---

jgsl.view.swing

Class AboutDialog

```
java.lang.Object
  └java.awt.Component
      └java.awt.Container
          └java.awt.Window
              └java.awt.Dialog
                  └javax.swing.JDialog
                      └jgsl.view.swing.AboutDialog
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

---

```
public class AboutDialog
```

extends javax.swing.JDialog

See Also:

Serialized Form

Field Summary

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

Fields inherited from interface javax.swing.WindowConstants
DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, EXIT_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

AboutDialog()

Method Summary

static void	**main**(java.lang.String[] args)

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, getTitle, hide, isModal, isResizable, isUndecorated, setModal, setResizable, setTitle, setUndecorated, show

Methods inherited from class java.awt.Window
addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor,

setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, toBack, toFront

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, transferFocusBackward, transferFocusDownCycle, validate

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable, dispatchEvent, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent,

hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object

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

Constructor Detail

AboutDialog

```
public AboutDialog()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
```

---

jgsl.model

Class AbstractStatement

```
java.lang.Object
  └ jgsl.model.AbstractStatement
```

All Implemented Interfaces:

Statement

```
public abstract class AbstractStatement
```

extends java.lang.Object
implements Statement

Abstract base class for all Statement interface implementations.

Version:

$Id: AbstractStatement.java,v 1.2 2005/05/16 00:54:17 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

AbstractStatement()

Method Summary

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

Methods inherited from interface jgsl.model.Statement
getJava, getType, setJGSL

Constructor Detail

AbstractStatement

```
public AbstractStatement()
```

---

jgsl.model

Interface Argument

All Known Implementing Classes:

JGSLColor, JGSLDouble, JGSLInteger, JGSLString

```
public interface Argument
```

The name of a script argument.

Version:

$Id: Argument.java,v 1.2 2005/05/16 00:54:17 zenarchitect Exp $

Author:

zenarchitect

Method Summary

java.lang.String	**getName**() Get the name of the argument

Method Detail

getName

```
java.lang.String getName()
```

Get the name of the argument

Returns:

String containing the name

---

jgsl.model

Class Assignment

```
java.lang.Object
  └ jgsl.model.Assignment
```

All Implemented Interfaces:

java.io.Serializable, Statement

```
public class Assignment
```

extends java.lang.Object
implements Statement, java.io.Serializable

An Assignment is a statement in which the value of one attribute is assigned to another via the "=" operator.

Version:

$Id: Assignment.java,v 1.2 2005/05/16 00:54:17 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

`Assignment``(java.lang.String lhs, java.lang.String rhs)`

Constructs an instance with the left-hand side and right-hand side arguments of the assignment statement

Method Summary

java.lang.String	**getJava**() This method returns the Java language equivalent of the JGSL statement.
java.lang.String	**getLhs**()
java.lang.String	**getRhs**()
java.lang.String	**getType**() Return the type of statement.
void	**setJGSL**(java.lang.String jgsl) Set the JGSL statement body
void	**setLhs**(java.lang.String lhs)
void	**setRhs**(java.lang.String rhs)

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

Constructor Detail

Assignment

```
public Assignment(java.lang.String lhs,
                  java.lang.String rhs)
```

Constructs an instance with the left-hand side and right-hand side arguments of the assignment statement

Parameters:

`lhs` - Left-hand side of the assignment
`rhs` - Right-hand side of the assignment

Method Detail

getJava

```
public java.lang.String getJava()
```

This method returns the Java language equivalent of the JGSL statement.

Specified by:
`getJava` in interface `Statement`

Returns:
Java language statement from the JGSL

---

getLhs

```
public java.lang.String getLhs()
```

---

getRhs

```
public java.lang.String getRhs()
```

---

getType

```
public java.lang.String getType()
```

Return the type of statement. The String form of the class name.

Specified by:
`getType` in interface `Statement`

---

setJGSL

```
public void setJGSL(java.lang.String jgsl)
```

Set the JGSL statement body

Specified by:
`setJGSL` in interface `Statement`

---

setLhs

