor	CURSOR_CONNECT_COMPLETE The cursor for the connect tool when connection completion is possible
static final java.awt.Cursor	CURSOR_CONNECT_OFF The cursor for the connect tool when no connection is possible

Constructor Summary

public	ConnectMouseHandler(ZoomablePanel parent) Creates a new mouse handler with the given parent

Method Summary

void	activate() {@inheritDoc}
void	completeConnection(DrawnComponent targetDrawing,ConnectingPt point) Completes the current connection.
void	deactivate() {@inheritDoc}
ImageIcon	getButtonIcon() Returns the icon for use in creating an activation toolbar button for this handler.
AbstractComponent	getConnectionSource() Returns the AbstractComponent that started the connection.
java.awt.Cursor	getCurrentCursor(java.awt.event.MouseEvent evt) Retrieves the cursor for the views based off of the currently selected tool.

java.awt.Cursor	getCursor(java.awt.event.MouseEvent evt) Retrieves the cursor for this MouseHandler within the context of the given MouseEvent
boolean	isInProgress() Returns true if a connection draw is in progress.
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.
void	mouseDragged(java.awt.event.MouseEvent orig_evt)
void	mouseMoved(java.awt.event.MouseEvent orig_evt)
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.
void	mouseReleased(java.awt.event.MouseEvent orig_evt)
boolean	requiresTarget() 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() 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 recieve 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
   |-com.cafean.client.ui.tools.MouseHandler
      |-com.cafean.client.ui.tools.InsertMouseHandler
```

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)` Creates a new mouse handler with the given parent

Method Summary

void	`cancelInsert()` Cancels the current insertion
void	`finishInsert(java.awt.event.MouseEvent orig_evt)` Completes the current insertion by adding the component and initializing it.
java.awt.Cursor	`getCurrentCursor(java.awt.event.MouseEvent evt)` Retrieves the cursor for the views based off of the currently selected tool.
void	`mouseClicked(java.awt.event.MouseEvent orig_evt)` Handles mouse clicked events by inserting a new component.
void	`mouseDragged(java.awt.event.MouseEvent orig_evt)` Forwards mouseDragged events to the child AbstractInsertHandler, if one is currently in use.
void	`mouseMoved(java.awt.event.MouseEvent orig_evt)` Forwards mouseMoved events to the child AbstractInsertHandler, if one is currently in use.
void	`mousePressed(java.awt.event.MouseEvent orig_evt)` the mouse pressed event is not handled by this tool but is instead forwarded.
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`.

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
   |-com.cafean.client.ui.tools.MouseHandler
      |-com.cafean.client.ui.tools.InteractiveMouseHandler
```

public class **InteractiveMouseHandler**
extends MouseHandler

A MouseHandler for interactive handling of events on the ZoomablePanel

Constructor Summary

public	`InteractiveMouseHandler(ZoomablePanel parent)` Creates a new mouse handler with the given parent

Method Summary

void	`activate()`
ImageIcon	`getButtonIcon()` 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()` 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 achive the desired mouse feedback.

See Also:
com.cafean.client.ui.ZoomablePanel.addMouseHandler(), com.cafean.client.ui.ZoomablePanel.removeMouseHandler()

Constructor Summary

public	`MouseHandler(ZoomablePanel parent)` Creates a new instance of MouseHandler

Method Summary

void	`activate()` Sets up this MouseHandler to be ready to recieve MouseEvents
void	`deactivate()` Deactivates this MouseHandler and cleans up any current operations.
ImageIcon	`getButtonIcon()` Returns the icon for use in creating an activation toolbar button for this handler.

java.awt.Cursor	`getCurrentCursor(java.awt.event.MouseEvent evt)` Retrieves the cursor for the views based off of the currently selected tool.
java.awt.Cursor	`getCursor(java.awt.event.MouseEvent evt)` Retrieves the cursor for this MouseHandler within the context of the given MouseEvent.
int	`getHandlerID()` Gets this handler's ID.
String	`getTooltipText()` returns the tooptip text to use for this MouseHandler's toggle button or null of no text is desired.
void	`mouseClicked(java.awt.event.MouseEvent orig_evt)` Forward the mouseClicked event to the underlying child container.
void	`mouseEntered(java.awt.event.MouseEvent orig_evt)` Forward the mouseEntered event to the underlying child container.
void	`mouseExited(java.awt.event.MouseEvent orig_evt)` Forward the mouseExited event to the underlying child container.
void	`mouseMoved(java.awt.event.MouseEvent orig_evt)` Forward the mouseMoved event to the underlying child container.
void	`setHandlerID(int id)` Sets this handler's ID.
String	`toString()` 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 recieve 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 tooptip text to use for this MouseHandler's toggle button or null of 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 java.awt.Cursor	CURSOR_PAN The cursor for the pan tool, when a mouse button is not being pressed
static final java.awt.Cursor	CURSOR_PAN_GRIP The cursor for the pan tool, while a mouse button is being pressed

Constructor Summary

public	PanMouseHandler(ZoomablePanel parent) Creates a new mouse handler with the given parent

Method Summary

ImageIcon	getButtonIcon() Returns the icon for use in creating an activation toolbar button for this handler.
java.awt.Cursor	getCurrentCursor(java.awt.event.MouseEvent evt) 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) Handle mouseDragged events for move and resize operations.
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.
void	mouseReleased(java.awt.event.MouseEvent orig_evt)
String	toString() 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)` Creates a new mouse handler with the given parent

Method Summary

static boolean	`canBeResized(java.awt.Component component)` Returns true if the given component can be resized by this handler.
void	`deactivate()`
ImageIcon	`getButtonIcon()` Returns the icon for use in creating an activation toolbar button for this handler.
java.awt.Cursor	`getCurrentCursor(java.awt.event.MouseEvent evt)` 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)` Handle mouseDragged events for move and resize operations.
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.
void	`mouseReleased(java.awt.event.MouseEvent orig_evt)` Handle mouseReleased events to support move, resize, select operations.
String	`toString()` 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	`ToolboxAction(Toolbox toolbox,String name,Icon icon,String description)` Creates a new ToolboxAction with the given toolbox.

Method Summary

void	`actionPerformed(java.awt.event.ActionEvent e)` Responds to ActionEvents by setting this as the current ToolboxAction in this action's Toolbox.
String	`getDescription()` Retrieves this action's description.
Icon	`getIcon()` Retrieves this action's icon.
String	`getName()` 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 java.awt.Cursor	CURSOR_ZOOM_IN The cursor for the Zoom Tool
static final java.awt.Cursor	CURSOR_ZOOM_OUT The cursor for the Zoom Tool

Constructor Summary

public	ZoomMouseHandler(ZoomablePanel parent) Creates a new mouse handler with the given parent

Method Summary

void	activate()
void	deactivate()
ImageIcon	getButtonIcon() Returns the icon for use in creating an activation toolbar button for this handler.
java.awt.Cursor	getCurrentCursor(java.awt.event.MouseEvent evt) Retrieves the cursor for the views based off of the currently selected tool.
void	mouseDragged(java.awt.event.MouseEvent evt) Handle mouseDragged events for move and resize operations.
void	mousePressed(java.awt.event.MouseEvent orig_evt)
void	mouseReleased(java.awt.event.MouseEvent orig_evt) Handle mouseReleased events to support move, resize, select operations.

String	toString() 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)` Creates a new instance of MouseHandler

Method Summary

void	`cancelInsert()` Cancels the insertion by calling `InsertMouseHandler#finishInsert`.
void	`finishInsert(java.awt.event.MouseEvent orig_evt)` Completes the insertion by calling `InsertMouseHandler#finishInsert`.
java.awt.Cursor	`getCurrentCursor()` Retrieves the cursor for the views based off of the currently selected tool.
void	`mouseClicked(java.awt.event.MouseEvent orig_evt)` This handler ignores mouse clicked events
void	`mouseEntered(java.awt.event.MouseEvent orig_evt)` sets the zoomable panel's cursor to the handler's current cursor
void	`mouseExited(java.awt.event.MouseEvent orig_evt)` sets the zoomable panel's cursor to the handler's current cursor
void	`mouseMoved(java.awt.event.MouseEvent orig_evt)` 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 The number of pixels the mouse is required to move before another point can be added. Value: **5**

Constructor Summary

public	AbstractPathHandler(ZoomablePanel parent) Creates a new mouse handler with the given parent

Method Summary

void	mouseClicked(java.awt.event.MouseEvent orig_evt) This insert handler ignores mouseClicked eventss
void	mouseDragged(java.awt.event.MouseEvent orig_evt) Handles mouse moved events by repainting the current rubber bands.
void	mouseMoved(java.awt.event.MouseEvent orig_evt) Handles mouse dragged events by repainting the current rubber bands.
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.
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).

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

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.

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 of none is appropriate

com.cafean.client.ui.tools.insert
Class RectangularInsertHandler

```
java.lang.Object
   |-com.cafean.client.ui.tools.insert.AbstractInsertHandler
      |-com.cafean.client.ui.tools.insert.RectangularInsertHandler
```

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)` Creates a new instance of RectangularInsertHandler

Method Summary

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.
void	`mouseDragged(java.awt.event.MouseEvent orig_evt)` Continues the process of drawing the rubber band rectangle for bounds selection.
void	`mousePressed(java.awt.event.MouseEvent orig_evt)` Begins the process of drawing the rubber band rectangle for bounds selection.
void	`mouseReleased(java.awt.event.MouseEvent orig_evt)` 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)` Creates a new MECodePlugin with the given name.

Method Summary

abstract AbstractModel	`createNewModel(boolean initNew)` Creates a new instance of the AbstractModel defined by this plugin.
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.
StyledDocument	`getAsciiStyledDocument(Writeable writeable)` Retrieves a StyledDocument suitable for displaying the written ASCII of the given Writeable component.
String	`getLabel()` Retrieves a short string that describes a singular unit of this plugin.
String	`getLabelPlural()` Retrieves a short string that describes multiple units of this plugin.
ImageIcon	`getPluginIcon()` Retrieves the icon associated with this plugin.
int	`getPluginType()`
abstract String	`getSamPackage()` This returns the package name of the SAM files for this plugin.
boolean	`isAnimatable()` Returns true if this MECodePlugin includes support for View animation.
boolean	`isBeanBased()` Returns true if this MECodePlugin's design is based on the JavaBeans architecture.
abstract AbstractModel	`open(java.io.File file)` Loads a new AbstractModel from the given local File.
abstract void	`submitModel(AbstractModel model,boolean local)` 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) Creates a new instance of MEFeaturePlugin

Method Summary

int	getPluginType()
abstract boolean	isAssociated(AbstractModel model) This returns 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.apt.xdr.PibFile pibfile) This is called on all Feature plugins when a model has been loaeded.
abstract void	modelAdded(AbstractModel model) This is called on all Feature plugins when a new model has been added to the model editor.
abstract void	modelRemoved(AbstractModel model)
abstract void	save(AbstractModel savedModel,com.apt.xdr.PibFile pibfile) 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 loaeded. 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 initalizes any data they need.

Parameters:

`model` - the AbstractModel just added to the model Editor.

isAssociated

`public abstract boolean` **`isAssociated`**`(AbstractModel model)`

This returns 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` the typecode for analysis code support plugins Value: **1**
static final int	`TYPE_FEATURE_PLUGIN` the typecode for feature addition plugins Value: **2**

Constructor Summary

public	`MEPlugin(String name)` Creates a new MEPlugin with the given name.

Method Summary

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}.
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}.
String	`getName()` This returns the descriptive name of this plugin.
abstract String	`getPluginId()` This returns the static plugin id.
abstract String	`getPluginInfo()` Returns the information about this plugin that will appear in the Plugins dialog.

Object	`getPluginPreferences()` Retrieves an Object that follows the Java Beans design paradigm that contains the preferences for this plugin.
abstract String[]	`getPluginPrereqs()` Returns the names and versions of the plugins this plugin depends upon.
abstract int	`getPluginType()` This returns the static type flag that indicates what type of plugin this is.
abstract String	`getVersion()` Returns the version number of this plugin as a String.
abstract void	`loadMainMenuItems()` Loads this plugin's menu items into the `MainFrame` menu bar.
abstract void	`loadSettings(Configurator config)` Loads user settings for this plugin from the configuration file.
abstract void	`loadViewMenuItems(DrawnView view)` Loads this plugin's menu items into a `DrawnView`'s Tools menu.
abstract void	`processCommand(Vector command)` Processes the given batch command.
void	`setName(String name)` This sets the descriptive name of this plugin.
abstract void	`storeSettings(Configurator config)` 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 JMenuItems 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 JMenuItems 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(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(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 prerequesite 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()` Creates a new instance of MEPluginData

Method Summary

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.
abstract String	`getPluginClass()` Returns the absolute path to the primary class for the plugin in string form.
abstract String	`getPluginId()` Returns this plugin's plugin-id.
abstract String[]	`getPluginPrereqs()` Returns the names and versions of the plugins this plugin depends upon.
abstract String	`getVersion()` Returns the current plugin version.
abstract MEPlugin	`loadPlugin()` 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 staticly 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: **1.5707963267948966**

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() {@inheritDoc}
double	getDisplayValue(int unittypes)
String	getENG_Units() {@inheritDoc}
java.text.DecimalFormat	getFormat(double value) {@inheritDoc}
String	getSI_Units() {@inheritDoc}
boolean	isMostlyVerticalDown() Return true if the angle is < -45 degress.
boolean	isMostlyVerticalUp() Return true if the angle is > 45 degress.
boolean	isVerticalDown() Return true if the angle is close to -180 degress.
boolean	isVerticalUp() Return true if the angle is close to 180 degress.