```
public void setLhs(java.lang.String lhs)
```

---

setRhs

```
public void setRhs(java.lang.String rhs)
```

---

jgsl.view.swing

Class BaseFrame

```
java.lang.Object
  └java.awt.Component
      └java.awt.Container
          └java.awt.Window
              └java.awt.Frame
                  └javax.swing.JFrame
                      └jgsl.view.swing.BaseFrame
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

---

```
public class BaseFrame
```

extends javax.swing.JFrame

The BaseFrame class is used as a template to generate the JGSL compiled script class. This class is loaded by Javassist and renamed to the script name plus a package specification. The generated class is writting to the user's home directory + ".jgsl/cache" as the full path.

Version:

$Id: BaseFrame.java,v 1.7 2005/05/21 01:42:11 zenarchitect Exp $

Author:

Joe Chavez

See Also:

Serialized Form

Field Summary

static boolean	**DEBUG**

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

Fields inherited from interface javax.swing.WindowConstants
DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

BaseFrame()	Constructs a new frame that is initially invisible.

Method Summary

static void	**main**(java.lang.String[] args)
void	**paint**(java.awt.Graphics g) Paints the container.

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, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, 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, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, print, printComponents, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, transferFocusBackward, transferFocusDownCycle, validate

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable, dispatchEvent, enable, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list, list,

list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object

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

Methods inherited from interface java.awt.MenuContainer

getFont, postEvent

Field Detail

DEBUG

```
public static boolean DEBUG
```

Constructor Detail

BaseFrame

```
public BaseFrame()
          throws java.awt.HeadlessException
```

Constructs a new frame that is initially invisible.

This constructor sets the component's locale property to the value returned by `JComponent.getDefaultLocale`.

Throws:

`java.awt.HeadlessException` - if GraphicsEnvironment.isHeadless() returns true.

See Also:

`GraphicsEnvironment.isHeadless()`, `Component.setSize(int, int)`, `Component.setVisible(boolean)`, `JComponent.getDefaultLocale()`

Method Detail

main

```
public static void main(java.lang.String[] args)
```

---

paint

```
public void paint(java.awt.Graphics g)
```

Paints the container. This forwards the paint to any lightweight components that are children of this container. If this method is reimplemented, super.paint(g) should be called so that lightweight components are properly rendered. If a child component is entirely clipped by the current clipping setting in g, paint() will not be forwarded to that child.

Overrides:

`paint` in class `java.awt.Container`

Parameters:

`g` - the specified Graphics window

See Also:

`Component.update(java.awt.Graphics)`

---

jgsl.model

Class Command

```
java.lang.Object
  └ jgsl.model.Command
```

All Implemented Interfaces:

java.io.Serializable, Statement

```
public class Command
```

extends java.lang.Object
implements Statement, java.io.Serializable

A command statement is a JGSL command that performs a graphics operation.

Version:

$Id: Command.java,v 1.3 2005/05/16 00:54:17 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

`Command(java.lang.String name)`
`Command(java.lang.String name, java.util.ArrayList<Argument> parameters)`

Command(java.lang.String name, java.util.ArrayList<Argument> attributes, java.util.ArrayList<Argument> parameters)

Method Summary

java.lang.String	**getJava**() This method returns the Java language equivalent of the JGSL statement.
java.lang.String	**getType**() Return the type of statement.
void	**setJGSL**(java.lang.String jgsl) Set the JGSL statement body
void	**setName**(java.lang.String name)

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

Constructor Detail

Command

```
public Command(java.lang.String name)
```

Command

```
public Command(java.lang.String name,
               java.util.ArrayList<Argument> parameters)
```

Command

```
public Command(java.lang.String name,
               java.util.ArrayList<Argument> attributes,
               java.util.ArrayList<Argument> parameters)
```

Method Detail

getJava

```
public java.lang.String getJava()
```

This method returns the Java language equivalent of the JGSL statement.

Specified by:
`getJava` in interface `Statement`

Returns:
Java language statement from the JGSL

---

getType

```
public java.lang.String getType()
```

Return the type of statement. The String form of the class name.

Specified by:
`getType` in interface `Statement`

---

setJGSL

```
public void setJGSL(java.lang.String jgsl)
```

Set the JGSL statement body

Specified by:

setJGSL in interface Statement

---

setName

```
public void setName(java.lang.String name)
```

---

jgsl.model

Enum Commands

```
java.lang.Object
  └java.lang.Enum<Commands>
      └jgsl.model.Commands
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<Commands>

```
public enum Commands
```

extends java.lang.Enum<Commands>

Enum of all possible commands and their corresponding command template and java code representation.

Version:

$Id: Commands.java,v 1.3 2005/05/21 01:42:07 zenarchitect Exp $

Author:

zenarchitect

Enum Constant Summary

ARC
CANVAS
CIRCLE
CLEAR
DEBUG
DRAW
ELIPSE
ERROR

LINE
LOG
POLYGON
RECTANGLE
SQUARE
TEXT
WAIT
WARNING

Method Summary

java.lang.String	**getCommandTemplate**()
java.lang.String	**getFormattedCommand**(java.util.ArrayList<Argument> attributes, java.util.ArrayList<Argument> parameters)
java.lang.String	**getName**()
static Commands	**valueOf**(java.lang.String name) Returns the enum constant of this type with the specified name.
static Commands[]	**values**() Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

ARC

public static final Commands **ARC**

CANVAS

public static final Commands **CANVAS**

CIRCLE

public static final Commands **CIRCLE**

CLEAR

public static final Commands **CLEAR**

DEBUG

public static final Commands **DEBUG**

DRAW

public static final Commands **DRAW**

ELIPSE

public static final Commands **ELIPSE**

ERROR

public static final Commands **ERROR**

LINE

public static final Commands **LINE**

LOG

public static final Commands **LOG**

POLYGON

public static final Commands **POLYGON**

RECTANGLE

public static final Commands **RECTANGLE**

SQUARE

public static final Commands **SQUARE**

TEXT

public static final Commands **TEXT**

WAIT

public static final Commands **WAIT**

WARNING

public static final Commands **WARNING**

Method Detail

getCommandTemplate

```
public java.lang.String getCommandTemplate()
```

---

getFormattedCommand

```
public java.lang.String getFormattedCommand(java.util.ArrayList<Argument> attributes,
                                            java.util.ArrayList<Argument> parameters)
```

---

getName

```
public java.lang.String getName()
```

---

valueOf

```
public static Commands valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
`name` - the name of the enum constant to be returned.
Returns:
the enum constant with the specified name
Throws:
`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

---

values

```
public static final Commands[] values()
```

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```
for(Commands c : Commands.values())
        System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they're declared

---

---

jgsl.model

Class Declaration

```
java.lang.Object
  └ jgsl.model.Declaration
```

All Implemented Interfaces:

java.io.Serializable, Statement

```
public class Declaration
```

extends java.lang.Object
implements Statement, java.io.Serializable

A Declaration statement is one that contains the declaration of a script variable.

Version:

$Id: Declaration.java,v 1.2 2005/05/16 00:54:18 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

Declaration(java.lang.String type, java.lang.String identifier, java.lang.String value) Create a declaration with a given type, identifier and initial value

Method Summary

java.lang.String	**getJava**() This method returns the Java language equivalent of the JGSL statement.

java.lang.String	**getType**() Return the type of statement.
void	**setJGSL**(java.lang.String jgsl) Set the JGSL statement body

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

Constructor Detail

Declaration

```
public Declaration(java.lang.String type,
                   java.lang.String identifier,
                   java.lang.String value)
```

Create a declaration with a given type, identifier and initial value

Parameters:

`type` - type of the declaration
`identifier` - script identifier
`value` - initial value

Method Detail

getJava

```
public java.lang.String getJava()
```

This method returns the Java language equivalent of the JGSL statement.

Specified by:

`getJava` in interface `Statement`

Returns:

Java language statement from the JGSL

---

getType

```
public java.lang.String getType()
```

Return the type of statement. The String form of the class name.