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 degress.

isVerticalDown

`public boolean isVerticalDown()`

Return true if the angle is close to -180 degress.

isMostlyVerticalDown

`public boolean isMostlyVerticalDown()`

Return true if the angle is < -45 degress.

isMostlyVerticalUp

`public boolean isMostlyVerticalUp()`

Return true if the angle is > 45 degress.

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 The enumeration value for British units. Value: **1**
static final int	SI The enumeration value for SI units. Value: **0**
static final String[]	UNIT_NAMES A list of supported unit Types and their labels.

Constructor Summary

public	BaseNumber() Creates a new instance of a BaseNumber.

Method Summary

Object	clone() {@inheritDoc}
static boolean	compareDouble(double a,double b,double eps) Compares two doubles to see if they are within epsilon of each other.
abstract void	convert(Number n) convert set the object to a value using the current unit type conversion.

abstract String	getENG_Units() Return the string used to represent BRITISH units.
abstract java.text.DecimalFormat	getFormat(double value) Return the format used to represent a value.
abstract String	getSI_Units() Return the string used to represent SI units.
String	getUnits() Retrieves a string representation of the current units.
static int	getUnitType() Retrieves the global default unit type
boolean	isKnown() Return true if the value if known, otherwise return false.
boolean	isUnknown()
void	setKnown() Set the unknown flag to false.
static void	setUnitType(int t) Sets the global default unit type
void	setUnknown() Set the unknown flag to true.
abstract void	setValue(BaseNumber n) setValue sets the object to a value from another BaseNumber.
abstract void	setValue(Number n) setValue sets the object to a value without unit type conversion.
abstract String	toString() 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()`

Retrieves the global default unit type

Returns:

SI or BRITISH.

setUnitType

`public static void setUnitType(int t)`

Sets the global default 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 if known, otherwise return false.

Returns:

true if the value if 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() Creates a new instance with an unknown value.
public	Dimless(double d) Creates a new instance initialized to a value.

Method Summary

double	getConversionFactor() {@inheritDoc}
String	getENG_Units() {@inheritDoc}
String	getSI_Units() {@inheritDoc}
static Dimless[]	makeArray(double[] dArray) Create an array of Dimless objects from an array of doubles.
static Dimless[]	makeArray(Vector vector) 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, 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 dynamicaly sized array of double.

Constructor Summary

public	`DoubleArray(int capacity)` Creates a new DoubleArray with the given initial capacity.
public	`DoubleArray(double[] array)`

Method Summary

void	`add(double value)` Appends the given double to the DoubleArray's data.
void	`add(double[] values)` Appends the given data to the DoubleArray's data.
int	`capacity()` Returns this DoubleArray's current capacity.
void	`clear()` Sets this DoubleArray empty.
Object	`clone()`
boolean	`contains(double value)` Returns true if the given value is contained in this DoubleArray.
double	`get(int index)` Returns the value of the double at the given index.
int	`indexOf(double value)` Retrieves the index of the first occurance of the given value in this DoubleArray.

static void	`main(String[] args)` Performs a thorough test of the DoubleArray's functionality.
void	`remove(int index)` Removes the double at the given index. NOTE: O(n) complexity.
void	`set(int index,double value)` Sets the given value at the given index.
int	`size()` Returns the number of ints stored in this DoubleArray
void	`sort()` Sorts this DoubleArray via `java.util.Arrays.sort()`.
double[]	`toArray()` Returns an double array containing only the used elements of this DoubleArray.
void	`trimToSize()` 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 occurance 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 dynamicaly sized array of float.

Constructor Summary

public	`FloatArray(int capacity)` Creates a new FloatArray with the given initial capacity.
public	`FloatArray(float[] array)`

Method Summary

void	`add(float value)` Appends the given float to the FloatArray's data.
void	`add(float[] values)` Appends the given data to the FloatArray's data.
int	`capacity()` Returns this FloatArray's current capacity.
void	`clear()` Sets this FloatArray empty.
Object	`clone()`
boolean	`contains(float value)` Returns true if the given value is contained in this FloatArray.
float	`get(int index)` Returns the value of the float at the given index.
int	`indexOf(float value)` Retrieves the index of the first occurance of the given value in this FloatArray.

static void	main(String[] args) Performs a thorough test of the FloatArray's functionality.
void	remove(int index) Removes the float at the given index. NOTE: O(n) complexity.
void	set(int index,float value) Sets the given value at the given index.
int	size() Returns the number of ints stored in this FloatArray
void	sort() Sorts this FloatArray via java.util.Arrays.sort().
float[]	toArray() Returns an float array containing only the used elements of this FloatArray.
void	trimToSize() 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 occurance 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
   |-com.cafean.Number.BaseNumber
      |-com.cafean.Number.Int
```

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 Value: **-99119900**
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

Int	add(int x)
Int	add(Int aInt)
int	compareTo(Object o)
void	convert(Long n)

void	convert(Number n)
void	convert(String s)
Int	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)
Int	multiply(Int aInt)
void	setValue(BaseNumber a)
void	setValue(int aInt)
void	setValue(Number n)
void	setValue(String aString)
Int	sqrt()
Int	subtract(int x)

Int	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 dynamicaly sized array of int.

Constructor Summary

public	`IntegerArray(int capacity)` Creates a new IntegerArray with the given initial capacity.
public	`IntegerArray(int[] array)`

Method Summary

void	`add(int value)` Appends the given int to the IntegerArray's data.
void	`add(int[] values)` Appends the given data to the IntegerArray's data.
int	`capacity()` Returns this IntegerArray's current capacity.
void	`clear()` Sets this IntegerArray empty.
Object	`clone()`
boolean	`contains(int value)` Returns true if the given value is contained in this IntegerArray.
int	`get(int index)` Returns the value of the int at the given index.
int	`indexOf(int value)` Retrieves the index of the first occurance of the given value in this IntegerArray.

static void	`main(String[] args)` Performs a thorough test of the IntegerArray's functionality.
void	`remove(int index)` Removes the int at the given index. NOTE: O(n) complexity.
void	`set(int index,int value)` Sets the given value at the given index.
int	`size()` Returns the number of ints stored in this IntegerArray
void	`sort()` Sorts this IntegerArray via `java.util.Arrays.sort()`.
int[]	`toArray()` Returns an int array containing only the used elements of this IntegerArray.
void	`trimToSize()` 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 occurance 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 dynamicaly sized array of long.

Constructor Summary

public	`LongArray(int capacity)` Creates a new LongArray with the given initial capacity.
public	`LongArray(long[] array)`

Method Summary

void	`add(long value)` Appends the given long to the LongArray's data.
void	`add(long[] values)` Appends the given data to the LongArray's data.
int	`capacity()` Returns this LongArray's current capacity.
void	`clear()` Sets this LongArray empty.
Object	`clone()`
boolean	`contains(long value)` Returns true if the given value is contained in this LongArray.
long	`get(int index)` Returns the value of the long at the given index.
int	`indexOf(long value)` Retrieves the index of the first occurance of the given value in this LongArray.

static void	main(String[] args) Performs a thorough test of the LongArray's functionality.
void	remove(int index) Removes the long at the given index. NOTE: O(n) complexity.
void	set(int index,long value) Sets the given value at the given index.
int	size() Returns the number of ints stored in this LongArray
void	sort() Sorts this LongArray via java.util.Arrays.sort().
long[]	toArray() Returns an long array containing only the used elements of this LongArray.
void	trimToSize() 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 occurance 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

```
java.lang.Object
   |-com.cafean.Number.BaseNumber
      |-com.cafean.Number.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 Value: **-1.0E30**

Fields inherited from class com.cafean.Number.BaseNumber
BRITISH, SI, UNIT_NAMES

Constructor Summary

public	Real() Creates a new instance of Real with an unknown value.
public	Real(double aDouble) Creates a new instance of Real initialized to a value.
public	Real(Real aReal) Creates a new instance of Real by copying another.

Method Summary

Real	abs()
Real	add(double aDouble)
Real	add(Real aReal)

static boolean	arraysEqual(Real[] a,Real[] b)
Object	clone()
int	compareTo(Object o)
void	convert(double d) Converts the double given into the global units, and sets the result as the value of this Real.
void	convert(double d,int unitType) Converts the double given into the units requested, and sets the result as the value for this Real.
void	convert(Number n) Converts the doubleValue of the Number given to the global units.
void	convert(String aString) Converts the String given into the global units, and sets the result as the value for this Real.
void	convert(String aString,int unitType) Converts the String given into the units requested, and sets the result as the value for this Real.
Real	divide(double aDouble)
Real	divide(Real aReal)
Real	divideby(double aDouble)
Real	divideby(Real aReal)
boolean	equals(double aDouble)
static boolean	equals(double a,double b) Returns true if the given doubles differ by less than 5E-9.
static boolean	equals(double a,double b,double delta) Returns true if the given doubles differ by less than the given (a + b) * delta.
boolean	equals(Object o)
boolean	equals(Real aReal) NOTE: This does not get called instead of Object.equals
double	getConversionFactor() Retrieves the conversion factor used to convert this Real's value from SI to British units.
double	getCurrentDisplayValue(int unittypes) Returns the value to display.
String	getDisplayName() Retrieves a Display name for this unit.

double	getDoubleValue() Returns the numerical value of this Real in SI units.
String	getENG_Units() Retrieves a String representation of this Real's English unit type.
java.text.DecimalFormat	getFormat(double value)
String	getHtmlUnits() Returns the current global units formated for HTML
String	getHtmlUnits(int units) Returns the units requested formated for HTML.
java.text.DecimalFormat	getLoadFormat(double value) Decides what DecimalFormat to use to display the given value in 12 characters.
String	getName() Returns the name of this Real.
static double	getReferencedValue(double value)
String	getSI_Units() Retrieves a String representation of this Real's SI unit type.
int	getSignificantFigs() Getter for property significantFigs.
static double	getStrippedValue(double value)
Object	getTransferData(java.awt.datatransfer.DataFlavor flavor) Returns an object which represents the data to be transferred.
java.awt.datatransfer.DataFlavor[]	getTransferDataFlavors() Returns an array of DataFlavor objects indicating the flavors the data can be provided in.
String	getUnitName() Returns the name of this Real value
String	getUnits(int units) Returns this real's units in the given format.
double	getValue() 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) Returns whether or not the specified data flavor is supported for this object.

boolean	isKnown() Return true if the value if known, otherwise return false.
boolean	isUnknown() 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) Locks Real's toString() method for use only by the given thread.
Real	multiply(double aDouble)
Real	multiply(Real aReal)
static void	normalizeArray(Real[] array)
void	restoreState(String prefix,Hashtable state) Restore the state of the Real from an earlier edit.
void	setSignificantFigs(int significantFigs) Setter for property significantFigs.
void	setUnknown()
void	setValue(BaseNumber a)
void	setValue(double aDouble) Sets the current value of this Real.
void	setValue(Number n)
void	setValue(UserDefinedValue input) Sets the current value to a reference to a given UserDefinedValue.
static void	sortParallelRealArrays(Real[][] arrays) Sorts the given parallel arrays by the values in the first array.
Real	sqrt()
void	storeState(String prefix,Hashtable state) Store the state of the Real to permit undo.
Real	subtract(double aDouble)
Real	subtract(Real aReal)

String	`toLoadString()` Returns a formated string for this Real in the current global units that is limited to 12 characters in length.
String	`toLoadString(int unitType)` Returns a formated string for this Real in the requested units that is limited to 12 characters in length.
String	`toString()`
String	`toString(int unitType)` Returns a formated string containing the current value of this Real.
static void	`unlockToString()` 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 formated for HTML

getHtmlUnits

`public String getHtmlUnits(int units)`

Returns the units requested formated 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 mathmatical 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 formated 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 formated 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 formated 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 formated 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 ) * 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 if known, otherwise return false.