Specified by:
`getType` in interface `Statement`

---

setJGSL

```
public void setJGSL(java.lang.String jgsl)
```

Set the JGSL statement body

Specified by:
`setJGSL` in interface `Statement`

---

jgsl.model

Class Documentation

```
java.lang.Object
  └ jgsl.model.Documentation
```

All Implemented Interfaces:

java.io.Serializable, Statement

```
public class Documentation
```

extends java.lang.Object
implements Statement, java.io.Serializable

A documentation statement is one that contains documentation of the JGSL script as written by the script author.

Version:

$Id: Documentation.java,v 1.3 2005/05/16 00:54:18 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

Documentation()

Method Summary

void	**addDoc**(java.lang.String doc)

java.lang.String	**getJava**() This method returns the Java language equivalent of the JGSL statement.
java.lang.String	**getType**() Return the type of statement.
void	**setJGSL**(java.lang.String jgsl) Set the JGSL statement body

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

Constructor Detail

Documentation

```
public Documentation()
```

Method Detail

addDoc

```
public void addDoc(java.lang.String doc)
```

---

getJava

```
public java.lang.String getJava()
```

This method returns the Java language equivalent of the JGSL statement.

Specified by:
getJava in interface Statement

Returns:

Java language statement from the JGSL

---

getType

```
public java.lang.String getType()
```

Return the type of statement. The String form of the class name.

Specified by:
`getType` in interface `Statement`

---

setJGSL

```
public void setJGSL(java.lang.String jgsl)
```

Set the JGSL statement body

Specified by:
`setJGSL` in interface `Statement`

---

jgsl.util

Class GifEncoder

```
java.lang.Object
  └jgsl.util.ImageEncoder
      └jgsl.util.GifEncoder
```

All Implemented Interfaces:

java.awt.image.ImageConsumer

```
public class GifEncoder
```

extends ImageEncoder

TODO - write java docs

Version:

$Id: GifEncoder.java,v 1.1 2005/05/21 01:42:10 zenarchitect Exp $

Author:

$Author: zenarchitect $

Field Summary

Fields inherited from interface java.awt.image.ImageConsumer
COMPLETESCANLINES, IMAGEABORTED, IMAGEERROR, RANDOMPIXELORDER, SINGLEFRAME, SINGLEFRAMEDONE, SINGLEPASS, STATICIMAGEDONE, TOPDOWNLEFTRIGHT

Constructor Summary

GifEncoder(java.awt.Image img, java.io.OutputStream out)

GifEncoder(java.awt.Image img, java.io.OutputStream out, boolean interlace)
GifEncoder(java.awt.image.ImageProducer prod, java.io.OutputStream out)
GifEncoder(java.awt.image.ImageProducer prod, java.io.OutputStream out, boolean interlace)

Method Summary

Methods inherited from class jgsl.util.ImageEncoder
encode, imageComplete, setColorModel, setDimensions, setHints, setPixels, setPixels, setProperties

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

Constructor Detail

GifEncoder

```
public GifEncoder(java.awt.Image img,
                  java.io.OutputStream out)
           throws java.io.IOException
```

Throws:

java.io.IOException

GifEncoder

```
public GifEncoder(java.awt.Image img,
                  java.io.OutputStream out,
                  boolean interlace)
           throws java.io.IOException
```

Throws:

java.io.IOException

GifEncoder

```
public GifEncoder(java.awt.image.ImageProducer prod,
                  java.io.OutputStream out)
           throws java.io.IOException
```

Throws:

java.io.IOException

GifEncoder

```
public GifEncoder(java.awt.image.ImageProducer prod,
                  java.io.OutputStream out,
                  boolean interlace)
           throws java.io.IOException
```

Throws:

java.io.IOException

---

jgsl.util

Class ImageEncoder

```
java.lang.Object
  └ jgsl.util.ImageEncoder
```

All Implemented Interfaces:

java.awt.image.ImageConsumer

Direct Known Subclasses:

GifEncoder

```
public abstract class ImageEncoder
```

extends java.lang.Object
implements java.awt.image.ImageConsumer

TODO - write java docs

Version:

$Id: ImageEncoder.java,v 1.1 2005/05/21 01:42:10 zenarchitect Exp $

Author:

$Author: zenarchitect $

Field Summary

Fields inherited from interface java.awt.image.ImageConsumer
COMPLETESCANLINES, IMAGEABORTED, IMAGEERROR, RANDOMPIXELORDER, SINGLEFRAME, SINGLEFRAMEDONE, SINGLEPASS, STATICIMAGEDONE, TOPDOWNLEFTRIGHT

Constructor Summary

ImageEncoder(java.awt.Image img, java.io.OutputStream out)
ImageEncoder(java.awt.image.ImageProducer producer, java.io.OutputStream out)

Method Summary

void	**encode**()
void	**imageComplete**(int status)
void	**setColorModel**(java.awt.image.ColorModel model)
void	**setDimensions**(int width, int height)
void	**setHints**(int hintflags)
void	**setPixels**(int x, int y, int w, int h, java.awt.image.ColorModel model, byte[] pixels, int off, int scansize)
void	**setPixels**(int x, int y, int w, int h, java.awt.image.ColorModel model, int[] pixels, int off, int scansize)
void	**setProperties**(java.util.Hashtable<?,?> props)

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

Constructor Detail

ImageEncoder

```
public ImageEncoder(java.awt.Image img,
                    java.io.OutputStream out)
             throws java.io.IOException
```

Throws:

`java.io.IOException`

ImageEncoder

```
public ImageEncoder(java.awt.image.ImageProducer producer,
                    java.io.OutputStream out)
             throws java.io.IOException
```

Throws:

`java.io.IOException`

Method Detail

encode

```
public void encode()
            throws java.io.IOException
```

Throws:

`java.io.IOException`

---

imageComplete

```
public void imageComplete(int status)
```

Specified by:

`imageComplete` in interface `java.awt.image.ImageConsumer`

---

setColorModel

```
public void setColorModel(java.awt.image.ColorModel model)
```

Specified by:
`setColorModel` in interface `java.awt.image.ImageConsumer`

---

setDimensions

```
public void setDimensions(int width,
                          int height)
```

Specified by:
`setDimensions` in interface `java.awt.image.ImageConsumer`

---

setHints

```
public void setHints(int hintflags)
```

Specified by:
`setHints` in interface `java.awt.image.ImageConsumer`

---

setPixels

```
public void setPixels(int x,
                      int y,
                      int w,
                      int h,
                      java.awt.image.ColorModel model,
                      byte[] pixels,
                      int off,
                      int scansize)
```

Specified by:
`setPixels` in interface `java.awt.image.ImageConsumer`

---

setPixels

```
public void setPixels(int x,
                      int y,
                      int w,
                      int h,
                      java.awt.image.ColorModel model,
                      int[] pixels,
                      int off,
                      int scansize)
```

Specified by:

`setPixels` in interface `java.awt.image.ImageConsumer`

---

setProperties

```
public void setProperties(java.util.Hashtable<?,?> props)
```

Specified by:

`setProperties` in interface `java.awt.image.ImageConsumer`

---

jgsl.io

Class ImageFileFilter

```
java.lang.Object
  └javax.swing.filechooser.FileFilter
      └jgsl.io.ImageFileFilter
```

```
public class ImageFileFilter
```

extends javax.swing.filechooser.FileFilter

FileFilter for Image files

Version:

$Id: ImageFileFilter.java,v 1.1 2005/05/21 01:42:07 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`ImageFileFilter()`

Method Summary

boolean	`accept(java.io.File f)`
java.lang.String	`getDescription()`

Methods inherited from class java.lang.Object

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

Constructor Detail

ImageFileFilter

public **ImageFileFilter**()

Method Detail

accept

public boolean **accept**(java.io.File f)

Specified by:
accept in class javax.swing.filechooser.FileFilter

---

getDescription

public java.lang.String **getDescription**()