Returns:

true if the value if 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 Time[]	makeArray(double[] dArray)
static Time[]	makeArray(Vector vector)

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

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()` Creates a new ValueOutOfRangeException with no message
public	`ValueOutOfRangeException(String message)` Creates a new ValueOutOfRangeException with the given message
public	`ValueOutOfRangeException(String name,int minVal,int maxVal,int val)` Creates a new ValueOutOfRangeException with the given values.
public	`ValueOutOfRangeException(String name,double minVal,double maxVal,double val)` Creates a new ValueOutOfRangeException with the given values.
public	`ValueOutOfRangeException(String s,Real minVal,Real maxVal,Real val)` 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.

Package
com.cafean.utils

com.cafean.utils
Class Configurator

```
java.lang.Object
   |
   +-com.cafean.utils.Configurator
```

Direct Known Subclasses:
SnapPreferences

---

public class **Configurator**
extends Object

A utility class for maintaining a set of properties sorted by module and stored in an XML file in the current user's home directory.

Settings are organized by Module, which indicates which application a particular setting is for. This allows for different packages to use the same setting name without overwritting each other's data.

The most commonly used methods are `#save`, `#getGeometry(String,Window)`, and `#setGeometry(String,Window)`.

Constructor Summary

public	`Configurator()` Creates a instance of Configurator and loads in configuration data from either $HOME/.snaprc or a blank copy.
public	`Configurator(String module)` Creates a instance of Configurator and loads in configuration data from either $HOME/.snaprc or a blank copy.

Method Summary

boolean	`addCodeInfos(CodeInfo s)` Adds a CodeInfo server to the ExecutionMonitor module list.
boolean	`addServer(Server s)` Adds a calculation server to the global module.
void	`destroy()` Cleans up all object references inside Configurator.
Vector	`getCodeInfos()` Retrieves the list of codeinfo's from the ExecutionMonitor module.
java.awt.Color	`getColor(String name)` Retrieves the color stored with the given name from the current module.
java.awt.Color	`getColor(String name,java.awt.Color def)` Retrieves the color stored with the given name from the current module.

String	`getConfigFileDir()` Gets the full path from to the directory that contains the .snaprc file.
String	`getConfigFileName()` Retrieves the name and location of this Configurator's config file.
void	`getDividerLocation(String name,JSplitPane pane,int location)` Loads the divider location stored under the given keyword in the current module to the `JSplitPane` provided.
java.awt.Font	`getFont(String name)` Retreives the `font` stored with the given name inside the current module.
java.awt.Font	`getFont(String name,String family,String size,String style)` Retrieves a named Font from the current module.
void	`getGeometry(String windowName,javax.help.HelpBroker broker)` Loads the geometry for a specified help window in the current module from the SNAP Config file into the `HelpBroker` provided.
void	`getGeometry(String windowName,javax.help.HelpBroker broker,int height,int width)` Loads the geometry for a specified help window in the current module from the SNAP Config file into the `HelpBroker` provided.
void	`getGeometry(String windowName,java.awt.Window frame)` Loads the geometry for a specified help window in the current module from the SNAP Config file into the `frame` provided.
void	`getGeometry(String windowName,java.awt.Window frame,int height,int width)` Loads the geometry for a specified help window in the current module from the SNAP Config file into the `frame` provided.
void	`getGeometry(String windowName,java.awt.Window frame,int height,int width,int x,int y)` Loads the geometry for a specified help window in the current module from the SNAP Config file into the `frame` provided.
Server	`getNrcDBServer()` Retrieves the NRC Databank server information from the Global module.
String	`getProperty(String propertyName)` Retrieves a named property from the current module.
String	`getProperty(String name,String def)` Retrieves a named property from the current module.
LinkedList	`getServers()` Retrieves the list of servers from the Global module.
LinkedList	`getServers(SnapCorba snapCorba)` Retrieves the list of servers from the Global list and adds those to the list of servers retrieved from the given corba interface.

Server	`getSnapDBServer()` Retrieves the database server information from the Global module.
WebSupport	`getWebSupport()` Retrieves the Web support configuration from the Global module.
static boolean	`reDirectStdErrOut(String prefix)` Redirects stderr & stdout to a file in the users $HOME/.snap/ directory named `[prefix].screen`.
void	`removeAllCodeInfos()` Removes all CodeInfo from the ExecutionMonitor module list in preperation for re-creating the list from a modified copy.
void	`removeAllServers()` Removes all calculation server entitys from the global list preperation for re-creating the list from a modified copy.
boolean	`removeServer(String name)` Removes a calculation server from the Global module.
boolean	`save()` Saves the currently loaded configuration to the user's .snaprc.
void	`setColor(String name,java.awt.Color color)` Stores the given `Color` with the the given name inside the current module.
void	`setDBServer(Server s)` Sets the given SnapDB Server info into the Global module.
void	`setDividerLocation(String keyword,JSplitPane pane)` Saves the divider location for the given `JSplitPane` under the given keyword in the current module to the SNAP Config file.
void	`setFont(String name,java.awt.Font font)` Stores the given `font` with the the given name inside the current module.
void	`setGeometry(String windowName,javax.help.HelpBroker broker)` Saves the geometry for a specified help window in the current module to the SNAP Config file from the `HelpBroker` provided.
void	`setGeometry(String windowName,java.awt.Window frame)` Saves the geometry for a specified window in the current module to the SNAP Config file from the `frame` provided.
void	`setLookAndFeel(String lafStr)` Set the Look-and-feel for the user interface to Windows, Motif, Mac, or the Java Look and Feel.
void	`setModule(String moduleName)` Sets the Configurator's current module to the one provided.
void	`setNrcDBServer(Server s)` Sets the given NRC Databank Server info into the Global module.

void	`setProperty(String name,String propertyValue)` Assigns a property to be saved under the given name inside the current module.
void	`setWebSupport(WebSupport s)` Sets the given Web support configuration into the Global module.
int	`updateAvailable(String application)` Retrieves an XML document from the SNAP web server and compares version information contained in the document with current version information.

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Configurator

`public` **`Configurator`**`()`

Creates a instance of Configurator and loads in configuration data from either $HOME/.snaprc or a blank copy. Also sets the current module to "Global".

Configurator

`public` **`Configurator`**`(String module)`

Creates a instance of Configurator and loads in configuration data from either $HOME/.snaprc or a blank copy. This sets the current module to the given value.

Parameters:

`module` - a String containing the module to set initially.

Methods

getConfigFileName

`public String` **`getConfigFileName`**`()`

Retrieves the name and location of this Configurator's config file.

getConfigFileDir

`public String` **`getConfigFileDir`**`()`

Gets the full path from to the directory that contains the .snaprc file.

Returns:

the full path to the .snap directory.

save

`public boolean` **`save`**`()`

Saves the currently loaded configuration to the user's .snaprc. Only modules that have data that changed are saved.

Returns:

(continued from last page)

the boolean success or failure of the save.

destroy

```
public void destroy()
```

Cleans up all object references inside Configurator. This is done to clear up the memory used by the Configurator.

setModule

```
public void setModule(String moduleName)
```

Sets the Configurator's current module to the one provided. If this module isn't found in the configuration file, a blank one is created.

Parameters:

`moduleName` - the String containing the name of the desired module.

setGeometry

```
public void setGeometry(String windowName,
          javax.help.HelpBroker broker)
```

Saves the geometry for a specified help window in the current module to the SNAP Config file from the `HelpBroker` provided.

Parameters:

`windowName` - the String that contains the keyword to save the data under. For example: Veda's main window's keyword is "main."
`broker` - the HelpBroker to save geometry from.

setGeometry

```
public void setGeometry(String windowName,
          java.awt.Window frame)
```

Saves the geometry for a specified window in the current module to the SNAP Config file from the `frame` provided.

Parameters:

`windowName` - the String that contains the keyword to save the data under. For example: Veda's main window's keyword is "main."
`frame` - the java.awt.Frame (or derivative) to save geometry from.

getGeometry

```
public void getGeometry(String windowName,
          javax.help.HelpBroker broker)
```

Loads the geometry for a specified help window in the current module from the SNAP Config file into the `HelpBroker` provided. If no geometry is loaded, or an exception results, the broker is set to 640x480 at the center of the screen.

Parameters:

`windowName` - the String that contains the keyword the data was saved under. For example: Veda's main window's keyword is "main."
`broker` - the HelpBroker to retrieve geometry for.

getGeometry

```
public void getGeometry(String windowName,
          javax.help.HelpBroker broker,
          int height,
          int width)
```

(continued from last page)

Loads the geometry for a specified help window in the current module from the SNAP Config file into the `HelpBroker` provided. If no geometry is loaded, or an exception results, the broker is set to the given default dimensions at the center of the screen.

Parameters:

`windowName` - the String that contains the keyword the data was saved under. For example: Veda's main window's keyword is "main."
`broker` - the HelpBroker to retrieve geometry for.
`height` - the default Height for the broker if none is found
`width` - the default width for the broker if none is found

---

getGeometry

```
public void getGeometry(String windowName,
          java.awt.Window frame)
```

Loads the geometry for a specified help window in the current module from the SNAP Config file into the `frame` provided. If no geometry is loaded, or an exception results, the broker is set to 640x480 at the center of the screen.

Parameters:

`windowName` - the String that contains the keyword the data was saved under. For example: Veda's main window's keyword is "main."
`frame` - the java.awt.Window to retrieve geometry for.

---

getGeometry

```
public void getGeometry(String windowName,
          java.awt.Window frame,
          int height,
          int width,
          int x,
          int y)
```

Loads the geometry for a specified help window in the current module from the SNAP Config file into the `frame` provided. If no geometry is loaded, or an exception results, the frame is set to the given default dimensions at the center of the screen.

Parameters:

`windowName` - the String that contains the keyword the data was saved under. For example: Veda's main window's keyword is "main."
`frame` - the java.awt.Window to retrieve geometry for.
`height` - the default Height for the frame if none is found
`width` - the default width for the frame if none is found

---

getGeometry

```
public void getGeometry(String windowName,
          java.awt.Window frame,
          int height,
          int width)
```

Loads the geometry for a specified help window in the current module from the SNAP Config file into the `frame` provided. If no geometry is loaded, or an exception results, the frame is set to the given default dimensions at the center of the screen.

Parameters:

`windowName` - the String that contains the keyword the data was saved under. For example: Veda's main window's keyword is "main."
`frame` - the java.awt.Window to retrieve geometry for.
`height` - the default Height for the frame if none is found
`width` - the default width for the frame if none is found

(continued from last page)

setDividerLocation

```
public void setDividerLocation(String keyword,
          JSplitPane pane)
```

Saves the divider location for the given `JSplitPane` under the given keyword in the current module to the SNAP Config file.

Parameters:

`keyword` - the String that contains the keyword the data was saved under.
`pane` - the JSplitPane (or derivative).

---

getDividerLocation

```
public void getDividerLocation(String name,
          JSplitPane pane,
          int location)
```

Loads the divider location stored under the given keyword in the current module to the `JSplitPane` provided. If the divider location is not found, it will be set to the given default value.

Parameters:

`name` - a String containing the name of the divider.
`pane` - the JSplitPane to retrieve a divider location for.
`location` - the default location.

---

setProperty

```
public void setProperty(String name,
          String propertyValue)
```

Assigns a property to be saved under the given name inside the current module. If the property already exists inside that module, the data already stored gets overwritten by the new value.

Parameters:

`name` - the String that contains the name of the property to be saved
`propertyValue` - the value to assign to the given property

---

getColor

```
public java.awt.Color getColor(String name,
          java.awt.Color def)
```

Retrieves the color stored with the given name from the current module. If no color is found with the given name, the default value is returned.

Parameters:

`name` - a String containing the name of the desired color.
`def` - the Color to return if no Color is found.

Returns:

the Color stored with the given name, or def, if no such color is found.

---

getColor

```
public java.awt.Color getColor(String name)
```

Retrieves the color stored with the given name from the current module.

Parameters:

`name` - a String containing the name of the desired color.

Returns:

(continued from last page)

the Color stored with the given name, or def, if no such color is found.

---

setColor

```
public void setColor(String name,
         java.awt.Color color)
```

Stores the given `Color` with the the given name inside the current module. Colors are stored by storing the RGB value as a string inside the .snaprc file. If a property already exists with the given name, the given Color's RGB replaces the current data.

Parameters:

`name` - a String containing the name of the desired color.

`color` - the Color to store. If this Color is null, nothing is saved.

---

setFont

```
public void setFont(String name,
         java.awt.Font font)
```

Stores the given `font` with the the given name inside the current module. Fonts are stored by saving the Font's Family, Size and Style strings to the .snaprc. If a Fonat already exists with the the given name, the stored data will be replaced with the new data.

Parameters:

`name` - the String that contains the name of the given font

`font` - the Font to store family, size and style data for.

---

getFont

```
public java.awt.Font getFont(String name)
```

Retreives the `font` stored with the given name inside the current module. If there is no font stored with that name a monospaced font size 12 will be returned instead.

Parameters:

`name` - the String that contains the name of the desired font.

Returns:

a Font containing the requested property value; If no Font with the given name is stored, monospaced size 12 with a style of 0 will be returned.

---

getFont

```
public java.awt.Font getFont(String name,
         String family,
         String size,
         String style)
```

Retrieves a named Font from the current module.

Parameters:

`name` - the name of the requested Font

`family` - the default family attribute of the requested Font

`size` - the default size attribute of the requested Font

`style` - the default style attribute of the requested Font

Returns:

a Font containing the requested property value; If no Font with the given name is stored, the default values are used.

(continued from last page)

getCodeInfos

`public Vector` **`getCodeInfos`**`()`

Retrieves the list of codeinfo's from the ExecutionMonitor module.

Returns:

a java.util.Vector of com.cafean.utils.CodeInfo enumerating the executables defined for each code plugin; will never be null;

---

removeAllCodeInfos

`public void` **`removeAllCodeInfos`**`()`

Removes all CodeInfo from the ExecutionMonitor module list in preperation for re-creating the list from a modified copy.

---

addCodeInfos

`public boolean` **`addCodeInfos`**`(CodeInfo s)`

Adds a CodeInfo server to the ExecutionMonitor module list.

Parameters:

`s -` the com.cafean.utils.CodeInfo containing the codeinfo to add to the given configuration Element.

Returns:

the success or failure of the addition.

---

getProperty

```
public String getProperty(String name,
          String def)
```

Retrieves a named property from the current module.

Parameters:

`name -` the name of the requested property

`def -` the value to return if the specified property is not available.

Returns:

a String containing the requested property value or def if no such value is found.

---

getProperty

`public String` **`getProperty`**`(String propertyName)`

Retrieves a named property from the current module.

Parameters:

`propertyName -` the name of the requested property

Returns:

a String containing the requested property value or null if no such value is found.

---

getSnapDBServer

`public Server` **`getSnapDBServer`**`()`

Retrieves the database server information from the Global module.

Returns:

an com.cafean.utils.Server containing the database server info; if none found, returns a new server: ("NONE","5005","user");

setDBServer

`public void setDBServer(Server s)`

Sets the given SnapDB Server info into the Global module.

Parameters:

`s` - the com.cafean.utils.Server containing the snapdb server info to save.

Returns:

the success or failure of the insertion.

getNrcDBServer

`public Server getNrcDBServer()`

Retrieves the NRC Databank server information from the Global module.

Returns:

an com.cafean.utils.Server containing the databank server info; if none found, returns a new server: ("NONE","5007","");

setNrcDBServer

`public void setNrcDBServer(Server s)`

Sets the given NRC Databank Server info into the Global module.

Parameters:

`s` - the com.cafean.utils.Server containing the databank server info to save.

Returns:

the success or failure of the insertion.

getWebSupport

`public WebSupport getWebSupport()`

Retrieves the Web support configuration from the Global module.

Returns:

an com.cafean.utils.WebSupport containing the databank server info; if none found, returns a default object;

setWebSupport

`public void setWebSupport(WebSupport s)`

Sets the given Web support configuration into the Global module.

Parameters:

`s` - the com.cafean.utils.WebSupport containing the connection info to save.

Returns:

the success or failure of the insertion.

removeServer

`public boolean removeServer(String name)`

Removes a calculation server from the Global module.

(continued from last page)

Parameters:

`name` - a String containing the name of the server to remove from the Global module.

Returns:

the success or failure of the removal.

---

getServers

`public LinkedList` **`getServers`**`()`

Retrieves the list of servers from the Global module.

Returns:

a java.util.LinkedList of com.cafean.utils.Server enumerating the server elements contained in global. will never be null;

---

removeAllServers

`public void` **`removeAllServers`**`()`

Removes all calculation server entitys from the global list preperation for re-creating the list from a modified copy.

---

addServer

`public boolean` **`addServer`**`(Server s)`

Adds a calculation server to the global module. Will not add any Server that is not "From File". (e.g. Those loaded from the database.)

Parameters:

`s` - the com.cafean.utils.Server containing the server info to add to the given configuration Element.

Returns:

the success or failure of the addition.

---

getServers

`public LinkedList` **`getServers`**`(SnapCorba snapCorba)`

Retrieves the list of servers from the Global list and adds those to the list of servers retrieved from the given corba interface.

Parameters:

`snapCorba` - the com.cafean.utils.SnapCorba interface to the snapdb server to retrieve a list of calculation servers from.

Returns:

a java.util.List of com.cafean.utils.Server enumerating the server elements contained in global & in the database. will never be null;

---

setLookAndFeel

`public void` **`setLookAndFeel`**`(String lafStr)`

Set the Look-and-feel for the user interface to Windows, Motif, Mac, or the Java Look and Feel. Valid names are: Windows, Metal, Macintosh and Motif. Of these, only Metal is available on all platforms.

Parameters:

`lafStr` - A string identifying the new look and feel.

(continued from last page)

reDirectStdErrOut

`public static boolean` **`reDirectStdErrOut`**`(String prefix)`

Redirects stderr & stdout to a file in the users $HOME/.snap/ directory named `[prefix].screen`.

Parameters:

`prefix -` a String containing the desired prefix for the .screen file.

Returns:

true for success

---

updateAvailable

`public int` **`updateAvailable`**`(String application)`

Retrieves an XML document from the SNAP web server and compares version information contained in the document with current version information.

Parameters:

`application -` a String containing the name of the application

Returns:

1 if a newer version is available 0 if versions match -1 on network error -2 on parser error -3 on data error -4 on unknown error

com.cafean.utils
Class OptionPane

```
java.lang.Object
   ┣-java.awt.Component
      ┣-java.awt.Container
         ┣-javax.swing.JComponent
            ┣-javax.swing.JOptionPane
               ┣-com.cafean.utils.OptionPane
```

public class **OptionPane**
extends JOptionPane

`OptionPane`Extension of JOptionPane, overridden to center dialogs on the display if the parent is the application's mainframe.

Fields inherited from class `javax.swing.JOptionPane`
`CANCEL_OPTION`, `CLOSED_OPTION`, `DEFAULT_OPTION`, `ERROR_MESSAGE`, `ICON_PROPERTY`, `INFORMATION_MESSAGE`, `INITIAL_SELECTION_VALUE_PROPERTY`, `INITIAL_VALUE_PROPERTY`, `INPUT_VALUE_PROPERTY`, `MESSAGE_PROPERTY`, `MESSAGE_TYPE_PROPERTY`, `NO_OPTION`, `OK_CANCEL_OPTION`, `OK_OPTION`, `OPTION_TYPE_PROPERTY`, `OPTIONS_PROPERTY`, `PLAIN_MESSAGE`, `QUESTION_MESSAGE`, `SELECTION_VALUES_PROPERTY`, `UNINITIALIZED_VALUE`, `VALUE_PROPERTY`, `WANTS_INPUT_PROPERTY`, `WARNING_MESSAGE`, `YES_NO_CANCEL_OPTION`, `YES_NO_OPTION`, `YES_OPTION`

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	`OptionPane()` Creates a `OptionPane`with a test message.
public	`OptionPane(Object message)` Creates a instance of `OptionPane`to display a message using the plain-message message type and the default options delivered by the UI.
public	`OptionPane(Object message,int messageType)` Creates an instance of `OptionPane`to display a message with the specified message type and the default options,
public	`OptionPane(Object message,int messageType,int optionType)` Creates an instance of `OptionPane`to display a message with the specified message type and options.

public	`OptionPane(Object message,int messageType,int optionType,Icon icon)` Creates an instance of `OptionPane`to display a message with the specified message type, options, and icon.
public	`OptionPane(Object message,int messageType,int optionType,Icon icon,Object[] options)` Creates an instance of `OptionPane`to display a message with the specified message type, icon, and options.
public	`OptionPane(Object message,int messageType,int optionType,Icon icon,Object[] options,Object initialValue)` Creates an instance of `OptionPane`to display a message with the specified message type, icon, and options, with the initially-selected option specified.

Method Summary

JDialog	`createDialog(java.awt.Component parentComponent,String title)` {@inheritDoc}
static java.awt.Window	`getWindowForComponent(java.awt.Component parentComponent)` Returns the specified component's toplevel `Frame`or `Dialog`.
static int	`showConfirmDialog(java.awt.Component parentComponent,Object message)` {@inheritDoc}
static int	`showConfirmDialog(java.awt.Component parentComponent,Object message,String title,int optionType)` {@inheritDoc}
static int	`showConfirmDialog(java.awt.Component parentComponent,Object message,String title,int optionType,int messageType)` {@inheritDoc}
static int	`showConfirmDialog(java.awt.Component parentComponent,Object message,String title,int optionType,int messageType,Icon icon)` {@inheritDoc}
static String	`showInputDialog(java.awt.Component parentComponent,Object message)` {@inheritDoc}
static String	`showInputDialog(java.awt.Component parentComponent,Object message,Object initialSelectionValue)` {@inheritDoc}
static String	`showInputDialog(java.awt.Component parentComponent,Object message,String title,int messageType)` {@inheritDoc}
static Object	`showInputDialog(java.awt.Component parentComponent,Object message,String title,int messageType,Icon icon,Object[] selectionValues,Object initialSelectionValue)` {@inheritDoc}
static String	`showInputDialog(Object message)` {@inheritDoc}

static String	showInputDialog(Object message,Object initialSelectionValue) {@inheritDoc}
static void	showMessageDialog(java.awt.Component parentComponent,Object message) {@inheritDoc}
static void	showMessageDialog(java.awt.Component parentComponent,Object message,String title,int messageType)
static void	showMessageDialog(java.awt.Component parentComponent,Object message,String title,int messageType,Icon icon)
static int	showOptionDialog(java.awt.Component parentComponent,Object message,String title,int optionType,int messageType,Icon icon,Object[] options,Object initialValue) Brings up a dialog with a specified icon, where the initial choice is determined by the initialValueparameter and the number of choices is determined by the optionTypeparameter.

Methods inherited from class javax.swing.JOptionPane

createDialog, createInternalFrame, getAccessibleContext, getDesktopPaneForComponent, getFrameForComponent, getIcon, getInitialSelectionValue, getInitialValue, getInputValue, getMaxCharactersPerLineCount, getMessage, getMessageType, getOptions, getOptionType, getRootFrame, getSelectionValues, getUI, getUIClassID, getValue, getWantsInput, selectInitialValue, setIcon, setInitialSelectionValue, setInitialValue, setInputValue, setMessage, setMessageType, setOptions, setOptionType, setRootFrame, setSelectionValues, setUI, setValue, setWantsInput, showConfirmDialog, showConfirmDialog, showConfirmDialog, showConfirmDialog, showInputDialog, showInputDialog, showInputDialog, showInputDialog, showInputDialog, showInputDialog, showInternalConfirmDialog, showInternalConfirmDialog, showInternalConfirmDialog, showInternalConfirmDialog, showInternalInputDialog, showInternalInputDialog, showInternalInputDialog, showInternalMessageDialog, showInternalMessageDialog, showInternalMessageDialog, showInternalOptionDialog, showMessageDialog, showMessageDialog, showMessageDialog, showOptionDialog, 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

(continued from last page)

Constructors

OptionPane

```
public OptionPane()
```

Creates a `OptionPane`with a test message.

OptionPane

```
public OptionPane(Object message)
```

Creates a instance of `OptionPane`to display a message using the plain-message message type and the default options delivered by the UI.

Parameters:

`message` - the `Object`to display

OptionPane

```
public OptionPane(Object message,
                  int messageType)
```

Creates an instance of `OptionPane`to display a message with the specified message type and the default options,

Parameters:

`message` - the `Object`to display

`messageType` - the type of message to be displayed: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`

OptionPane

```
public OptionPane(Object message,
                  int messageType,
                  int optionType)
```

Creates an instance of `OptionPane`to display a message with the specified message type and options.

Parameters:

`message` - the `Object`to display

`messageType` - the type of message to be displayed: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`

`optionType` - the options to display in the pane: `DEFAULT_OPTION`, `YES_NO_OPTION`, `YES_NO_CANCEL_OPTION`, `OK_CANCEL_OPTION`

OptionPane

```
public OptionPane(Object message,
                  int messageType,
                  int optionType,
                  Icon icon)
```

Creates an instance of `OptionPane`to display a message with the specified message type, options, and icon.