Specified by:
getDescription in class javax.swing.filechooser.FileFilter

---

jgsl.io

Class JARFileFilter

```
java.lang.Object
  └javax.swing.filechooser.FileFilter
      └jgsl.io.JARFileFilter
```

```
public class JARFileFilter
```

extends javax.swing.filechooser.FileFilter

FileFilter for .JAR files

Version:

$Id: JARFileFilter.java,v 1.2 2005/05/16 00:54:16 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`JARFileFilter()`

Method Summary

boolean	`accept(java.io.File f)`
java.lang.String	`getDescription()`

Methods inherited from class java.lang.Object

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

Constructor Detail

JARFileFilter

public **JARFileFilter**()

Method Detail

accept

public boolean **accept**(java.io.File f)

Specified by:
accept in class javax.swing.filechooser.FileFilter

---

getDescription

public java.lang.String **getDescription**()

Specified by:
getDescription in class javax.swing.filechooser.FileFilter

---

jgsl.util

Class JarPackager

```
java.lang.Object
  └ jgsl.util.JarPackager
```

```
public class JarPackager
```

extends java.lang.Object

Create an executable JAR file for a JGSL script Java class.

Author:

jchavez

Constructor Summary

`JarPackager()`

Method Summary

`static void`	`makeJar(java.io.File jarFileName, java.lang.String classFileName, java.lang.String className)` Create a jar file for JGSL distribution.

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

Constructor Detail

JarPackager

```
public JarPackager()
```

Method Detail

makeJar

```
public static void makeJar(java.io.File jarFileName,
                           java.lang.String classFileName,
                           java.lang.String className)
                    throws JarPackagerException
```

Create a jar file for JGSL distribution. The className parameter will be used to set the main class attribute.

Main-Class: className

Parameters:

`jarFileName` - Name of JAR to create
`classFileName` - Full path to the class file to add to the jar
`className` - Name of the class with full package specification. The "." will be replaced with "/".

Throws:

`JarPackagerException`

---

jgsl.util

Class JarPackagerException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └jgsl.util.JarPackagerException
```

All Implemented Interfaces:

java.io.Serializable

```
public class JarPackagerException
```

extends java.lang.Exception

Report JarPackageer exceptions.

Author:

jchavez

See Also:

Serialized Form

Constructor Summary

JarPackagerException(java.lang.String string)

Method Summary

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, wait, wait, wait

Constructor Detail

JarPackagerException

```
public JarPackagerException(java.lang.String string)
```

---

jgsl.parser

Class JavaCharStream

```
java.lang.Object
  └ jgsl.parser.JavaCharStream
```

```
public class JavaCharStream
```

extends java.lang.Object

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (with java-like unicode escape processing).

Field Summary

int	**bufpos**
static boolean	**staticFlag**

Constructor Summary

JavaCharStream(java.io.InputStream dstream)
JavaCharStream(java.io.InputStream dstream, int startline, int startcolumn)
JavaCharStream(java.io.InputStream dstream, int startline, int startcolumn, int buffersize)
JavaCharStream(java.io.Reader dstream)

JavaCharStream(java.io.Reader dstream, int startline, int startcolumn)
JavaCharStream(java.io.Reader dstream, int startline, int startcolumn, int buffersize)

Method Summary

void	**adjustBeginLineColumn**(int newLine, int newCol) Method to adjust line and column numbers for the start of a token.
void	**backup**(int amount)
char	**BeginToken**()
void	**Done**()
int	**getBeginColumn**()
int	**getBeginLine**()
int	**getEndColumn**()
int	**getEndLine**()
java.lang.String	**GetImage**()
char[]	**GetSuffix**(int len)
char	**readChar**()
void	**ReInit**(java.io.InputStream dstream)

void	**ReInit**(java.io.InputStream dstream, int startline, int startcolumn)
void	**ReInit**(java.io.InputStream dstream, int startline, int startcolumn, int buffersize)
void	**ReInit**(java.io.Reader dstream)
void	**ReInit**(java.io.Reader dstream, int startline, int startcolumn)
void	**ReInit**(java.io.Reader dstream, int startline, int startcolumn, int buffersize)

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

Field Detail

bufpos

public int **bufpos**

---

staticFlag

public static final boolean **staticFlag**

See Also:

Constant Field Values

Constructor Detail

JavaCharStream