Parameters:

`message` - the `Object`to display

`messageType` - the type of message to be displayed: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`

`optionType` - the options to display in the pane: `DEFAULT_OPTION`, `YES_NO_OPTION`, `YES_NO_CANCEL_OPTION`, `OK_CANCEL_OPTION`

`icon` - the `Icon`image to display

(continued from last page)

OptionPane

```
public OptionPane(Object message,
                  int messageType,
                  int optionType,
                  Icon icon,
                  Object[] options)
```

Creates an instance of `OptionPane`to display a message with the specified message type, icon, and options. None of the options is initially selected. The options objects should contain either instances of `Component`s, (which are added directly) or `String`s(which are wrapped in a `JButton`). If you provide `Component`s, you must ensure that when the `Component`is clicked it messages `setValue`in the created `OptionPane`.

Parameters:

`message` - the `Object`to display

`messageType` - the type of message to be displayed: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`

`optionType` - the options to display in the pane: `DEFAULT_OPTION`, `YES_NO_OPTION`, `YES_NO_CANCEL_OPTION`, `OK_CANCEL_OPTION`

`icon` - the `Icon`image to display

`options` - the choices the user can select

OptionPane

```
public OptionPane(Object message,
                  int messageType,
                  int optionType,
                  Icon icon,
                  Object[] options,
                  Object initialValue)
```

Creates an instance of `OptionPane`to display a message with the specified message type, icon, and options, with the initially-selected option specified.

Parameters:

`message` - the `Object`to display

`messageType` - the type of message to be displayed: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`

`optionType` - the options to display in the pane: `DEFAULT_OPTION`, `YES_NO_OPTION`, `YES_NO_CANCEL_OPTION`, `OK_CANCEL_OPTION`

`icon` - the Icon image to display

`options` - the choices the user can select

`initialValue` - the choice that is initially selected; if `null`, then nothing will be initially selected; only meaningful if `options`is used

Methods

showMessageDialog

```
public static void showMessageDialog(java.awt.Component parentComponent,
         Object message)
  throws java.awt.HeadlessException
```

{@inheritDoc}

---

showMessageDialog

```
public static void showMessageDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int messageType)
  throws java.awt.HeadlessException
```

showMessageDialog

```
public static void showMessageDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int messageType,
         Icon icon)
  throws java.awt.HeadlessException
```

showConfirmDialog

```
public static int showConfirmDialog(java.awt.Component parentComponent,
         Object message)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showConfirmDialog

```
public static int showConfirmDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int optionType)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showConfirmDialog

```
public static int showConfirmDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int optionType,
         int messageType)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showConfirmDialog

```
public static int showConfirmDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int optionType,
         int messageType,
         Icon icon)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showInputDialog

```
public static String showInputDialog(Object message)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showInputDialog

```
public static String showInputDialog(Object message,
         Object initialSelectionValue)
```

{@inheritDoc}

showInputDialog

```
public static String showInputDialog(java.awt.Component parentComponent,
         Object message)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showInputDialog

```
public static String showInputDialog(java.awt.Component parentComponent,
         Object message,
         Object initialSelectionValue)
```

{@inheritDoc}

showInputDialog

```
public static String showInputDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int messageType)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showInputDialog

```
public static Object showInputDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int messageType,
         Icon icon,
         Object[] selectionValues,
         Object initialSelectionValue)
  throws java.awt.HeadlessException
```

{@inheritDoc}

showOptionDialog

```
public static int showOptionDialog(java.awt.Component parentComponent,
         Object message,
         String title,
         int optionType,
         int messageType,
         Icon icon,
         Object[] options,
         Object initialValue)
  throws java.awt.HeadlessException
```

Brings up a dialog with a specified icon, where the initial choice is determined by the `initialValue`parameter and the number of choices is determined by the `optionType`parameter. If `optionType`is `YES_NO_OPTION`, or `YES_NO_CANCEL_OPTION`and the `options`parameter is `null`, then the options are supplied by the look and feel. The `messageType`parameter is primarily used to supply a default icon from the look and feel.

Parameters:

`parentComponent` - determines the `Frame`in which the dialog is displayed; if `null`, or if the `parentComponent`has no `Frame`, a default `Frame`is used

`message` - the `Object`to display

`title` - the title string for the dialog

`optionType` - an integer designating the options available on the dialog: `YES_NO_OPTION`, or `YES_NO_CANCEL_OPTION`

`messageType` - an integer designating the kind of message this is, primarily used to determine the icon from the pluggable Look and Feel: `ERROR_MESSAGE`, `INFORMATION_MESSAGE`, `WARNING_MESSAGE`, `QUESTION_MESSAGE`, or `PLAIN_MESSAGE`

(continued from last page)

`icon` - the icon to display in the dialog

`options` - an array of objects indicating the possible choices the user can make; if the objects are components, they are rendered properly; non-`String`objects are rendered using their `toString`methods; if this parameter is `null`, the options are determined by the Look and Feel

`initialValue` - the object that represents the default selection for the dialog; only meaningful if `options`is used; can be `null`

Returns:

an integer indicating the option chosen by the user, or `CLOSED_OPTION`if the user closed the dialog

Throws:

`HeadlessException` - if `GraphicsEnvironment.isHeadless`returns `true`

See Also:

`java.awt.GraphicsEnvironment.isHeadless()`

---

createDialog

```
public JDialog createDialog(java.awt.Component parentComponent,
          String title)
  throws java.awt.HeadlessException
```

---

getWindowForComponent

```
public static java.awt.Window getWindowForComponent(java.awt.Component parentComponent)
  throws java.awt.HeadlessException
```

Returns the specified component's toplevel `Frame`or `Dialog`.

Parameters:

`parentComponent` - the `Component`to check for a `Frame`or `Dialog`

Returns:

the `Frame`or `Dialog`that contains the component, or the default frame if the component is `null`, or does not have a valid `Frame`or `Dialog`parent

Throws:

`HeadlessException` - if `GraphicsEnvironment.isHeadless`returns `true`

See Also:

`java.awt.GraphicsEnvironment.isHeadless()`

com.cafean.utils
Class Version

```
java.lang.Object
   |
   +-com.cafean.utils.Version
```

All Implemented Interfaces:
Comparable

---

public class **Version**
extends Object
implements Comparable

A class representing a Version id as a convinient way to compare, load, and store Version information. Originally intended for use as an immutable data structure. Converted to a JavaBean to be able to load/save with the archiver classes.

Field Summary

static final boolean	DEMO Value: **false**

Constructor Summary

public	Version() Creates a new Version instance initialized to 0.0.0
public	Version(int major,int minor,int bugfix) Creates a new Version instance initialized to the given version numbers.
public	Version(int major,int minor,int bugfix,String title,String date,String prefix) Creates a new Version instance initialized to the given version numbers.

Method Summary

int	compareTo(Object o) Compares this object with the specified object for order.
boolean	equals(Object o) Returns true if and only if the given Object is equivilant to this Version.
int	getBugfix() returns the bugfix revision number.
static Version	getCurrent() Returns the current version.

String	`getDate()` Returns the release date for this Version.
int	`getMajor()` returns the major revision number.
int	`getMinor()` returns the minor revision number.
String	`getPrefix()` Returns the prefix or connotation for this Version.
String	`getTitle()` returns this Version's title.
String	`getVersionString()`
int	`hashCode()` Generates an integer hash code for this Version.
boolean	`isDemo()` returns true if this Version is a demo.
static void	`main(String[] args)` Tests the capabilities of this Version class.
static Version	`parseVersion(String version)` Creates a Version instance parsed from the given String.
void	`setBugfix(int bugfix)` Sets this Version's Bugfix Revision Number.
void	`setDate(String date)` Sets this Version's release date.
void	`setDemo(boolean demo)` determines whether this Version is a demo;
void	`setMajor(int major)` Sets this Version's Major Revision Number.
void	`setMinor(int minor)` Sets this Version's Minor Revision Number.
void	`setPrefix(String prefix)` Sets this Versions prefix or connotation.
void	`setTitle(String title)` Sets this Version's application title.
static void	`showDemoMessage(java.awt.Component parent)` A convinience method for telling the user that something is not usable in a demonstration version.

String	toString() Creates a String representation of this Version.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

DEMO

```
public static final boolean DEMO
```

Constructors

Version

```
public Version()
```

Creates a new Version instance initialized to 0.0.0

Version

```
public Version(int major,
               int minor,
               int bugfix)
```

Creates a new Version instance initialized to the given version numbers.

Version

```
public Version(int major,
               int minor,
               int bugfix,
               String title,
               String date,
               String prefix)
```

Creates a new Version instance initialized to the given version numbers.

Methods

getCurrent

```
public static Version getCurrent()
```

Returns the current version.

getMajor

```
public int getMajor()
```

returns the major revision number.

(continued from last page)

getMinor

`public int **getMinor**()`

returns the minor revision number.

getBugfix

`public int **getBugfix**()`

returns the bugfix revision number.

getTitle

`public String **getTitle**()`

returns this Version's title.

getDate

`public String **getDate**()`

Returns the release date for this Version. Should probably return a Date or Calendar instead.

getPrefix

`public String **getPrefix**()`

Returns the prefix or connotation for this Version.

isDemo

`public boolean **isDemo**()`

returns true if this Version is a demo.

setMajor

`public void **setMajor**(int major)`

Sets this Version's Major Revision Number.

setMinor

`public void **setMinor**(int minor)`

Sets this Version's Minor Revision Number.

setBugfix

`public void **setBugfix**(int bugfix)`

Sets this Version's Bugfix Revision Number.

setTitle

`public void **setTitle**(String title)`

Sets this Version's application title.

setDate

`public void **setDate**(String date)`

Sets this Version's release date.

setPrefix

`public void setPrefix(String prefix)`

Sets this Versions prefix or connotation.

setDemo

`public void setDemo(boolean demo)`

determines whether this Version is a demo;

getVersionString

`public String getVersionString()`

toString

`public String toString()`

Creates a String representation of this Version.

parseVersion

`public static Version parseVersion(String version)`

Creates a Version instance parsed from the given String.

Returns:

a Version parsed from the given String

main

`public static void main(String[] args)`

Tests the capabilities of this Version class.

equals

`public boolean equals(Object o)`

Returns true if and only if the given Object is equivilant to this Version.

hashCode

`public int hashCode()`

Generates an integer hash code for this Version.

compareTo

`public int compareTo(Object o)`

Compares this object with the specified object for order. Returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

showDemoMessage

`public static void showDemoMessage(java.awt.Component parent)`

A convinience method for telling the user that something is not usable in a demonstration version.

com.cafean.utils
Interface WriteMetrics

public interface **WriteMetrics**

Method Summary

boolean	`writeMetrics(java.io.PrintWriter pw)` Write the data block metrics to an XML file.

Methods

writeMetrics

`public boolean writeMetrics(java.io.PrintWriter pw)`

Write the data block metrics to an XML file.

Parameters:

`pw` - the file to write the entry.

Returns:

true - success, false - failure.

Index

A

abs 692
AbstractBeanComponent 5, 6
AbstractComponent 12, 13
AbstractInsertHandler 623
AbstractModel 32
AbstractPathHandler 626
actionPerformed 312, 543, 616
activate 597, 604, 606, 620
activateRedoButton 362
activateUndoButton 362
add 92, 218, 662, 666, 673, 676, 680, 691
addAbstractModel 365
addAnnotation 142, 324
addBoxSelectionListener 227
addChild 57
addCodeInfos 709
addComment 48, 102
addComponent 34, 35, 139
addComponentListener 20
addComponentLookup 44
addComponents 139, 323
addConnection 18, 81
addConnectionSelectionListener 255
addContextHelp 373
addCurrentExportItems 642
addCurrentToolsItems 642
addDrawnComponent 323
addDrawnConnection 217
addDrawnViewTab 359
addElement 41
addExportItem 369
addImportItem 369
addInputId 175
addMenuItem 328, 369
addMessage 13, 363, 364, 380, 381, 382
addMouseHandler 466
addMouseListenerToHeaderInTable 442
addMultipleComments 48, 102
addNotify 216, 293, 465
addOption 123
addOutputId 174
addOverlapPanel 322
addParameter 164
addPibBlock 138
addPlaybackPanel 373
addPoint 303, 498
addPopupMenuEntries 37
addPopupMenuItems 476, 487, 500
addPropertyChangeListener 546, 580
addRegisteredDialog 372
addSelectedComponent 223
addSelectedComponents 224
addServer 711
addSound 362, 382
addToFront 219
addToModel 15
addToolbar 322
addToolChangeListener 449
AlertMsg 378
AlertSound 379
ALL 569
Angle 650
Annotation 474
AnnotationAction 589
areComponentsScaled 425
ArrayChangedEvent 189
arraysEqual 691
ARROW_FILLED 495
ARROW_HOLLOW 495
ARROW_LINE 496
ARROW_NAMES 496
ARROW_NONE 495
ARROW_TYPES 496
AsciiViewer 205

B

BaseNumber 654
BeanAction 591
BeanBox 216
BooleanEditor 518
BorderStyleEditor 522
BOTTOM 273
boundsChanged 218
boxSelectionChanged 228
BRITISH 654

buildView 138

C

calcSelectionBounds 224
canAddPoint 303, 498
canAlign 222
canBeResized 280, 614
cancelInsert 602, 623
canConnectTo 17, 138
canCopy 221
canDelete 221
canPaste 221
canRedo 454, 456, 458, 460
canRemovePoint 303, 498
canRepair 129
canUndo 454, 456, 458, 460
capacity 662, 666, 676, 680
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 593
ccNumberCheck 40
CENTER 273
CENTER_H 273
CENTER_V 273
checkModel 38, 440
checkRealArrayList 108
checkRealArrayTable 108
CIRCLE 271
cleanUpDeleted 46
Clear 277
clear 81, 92, 661, 665, 675, 679
clearCheckErrors 364, 380
clearCheckWarnings 365
clearComponentLookup 43
clearConnections 18
clearDbIds 44, 91, 104
clearInputs 175
clearLinks 277
clearOutputs 174
clearRubberBoxes 341
clearSelection 224, 255, 327
clearViewSelection 142
clearWarningCount 380
ClientPluginLoader 229
clone 14, 58, 73, 79, 81, 94, 102, 121, 137, 162, 172, 185, 247, 276, 655, 661, 665, 675, 679, 689
closeAbstractModel 360
closeAllViews 106
closeMessageWindow 360
COLOR_OPTIONAL 569
compare 439
compareDouble 656
compareRowsByColumn 439
compareTo 58, 105, 672, 690, 726
compareVersion 45
complete 17, 143, 168, 177, 187
completeConnection 598
componentChanged 65, 206, 292, 306
ComponentChangedEvent 191
componentConnected 65, 206, 293
componentDeleted 65, 206, 292
componentDisconnected 66, 206, 293
componentHidden 488
ComponentList 63
componentMoved 488
componentRemoved 305
componentReshaped 305
componentResized 488
ComponentSelectionDialog 234
ComponentSelectionEditor 526
ComponentSelector 239, 240
componentShown 488
Configurator 704
ConnectingPt 245, 246
Connection 70, 71
ConnectionBean 76
ConnectionData 79
ConnectionList 81
connectionPointRemoved 218, 305
ConnectionSetPanel 254

connectLinks 278
ConnectMouseHandler 597
ConnectSound 379
connectTo 19
contains 82, 92, 283, 304, 496, 497, 661, 665, 675, 679
containsRestartChanges 46
convert 650, 655, 671, 688
copy 7, 14, 137, 221
copyFrom 6, 102, 138, 162, 172, 184
createBeanSelectionMenu 449
createBorderRegion 282
createCategoryToolBars 449
createCenterShape 282
createComponent 35, 60, 590, 592
createConnectionPrototypes 276
createConnectionPt 276
createCursor 450
createDataPages 106
createDialog 721
createDisplayBeans 274, 302, 451
createDrawnComponent 20, 74, 142, 173, 180
createDrawnConnection 307
createModel 361
createNewModel 634
createPibBlock 489
createPopupMenu 207, 224, 289
createSelected 450
createSourceData 19
createTargetData 19
createTemplateEntry 294, 307
CreateViewsDialog 258
createZoomMenu 225
CROSSFLOW_DROP_ZONE 245
CROSSHATCH 272
CURSOR_CONNECT 596
CURSOR_CONNECT_COMPLETE 597
CURSOR_CONNECT_OFF 596
CURSOR_MANIP_P 448
CURSOR_PAN 610
CURSOR_PAN_GRIP 610
CURSOR_ZOOM_IN 619
CURSOR_ZOOM_OUT 619
cut 221

D

DATA_COMPLETE 101
DATA_ERROR 101
DATA_INCOMPLETE 100
DATA_WARNING 101
dateStamp 366, 382
DBtypeCode 16
DCB_MINI_NAV 423
DCB_NAV_SELECT 423
deactivate 597, 606, 613, 620
deactivateRedoButtons 367
debug 108, 358
deepClone 59
DEFAULT_SIZE 301
delete 221
deleteAllComments 49, 103
deleteComment 49, 103
deleteRubberBox 341
deleteRubberBoxes 341
deleteRubberLine 341
deleteRubberLines 340
DEMO 724
destroy 705
DIAMOND 272
DIFF_VAL 549
Dimless 658
disable 209
DISABLED 569
disableMainFrame 367
disconnect 17, 73
disconnectAllMyLinks 288
disconnectFrom 17
DisconnectSound 379
divide 673, 691, 692
divideby 691
doMultiEdit 398
DoubleArray 661
DOWN 273
draw 279, 304, 313, 334
drawLabelStrings 279
DrawnComponent 273
DrawnConnection 301, 302
DrawnUserValue 312
DrawnView 321
DrawnViewComponent 332

drawRubberBox 341
drawRubberBoxes 341
drawRubberLine 340
drawRubberLines 340
DummyCompElement 83
dumpBlockParams 6, 76, 86, 124

E

editorChangeValue 522
ElementBean 86
ElementList 90
ElementsAddedEvent 193
ElementsRemovedEvent 195
EllipticalAnnotation 481
enableMainFrame 367
end 454, 458
ensureVersion21 514
equals 58, 79, 83, 106, 672, 690, 726
execute 176
executeModelValidations 53
executeUserDefinedFunctions 51
exportModelMetrics 53
exportModelMetricsSpec 53

F

findByCC 63
findByDB_ID 91
findByIdent 90, 91
findClientPlugin 370
findComponentByCC 33
findComponentByDB_ID 34
findComponentByIdent 34
findComponentGroup 38
findComponentsInside 222
findConstant 176
findDisplayBeanClass 451
findElementByDB_ID 42
findElementByIdent 42
findEmbedded 333
findEquivalentSharedComponent 51
findFunction 177
findModelByIdent 365
findPlugin 370
findReal 52
findVariable 177
finishInsert 602, 623
fireComponentChanged 20, 144
fireComponentConnected 20
fireComponentDeleted 20, 144, 181
fireComponentDisconnected 20
firePropertyChange 556, 587
fixme 108
flip 284
FloatArray 665
formatChanged 410
forwardMouseEvent 343
FunctionSource 152, 153

G

GenericObject 101
get 91, 662, 666, 676, 680
getAbstractModelAt 365
getAbstractModelCount 366
getAlertSound 432
getAnnotationArray 143
getAnnotations 217, 328
getArcHeight 508
getArcWidth 507
getArray 395
getArrows 500
getArrowSize 501
getArrowTypeName 496
getAsciiPrintWriter 635
getAsciiStyledDocument 635
getAsText 519, 523, 527, 532, 539, 550, 577, 580, 586
getAttributeGroup 149, 432
getAttributeGroups 149, 432
getAttributeIndex 148, 167, 186, 435, 570
getAttributesForGroup 149, 432
getBackground 144, 513
getBadgeColors 249
getBadges 249
getBaseErrorColor 427
getBaseHeatedColor 427
getBaseIncompleteColor 427
getBaseNormalColor 427
getBaseNormCsColor 427

getBasePoweredColor 430
getBaseSelectedColor 427
getBaseSpecialColor 427
getBaseWarningColor 427
getBeanBox 291, 328, 465
getBorderType 523
getBugfix 725
getButtonIcon 597, 604, 607, 610, 613, 619
GetCalcServeritorPort 424
getCanvasFont 429
getCanvasFontItalic 429
getCanvasFontItalicMetrics 429
getCanvasFontMetrics 429
getCanvasSize 140, 326
getCatCCComparator 19
getCategories 33
getCategory 13, 73, 93, 138, 163, 173, 185, 528, 593
getCategoryObject 36
getCCnumber 105, 149
getCCNumberComparator 18
getCCNumberIncrement 39
getCellEditorValue 397
getChildren 57
getClassPrereqs 645
getClientPluginAt 370
getClientPluginCount 370
getClipboard 359
getClockwiseFace 286
getCodeInfos 708
getCodePlugins 371
getCollapsedIcon 58
getColor 707
getColorArray 144
getColumnClass 441
getColumnCount 441
getColumnName 441
getComment 49, 103
getComments 103
getComponent 14, 61, 84, 191, 286, 431
getComponentCategory 450
getComponentCCNumber 106
getComponentCount 36
getComponentDependencies 14
getComponentGroups 38
getComponentID 287
getComponentIterator 35
getComponentNumber 106
getComponents 33, 225, 325
getConfigFileDir 704
getConfigFileName 704
getConnectingLocation 295
getConnectingPt 288, 289, 333
getConnectingPtAt 283
getConnectionColor 74, 427
getConnectionCount 18, 82
getConnectionData 246
getConnectionIndex 79
getConnectionName 21
getConnections 18, 81
getConnectionSize 426
getConnectionSource 344, 598
getConnectionStroke 74
getConnectionTypes 13