```
public JavaCharStream(java.io.InputStream dstream)
```

JavaCharStream

```
public JavaCharStream(java.io.InputStream dstream,
                      int startline,
                      int startcolumn)
```

JavaCharStream

```
public JavaCharStream(java.io.InputStream dstream,
                      int startline,
                      int startcolumn,
                      int buffersize)
```

JavaCharStream

```
public JavaCharStream(java.io.Reader dstream)
```

JavaCharStream

```
public JavaCharStream(java.io.Reader dstream,
                      int startline,
                      int startcolumn)
```

JavaCharStream

```
public JavaCharStream(java.io.Reader dstream,
```

```
                      int startline,
                      int startcolumn,
                      int buffersize)
```

Method Detail

adjustBeginLineColumn

```
public void adjustBeginLineColumn(int newLine,
                                  int newCol)
```

Method to adjust line and column numbers for the start of a token.

---

backup

```
public void backup(int amount)
```

BeginToken

```
public char BeginToken()
               throws java.io.IOException
```

Throws:

`java.io.IOException`

Done

```
public void Done()
```

---

getBeginColumn

```
public int getBeginColumn()
```

---

getBeginLine

```
public int getBeginLine()
```

---

getEndColumn

```
public int getEndColumn()
```

---

getEndLine

```
public int getEndLine()
```

GetImage

```
public java.lang.String GetImage()
```

GetSuffix

```
public char[] GetSuffix(int len)
```

---

readChar

```
public char readChar()
             throws java.io.IOException
```

Throws:

java.io.IOException

ReInit

```
public void ReInit(java.io.InputStream dstream)
```

ReInit

```
public void ReInit(java.io.InputStream dstream,
                   int startline,
                   int startcolumn)
```

ReInit

```
public void ReInit(java.io.InputStream dstream,
                   int startline,
                   int startcolumn,
                   int buffersize)
```

ReInit

```
public void ReInit(java.io.Reader dstream)
```

ReInit

```
public void ReInit(java.io.Reader dstream,
                   int startline,
                   int startcolumn)
```

ReInit

```
public void ReInit(java.io.Reader dstream,
                   int startline,
                   int startcolumn,
                   int buffersize)
```

---

jgsl

Class JGSL

```
java.lang.Object
  └ jgsl.JGSL
```

```
public class JGSL
```

extends java.lang.Object

JGSL program main class. This is main entry point for executing the JGSL in both command line and GUI modes. This class configures 2 loggers using the Log4J API. The Log4J properties file is bundled with the jgsl.jar file and is located in the

```
jgsl/resources/jgsl_log.prop
```

file.

After the loggers are configured the control flow is passed to the ScriptEngine class.

Version:
$Id: JGSL.java,v 1.8 2005/05/16 00:54:23 zenarchitect Exp $
Author:
zenarchitect

Constructor Summary

JGSL()

Method Summary

static void	**main**(java.lang.String[] args)

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

Constructor Detail

JGSL

```
public JGSL()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
```

---

jgsl.parser

Class JGSL_Parser

```
java.lang.Object
  └ jgsl.parser.JGSL_Parser
```

All Implemented Interfaces:

JGSL_ParserConstants