getConnectSize 288
getConnectSound 434
getConversionFactor 650, 658, 670, 687, 696
getCounterFace 286
getCreateDate 47
getCrossflowIndex 284
getCurrent 724
getCurrentCursor 449, 450, 597, 601, 607, 611, 614, 619, 623
getCurrentDisplayValue 689
getCurrentIndex 163
getCurrentModel 360
getCurrentParametric 165
getCurrentTool 450
getCurrentValue 163, 180, 185
getCursor 597, 607, 614
getCustomEditor 519, 524, 527, 533, 540, 546, 551, 577, 581, 587
getCustomPopupActions 16, 290
getCustomPopupItems 16, 74, 77, 141, 173, 294, 333
getData 124
GetDatabaseHostName 424
GetDatabaseName 424
getDataState 105
getDate 725
getDB_ID 105
getDeckWriter 37
getDefaultDrawLength 278

getDefaultDrawWidth 277
getDefaultIndex 164
getDefaultUnits 434
getDefaultValue 166
getDescription 48, 103, 616
getDimensionless 52
getDisconnectSound 434
getDisplayName 128, 687
getDisplayValue 651
getDividerLocation 707
getDocDescription 73
getDocumentLinks 50
getDoubleClickBehavior 423
getDoubleValue 410, 689
getDrawAngle 278
getDrawingFace 284
getDrawingObject 278
getDrawnComponent 217
getDrawnComponentAt 324
getDrawnComponentCount 324
getDrawnComponents 225, 324
getDrawnView 140, 468, 474
getElementAt 42
getElementCount 41
getElementIterator 41
getEmbedConnArray 143
getEmbeddedCons 145
GetEnaPort 424
GetEnaServerName 424
getEndValue 166
getENG_Units 650, 654, 658, 671, 687, 696
getErrorColor 426
GetExeServer 425
getExpandedIcon 58
getExportNote 51
getExportUnits 52
getFace 390
getFaceByAngle 285
getFeaturePlugins 371
GetFileArcPortNumber 424
getFileChooser 372
getFillColor 279
getFont 513, 708
getForeground 513
getFormat 409, 650, 655, 672, 689, 697
getFunction 176
getGeometry 705, 706
getGlassPane 291, 465
getGridLineColor 146
getGroupedConnections 21
getHandle 249
getHandlerID 607
getHandleSize 250, 288, 426
getHead1 501
getHead2 501
getHeatedColor 427
getHorizGridSpacing 147
getHorizSnapSpacing 147
getHtmlUnits 687
getIcon 616
getIdent 104
getImage 142
getImageData 142, 489
getImageElement 145
getImageURL 58
getIncompleteColor 426
getIncrement 166
getIndexes 194, 196
getIndexForRow 442
getIndicatorColor 274
getInfoSound 433
getInitialCCNumber 38
getInputAt 175
getInputCount 174
getInputIdAt 175
getInputValue 175, 176
getInternalSound 434
getIterator 82
getJavaInitializationString 519, 523, 527, 533, 540, 546, 551, 576, 581, 586
getKeys 124
getLabel 314, 514, 635
getLabelPlural 635
getLastCCNumber 40
getLastPath 430
getLayoutIterations 432
getLeftComponent 72
getLeftComponentID 72
getLeftConnectData 71
getLength 278

getLenScaleFactor 50, 275
getLineThickness 482, 499, 508
getLoadFormat 689
getLocation 140, 390
getLongValue 410
getLoopCheck 38
getMajor 724
getMajorCreationVersion 107
getMajorVersion 107
getMaxCCNumber 39
getMaxHeight 286
getMaximumSize 219, 340, 467
getMaxWidth 286
getMenubar 322
getMessageFont 430
getMinimumSize 219, 334, 467, 488
getMinor 724
getMinorCreationVersion 108
getMinorVersion 107
getMinWidth 286
getMirrorImageShape 282
getMode 372
getModel 14, 53, 84, 117, 118, 217, 322, 360, 398, 412, 431, 439, 528, 582, 587
getModelDB_ID 43
getModelMajorVersion 45
getModelMinorVersion 45
getModelOptions 49
getMouseHandler 466
getName 16, 48, 102, 127, 616, 641, 694
getNeedsTarget 251
getNewArray 190
getNewCompIdent 14, 108
getNewComponentIdent 36
getNewElementIdent 42
getNewInsertHandler 483, 501, 508, 628
getNextCCNumber 39
getNormalColor 426
getNormalObj 276
getNormCsColor 426
getNrcDBServer 710
getNumberConnections 288
getNumberOfErrorsFound 362, 382
getNumberOfWarningsFound 362
getNumberValue 410
getNumComments 49, 103
GetNumExecServers 424
getNumSelected 224
getOldArray 190
getOppositeFace 285
getOption 123
getOrb 369
getOrder 21
getOrderComparator 18
getOrientation 285
getOrientationMenu 289
getOrientationName 274
getOtherSide 71
getOtherSideData 71
getOutputAt 174
getOutputCount 173
getOutputIdAt 173
getOutputStream 380
getOwner 14, 47, 61, 84, 153, 432
getPad 248
getPadXY 390
getPanel 314
getPanelSize 219, 467
getParameterAt 163
getParameters 164
getParametersCount 163
getParent 57, 293
getParentDB_ID 44
getPath 306, 497
getPdfViewer 373
getPixelsPerMeter 145
getPlugin 33
getPluginAt 370
getPluginClass 645
getPluginCount 371
getPluginIcon 635
getPluginId 36, 641, 645
getPluginInfo 643
getPluginPreferences 643
getPluginPrereqs 643, 645
getPluginType 634, 637, 641
GetPortNumber 424
getPoweredColor 430
getPreferredSize 219, 290, 313, 467
getPrefix 725

getPrefs 368
getPresentationName 454, 456, 458, 460
getProgramOptions 49
getProjectDB_ID 43
getPropertiesOrientation 431
getProperty 709
getPythonHeader 177
getPythonInit 177
getRealByIndex 52
getRealSize 17
getRecentFileName 430
getReferencedValue 692
getReferenceModel 366
getRegisteredDialogs 372
getRightComponent 72
getRightComponentID 72
getRightConnectData 71
getRootComponents 49
getRootParent 57
getRotatedLocation 250
getRowCount 440
getRunValues 168
getSamPackage 634
getSaveFile 44
getSaveFileName 44
getScale 220, 468
getScaleComponents 425
getScrollPane 468
getSelected 224
getSelectedColor 426
getSelectedComponents 325
getSelectedConnector 283
getSelectedDropZone 284
getSelectedIndex 386
getSelection 222, 224, 236, 240
getSelectionBounds 224
getServers 711
getSharedComponents 13
getShortDescription 128
getShowCreateViewDialog 435
getShowPoints 141
getShowWelcomeDialog 435
getSI_Units 650, 654, 658, 671, 687, 696
getSignificantFigs 693
getSize 140
getSnapDBServer 709
getSnapHomeDirectory 371
getSound 425
getSource 305
getSourceCode 153
getSpecialColor 427
getSplashStatus 359
getSplashTitle 360
getStartValue 166
getStrippedValue 692
getSubComponentAt 294, 333
getTableCellEditor 387, 529
getTableCellEditorComponent 398
getTableCellRendererComponent 402
getTags 519, 524, 527, 533, 540, 546, 551, 577, 581, 586
getTarget 305
getText 513
getThisSideData 72
getTitle 328, 725
getToolbox 322
getToolTipText 58, 278, 304, 340, 499
getTooltipText 608
getTransferData 692
getTransferDataFlavors 692
getTransparency 146
getType 247
getUndoManager 361
getUndoSupport 361
getUnitIndex 52
getUnitName 689
getUnits 52, 166, 180, 186, 314, 656, 687
getUnitsDisplay 52
getUnitType 654
getUsedBounds 305, 335, 499
getUserName 369
getUserSound 433
getValidationOptions 53
getValidationTests 53
getValidLabel 371
getValue 411, 519, 523, 527, 533, 539, 546, 551, 576, 581, 586, 671, 688
getValueAt 440
getValueLabel 315
getValueText 315
getVerbosity 129

getVersion 360, 643, 645
getVersionString 726
getVertGridSpacing 147
getVertSnapSpacing 147
getViewComponent 225, 322
getViewportCenter 465
getViewPosition 141, 326
getViewSize 326
getWarningColor 426
getWarningCount 382
getWarningSound 433
getWebSupport 710
getWidthScaleFactor 50, 149, 225, 275
getWindowForComponent 721
getWindowingMode 431
getWriteable 206
getX_Pos 287
getXDistBetweenCPs 280
getXDistBetweenXflowCPs 280
getXpdfPath 435
getY_Pos 287
getZoomablePanel 292, 328
getZoomScale 141, 326
getZoomSize 467
GlassPanel 339
greaterthan 672, 673, 690, 691

H

halfPI 649
HALFWIDTH 301
hasBlocks 139
hashCode 672, 726
hasModelMetrics 53
hasNext 156, 164
hasSubComponents 294, 333
hideComponentNumber 240
hideComponentType 240

I

ICON_ELL_ANNOT 448
ICON_IMAGE_ANNOT 448
ICON_LINE_ANNOT 448
ICON_NEW 448
ICON_REC_ANNOT 448
ICON_TXT_ANNOT 448
ImageAnnotation 487
INACTIVE 550
includeInLoopcheck 21
incrementMajorVersion 46
incrementMinorVersion 46
indexOf 92, 662, 666, 676, 680
InfoMsg 378
InfoSound 379
init 254, 368, 534, 581, 586
InitDrawing 274, 302, 312, 333
INLET_DROP_ZONE 245
INLET_MOVEABLE_CONNECTOR 245
insertFailed 412
InsertMouseHandler 601
instance 357
Int 670
IntegerArray 675
InteractiveMouseHandler 603
InternalErrMsg 379
InternalErrSound 379
IntNonMultiEdit 532
IntSpinnerEditor 538, 539
inverseTransformPoint 468
INVISIBLE_CONNECTOR 245
isAnimatable 635
isAssociated 638
isAutoScale 290
isBeanBased 635
isBeepOn 425
isCanceled 241
isCancelled 236, 386, 394, 527
isCellEditable 441
isCompActionSelected 450
isConnected 248
isContained 64
isDashed 501
isDatabaseConfigured 423
isDataFlavorSupported 692
isDeleted 104
isDemo 725
isDirty 51
isDrawBadges 290
isEditable 313

isEditingRestart 46
isEnabled 128
isEqualTo 74
isEquivalent 119
isFilled 482, 507
isGeneratePoints 306
isGridAbove 148
isIncremental 165
isIndependentComponent 71
isInProgress 597
isJeditAvailable 423
isKnown 655, 693
isLocked 148
isMostlyVerticalDown 651
isMostlyVerticalUp 651
isObjectInsideBounds 282, 304, 499
isOkayForExport 17, 18
isOpaque 507
isPaintable 519, 523, 527, 533, 540, 546, 551, 576, 581, 586
isParametric 165
isPlenumShaped 275
isPosnSet 289
isPropertyActive 149, 167, 186, 436, 570
isPropertyEnabled 148, 167, 186, 435, 569
isPropertyRequired 148, 167, 186, 435, 570
isPropertyResizable 149, 167, 186, 436, 570
isPropertyRestartEditable 148, 167, 186, 435, 570
isRepresenting 333
isRestartableModel 46
isRestartResizable 149, 167, 187, 436, 571
isRounded 507
isScalable 280
isSegmentSet 274
isSelected 224, 293
isShowGrid 145
isShowLabel 313
isShowNumbers 551
isShowUnits 313
isSingleFrame 431
isSnapToGrid 146
isSorted 92
isSoundOn 425
isSubset 57
isSuperset 57
isUncertain 412
isUnknown 410, 655, 693
isValidCCNumber 40
isValveShaped 275
isVerticalDown 651
isVerticalUp 651
isViewVisible 142
isVisible 248
isVisual 58, 74
itemStateChanged 522, 550, 556, 558, 561, 567
iterate 164
iterator 64, 93

J

joinPipe 47

L

label 16, 74, 137, 163, 176, 185
layoutComponents 40
layoutView 138, 324
LEFT 272
lessthan 672, 690
LimitedLongStringEditor 543
LineAnnotation 496
load 137, 423, 637
loadComponent 145
loadComponents 222
loadDisplayBean 451
loadDisplayBeanClasses 452
loadDrawnComponent 51, 294, 307, 316, 333
loadMainMenuItems 642
loadOptions 128
loadPlugin 645
loadReferenceModel 366
loadRestartData 37
loadSettings 642
loadUserConstant 200
loadUserFunction 200
loadUserVariable 199
loadVedaExport 451
loadViewMenuItems 642
loadVisualComponents 199
LocalSubmitDialog 347
locateComponentAt 343

lockToString 687
LongArray 679
LongStringEditor 546
lookupComponent 44
lostOwnership 368, 412

M

main 369, 662, 666, 676, 680, 726
MainFrame 358
makeArray 659, 696, 697
makeBumpyTexture 430
max_positions 273
MECodePlugin 634
MEDReader 199
MEFeaturePlugin 637
menuCanceled 327
menuDeselected 327
menuSelected 327
MEPlugin 640
MEPluginData 644
MessageWindow 379
minimizeView 226
MODE_CLEAR_CONSTANTS 358
MODE_EXPORT_ASCII 358
MODE_IMPORT 358
MODE_NORMAL 358
MODE_SAVE_MED 358
modelAdded 637
modelRemoved 637
mouseClicked 207, 291, 303, 334, 342, 476, 497, 598, 601, 607, 611, 623, 626, 630
mouseDragged 291, 304, 342, 498, 598, 601, 604, 611, 613, 620, 626, 630
mouseEntered 207, 290, 304, 342, 476, 499, 607, 623
mouseExited 207, 291, 304, 342, 476, 499, 608, 624
MouseHandler 606
mouseMoved 291, 304, 334, 342, 499, 598, 601, 608, 623, 626
mousePressed 207, 291, 302, 342, 476, 497, 598, 601, 604, 610, 613, 620, 626, 630
mouseReleased 207, 291, 303, 342, 476, 498, 598, 601, 604, 611, 613, 620, 627, 630
move 250
moveRel 281
moveTo 281, 475, 496
multiply 673, 691

N

NAME_NUMBER 155
NAME_VALUE 155
NamedIntEditor 550
NamedValueSelector 385, 386
NamelistBooleanEditor 555
NamelistDoubleEditor 558
NamelistIntEditor 561
NamelistNamedIntEditor 562
NamelistRealEditor 566
next 156
NONE 271, 569
normalize 411
normalizeArray 693
NoSound 379
Num_Databases 422

O

objectAdded 40
offsetWindowLocation 368
open 634
openFileSelector 367
openMessageArea 364
OPTIONAL 569
OptionPane 716, 717, 718
organizeSelection 226
organizeView 225
OUTLET_DROP_ZONE 245
OUTLET_MOVEABLE_CONNECTOR 245

P

Pad 389
paint 220, 279
paintComponent 279, 343, 482, 488, 496, 506
paintImmediately 220, 221
paintValue 519, 523, 528, 533, 540, 547, 551, 576, 581, 587
PanMouseHandler 610
ParametricIteration 155
parseVersion 726
paste 222

performExport 259
performLoopCheck 38
PIXELS_P_METER 273
PIXELS_PER_METER 136
pluginException 230
pluginLoaded 229
popupBeanDataDialog 6
popupDataDialog 7, 16, 77, 106, 145, 172, 181
popupEditor 315, 475, 482, 488, 497, 507, 513
print 220, 279
printView 325
processCommand 361, 641
PropertyEditorPanelSupport 575

R

rangeCheck 109, 110, 111
readAnnotationArray 143
readBlock 86, 124
readBlockParams 6, 77, 87, 124
readByteArray 316
readDrawingBlock 199
readEmbedConnArray 143
readTemplateEntry 294, 307
Real 686
RealArrayDialog 393, 394
RealArrayEditor 580
RealBeanEditor 585
RealEditor 397
RealRenderer 402
RealTextField 408, 409
rebuildConnections 16
RECENT_START 423
reconnectIdentReferences 43, 73, 81, 91, 104, 116, 173
reconnectImage 15
RectangularAnnotation 506
RectangularInsertHandler 629
reDirectStdErrOut 711
redo 454, 456, 458, 460
redrawSelection 226
refresh 206, 220, 255, 414, 466, 476
refreshCompBounds 466
refreshSteamTable 367
remove 92, 218, 662, 666, 676, 680
removeAbstractModel 365
removeAllCodeInfos 709
removeAllServers 711
removeBoxSelectionListener 227
removeClosestPoint 303, 497
removeComponent 35
removeComponentListener 20
removeConnection 82
removeConnectionSelectionListener 255
removeDeleted 91
removeDrawnViewTab 359
removeElement 41
removeFromModel 15, 137
removeInputAt 175
removeMouseHandler 466
removeNotify 293, 340
removeOutputAt 174
removeParameter 164
removePlaybackPanel 373
removePoint 303, 498
removePropertyChangeListener 547, 581
removeRegisteredDialog 372
removeSelectedComponent 223
removeServer 710
removeToolChangeListener 449
removeVerify 15, 137
renodalizePipeCells 48
renumberComponents 50
renumberSelectedComponents 225
repaint 220, 305, 343, 499
repair 129
replaceNumericNames 181
reportModelCheck 41
repositionLinks 284
requestSaveFileName 361
REQUIRED 569
requiresTarget 344, 598
reset 488
resetAllUnits 361
resetConnections 226
resetCursor 366
resetIteration 156, 164
resetModule 423
resetPosition 281
resetSize 476, 482, 488, 499, 507, 514
resetWindowingMode 359

resetZoomMenu 326
restoreState 21, 87, 101, 125, 168, 181, 187, 223, 292, 306, 315, 334, 475, 482, 489, 500, 513, 694
revalidate 466, 500
RIGHT 272
rotate 250
rotateTo 281
run 209
runValidation 128

S

save 423, 637, 704
saveModel 37
saveOptions 128
scaleIt 280
SEGMENT_BOTH 272
SEGMENT_INLET 272
SEGMENT_NONE 272
SEGMENT_OUTLET 272
SEGMENT_SPECIAL 272
selectCategory 226
selectComp 382
SelectMouseHandler 613
SELECTOR_OPEN 358
SELECTOR_SAVE 358
set 662, 666, 676, 680
setAlertSound 433
setAllowEmpty 235
setArcHeight 508
setArcWidth 507
setArrows 500
setArrowSize 501
setAsText 519, 523, 528, 533, 540, 547, 551, 577, 581, 586
setAutoScale 290
setAvailable 526
setAvailableComponents 235
setBackground 144, 316, 513
setBackupComponent 274
setBadgeColors 249
setBadges 249
setBaseErrorColor 428
setBaseFile 156
setBaseHeatedColor 428
setBaseIncompleteColor 428
setBaseNormalColor 428
setBaseNormCsColor 428
setBasePoweredColor 431
setBaseSelectedColor 428
setBaseSpecialColor 428
setBaseWarningColor 428
setBorder 475
setBounds 276, 334, 475
setBugfix 725
setCanvasSize 140, 326
setCategory 93, 234, 528
setCCNumberIncrement 39
setChildLocation 368
setColor 708
setColorArray 144
setComments 104
setComponent 287
setComponentNumber 17, 106
setComponentNumberConstrained 7
setConnected 248
setConnection 454, 458
setConnectionColor 429
setConnectionIndex 79
setConnectionSize 426
setConnectionStart 598
setConnectPtType 248
setConnectSound 434
setConstants 156
setContext 550, 556, 558, 559, 561, 567
setCreateDate 47
setCreationVersion 107
setCreator 235, 528
setCurrentAction 450
setCurrentModel 360
setCurrentParametric 165
setCurrentTool 449
setCurrentValue 185
setCurrentValueDouble 185
setDashed 501
setData 124
setDataState 105
setDate 725
setDB_ID 105
setDBServer 710
setDefaultIndex 165

setDefaultUnits 435
setDefaultValue 166
setDeleted 16, 104
setDemo 726
setDescription 48, 103
setDimension 582
setDirty 51
setDisconnected 248
setDisconnectSound 434
setDividerLocation 706
setDoubleClickBehavior 424
setDrawAngle 278
setDrawBadges 290
setDrawHeight 288
setDrawWidth 288
setEditable 313, 527
setEmbeddedCons 145
setEnabled 128
setEndValue 166
setEqualTo 277, 482, 497, 507
setErrorColor 429
setExportNote 51
setFace 390
setFileChooserLocation 368
setFilled 482, 507
setFont 513, 527, 708
setForeground 316, 513, 576, 586
setFormat 409
setFunction 176
setGeneratePoints 306
setGeometry 705
setGridAbove 148
setGridLineColor 146
setHandle 249
setHandlerID 607
setHandleSize 249, 426
setHead1 501
setHead2 501
setHeatedColor 429
setHeight 475
setHorizGridSpacing 147
setHorizSnapSpacing 148
setIdent 42, 43, 104
setImageData 142, 489
setImageElement 145
setIncompleteColor 429
setIncrement 166, 539
setIncremental 165
setInfoSound 433
setInitialValues 348
setInputIdAt 175
setInternalSound 434
setKeys 124
setKnown 655
setLabel 314, 514
setLabelString 289
setLastPath 430
setLayoutIterations 432
setLeftComponent 72
setLeftComponentID 72
setLenScaleFactor 50, 275
setLineThickness 483, 500, 508
setLocation 140, 302
setLocked 148, 322, 449
setLockedConstrained 148
setLookAndFeel 711
setMajor 725
setMajorCreationVersion 107
setMajorVersion 107
setMaximum 539
setMinimum 539
setMinor 725
setMinorCreationVersion 107
setMinorVersion 107
setMode 372
setModel 15, 117, 156, 235, 398, 412, 439, 528, 582, 587
setModelDB_ID 42
setModelMajorVersion 45
setModelMinorVersion 46
setModule 705
setName 48, 102, 181, 641
setNameConvention 156
setNeedsTarget 251
setNormalColor 428
setNormCsColor 428
setNrcDBServer 710
setOpaque 481
setOrder 21
setOrientation 285
setOrientationByAngle 285

setOutputIdAt 173
setOutputStream 380
setOutputValue 174
setOwner 47, 153
setPad 248
setPadXY 389
setPaintEnabled 220
setPanel 314
setPanelSize 466
setParameter 168
setParameterAt 164
setParameters 163
setParametric 165
setParent 293
setParentDB_ID 44
setPath 302, 497
setPixelsPerMeter 145
setPlaybackTime 373
setPoweredColor 430
setPrefix 123, 726
setProgramOptions 50
setProjectDB_ID 43
setPropertiesOrientation 431
setProperty 707
setRecentFileName 430
setRightComponent 73
setRightComponentID 72
setRounded 507
setRunOptions 348
setSaveFile 45
setScale 219, 467
setScaleComponents 425
setSelected 240, 293, 306
setSelectedColor 429
setSelectedComponent 223
setSelection 235
setShowCategory 235
setShowComponentNumber 235
setShowCreateViewDialog 435
setShowGrid 146
setShowHS_connections 425
setShowLabel 313
setShowNumbers 552
setShowPoints 141
setShowSize 539
setShowUnits 314
setShowWelcomeDialog 435
setSignificantFigs 693
setSize 140, 514
setSizeTo 280
setSnapToGrid 146
setSound 425
setSourceCode 153
setSpecialColor 429
setSplashStatus 359
setStartValue 166
setStoredBackground 144
setTargetComponent 206
setText 410, 513
setTextLabelString 575
setTitle 235, 328, 725
setTransparency 146
setUnits 52, 166, 180, 185, 314
setUnitsConstrained 52
setUnitType 654
setUnknown 654, 690
setupContextHelp 380
setUpEditor 398
setUpRealEditor 398
setUserName 368
setUserSound 433
setValue 411, 519, 522, 528, 533, 539, 547, 551, 555, 558, 561, 566, 576, 581, 586, 655, 671, 688
setValueAt 440
setValueLabel 315, 386
setValueText 315
setValueUncertain 412
setValueUnknown 411
setVerbosity 129
setVertGridSpacing 147
setVertSnapSpacing 147
setViewCenter 468
setViewPosition 142, 327
setVisible 138, 206, 235, 258, 325, 347, 367
setWaitCursor 366
setWarningColor 429
setWarningSound 433
setWebSupport 710
setWidth 475
setWidthScaleFactor 50, 150, 275

setWindowingMode 431
setWindowLocation 367
setX_Pos 287
setXpdfPath 435
setY_Pos 287
setZoomScale 141, 326
showAllConnections 226
showCancelButton 241
showComment 49, 103
showConfirmDialog 719
showConnections 291
showCreateDialog 361
showDemoMessage 726
ShowHS_connections 425
showInputDialog 719, 720
showMessageDialog 718, 719
showOpenDialog 361
showOptionDialog 720
SI 653
size 63, 92, 662, 666, 676, 680
snapCorba 358
SnapPreferences 423
snapshotView 325
sort 93, 440, 662, 666, 676, 680
sortByColumn 441
sortParallelRealArrays 693
SOUND_BEEP 422
SOUND_OFF 422
SOUND_ON 422
SpecialConnectionData 120, 121
splitPipe 47
sqrt 673, 692
SQUARE 272
STATIC_CONNECTOR 245
store 139, 294, 307, 316, 333, 475
storeComponent 139
storeDisplayBean 451
storeDrawnComponent 51
storeSettings 642
storeState 87, 101, 102, 124, 125, 168, 187, 223, 292, 306, 315, 334, 475, 482, 488, 500, 512, 693
storeUserConstant 199
storeUserFunction 200
storeUserVariable 199
submitModel 634
subtract 673, 691
supportsCustomEditor 520, 524, 528, 533, 540, 547, 551, 577, 582, 587

T

tableChanged 440
TableSorter 439
TextAnnotation 512
THEORYMAN 32
Time 696
toArray 63, 93, 663, 667, 677, 681
toFront 328, 382
toFrontDrawnViewTab 359
TOLERANCE 626
toLoadString 689
TOOL_CONNECT 448
TOOL_INSERT 448
TOOL_INTERACTIVE 448
TOOL_NONE 447
TOOL_PAN 447
TOOL_SELECT 447
TOOL_ZOOM 448
Toolbox 449
ToolboxAction 615
toolChanged 465, 617
TOP 273
toString 18, 52, 58, 64, 73, 79, 94, 121, 129, 138, 163, 176, 185, 246, 276, 306, 412, 477, 597, 604, 608, 610, 613, 619, 655, 671, 689, 726
trace 108
transformPoint 468
translate 305, 335, 499
translateConnectionToScreen 292
translatePointToScreen 292
TRIANGLE 271
trimToSize 663, 667, 677, 681
TYPE_CODE_PLUGIN 640
TYPE_FEATURE_PLUGIN 640
typeIsConnector 246
typeIsCrossflow 247
typeIsDropZone 246
typeIsInlet 247
typeIsOutlet 247
typeIsStatic 247

U

undo 454, 456, 458, 460
UndoableConnect 453
UndoableDelete 455, 456
UndoableDisconnect 458
UndoableInsert 459
undockView 141
UNIT_NAMES 653
unitsChanged 207, 414
UNKNOWN 550
Unknown 670, 686
unlockToString 687
UP 273
update 209
updateAvailable 712
updateComponentList 226
updateLabel 544, 575
updatePlaybackButtons 373
Updater 209
updateSelection 227
updateToolbars 327
updateValue 576
updateVersion 18, 107
UseExecServer 425
UserDefinedConstant 162
UserDefinedFunction 172
UserDefinedValue 179, 180
UserDefinedVariable 184
userDisconnect 73
UserErrorMsg 379
UserErrorSound 379
USERSMAN 32

V

validate 105, 177, 302, 500
validateAllComponents 36
ValidationOptions 123
ValidationTest 127
value 670
valueChanged 236
ValueOutOfRangeException 698, 699
VERBOSITY_HIGH 127
VERBOSITY_LOW 127
VERBOSITY_QUIET 127
verifyBeansAvailable 451
Version 724
ViewComponent 137

W

WarningMsg 378
WarningSound 379
WINDOW_MULTIPLE 422
WINDOW_NAMES 422
WINDOW_SINGLE 422
WINDOW_VALUES 422
write 201
writeArrayLoadValue 113, 114
writeBlockParams 6, 77, 87, 124
writeConstantTable 157
writeDescription 156
writeExportNote 156
writeMetrics 727
writeMuxLoadArray 112, 113
writeName 21, 156
writeSP 112
writeTableHeader 157
writeTableLine 157

Z

ZoomablePanel 464
ZoomMouseHandler 619
zoomToFit 469