```
public class JGSL_Parser
```

extends java.lang.Object
implements JGSL_ParserConstants

Version:

$Id: JGSL_Parser.jj,v 1.5 2005/05/01 01:48:23 zenarchitect Exp $

Author:

zenarchitect

Field Summary

Token	**jj_nt**
Token	**token**
JGSL_ParserTokenManager	**token_source**

Fields inherited from interface jgsl.parser.JGSL_ParserConstants

AND, ARC, ASSIGN, BACKGROUND, BANG, BEGIN, BLACK, BLUE, BORDER, CANVAS, CIRCLE, CLEAR, COLON, COLOR, COMMA, DARK_GRAY, DEBUG, DECIMAL_LITERAL, DECLARE, DEFAULT, DIGIT, DOC, DRAW, ELIPSE, ELSE, ELSEIF, END, EOF, EQ, ERROR, FALSE, FILL, FLOATING_POINT_LITERAL, FOREGROUND, FORMAL_COMMENT, GE, GRADIENT, GRAY, GREEN, GT, HEX_LITERAL, IDENTIFIER, IF, IN_FORMAL_COMMENT, IN_MULTI_LINE_COMMENT, IN_SINGLE_LINE_COMMENT, INTEGER_LITERAL, JGSL, LE, LETTER, LIGHT_GRAY, LINE, LOG, LOOP, LPAREN, LT, MAGENTA, MINUS, MOD, MULTI_LINE_COMMENT, NE, NOT, OCTAL_LITERAL, OR, ORANGE, PINK, PLUS, POLYGON, READ, RECTANGLE, RED, REPEAT, RPAREN, SC_AND, SC_OR, SEMICOLON, SINGLE_LINE_COMMENT, SLASH, SQUARE, STAR, STRING_LITERAL, TEXT, THEN, tokenImage, TRUE, VERSION, WAIT, WARNING, WHITE, WRITE, YELLOW

Constructor Summary

JGSL_Parser(java.io.InputStream stream)
JGSL_Parser(JGSL_ParserTokenManager tm)
JGSL_Parser(java.io.Reader stream)

Method Summary

java.util.ArrayList	**ArcAttributes**()
void	**Assignment**()
void	**Canvas**()
java.util.ArrayList	**CanvasAttributes**()
java.util.ArrayList	**CircleAttributes**()

void	**Clear**()
void	**Command**()
void	**Debug**()
void	**Declaration**()
void	**DeclareCanvas**()
void	**DeclareColor**()
void	**DeclareRGB**(Token type, Token id)
void	**DeclareStandardColor**(Token type, Token id)
void	**disable_tracing**()
void	**Documentation**()
void	**Draw**()
void	**DrawArc**()
java.util.ArrayList	**DrawAttributes**()
void	**DrawCircle**()
void	**DrawElipse**()
void	**DrawLine**()

void	**DrawPolygon**()
void	**DrawRectangle**()
void	**DrawShape**()
void	**DrawSquare**()
void	**DrawText**()
java.util.ArrayList	**ElipseAttributes**()
void	**enable_tracing**()
void	**Error**()
ParseException	**generateParseException**()
java.awt.Color	**GetColor**()
Token	**getNextToken**()
java.awt.Color	**GetRGB**()
JGSLScript	**getScript**()
java.awt.Color	**GetStandardColor**()
Token	**getToken**(int index)
java.util.ArrayList	**LineAttributes**()

void	**Log**()
static void	**main**(java.lang.String[] args)
void	**Message**()
void	**parseScript**() THE JGSL GRAMMAR STARTS HERE *
java.util.ArrayList	**PolygonAttributes**()
void	**PolygonParameters**(java.util.ArrayList parameters)
java.util.ArrayList	**RectangleAttributes**()
void	**ReInit**(java.io.InputStream stream)
void	**ReInit**(JGSL_ParserTokenManager tm)
void	**ReInit**(java.io.Reader stream)
void	**Script**()
void	**ScriptBody**()
java.util.ArrayList	**SquareAttributes**()
java.util.ArrayList	**TextAttributes**()
void	**Wait**()
void	**Warning**()

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

Field Detail

jj_nt

public Token **jj_nt**

---

token

public Token **token**

---

token_source

public JGSL_ParserTokenManager **token_source**

Constructor Detail

JGSL_Parser

public **JGSL_Parser**(java.io.InputStream stream)

JGSL_Parser

public **JGSL_Parser**(JGSL_ParserTokenManager tm)

JGSL_Parser

```
public JGSL_Parser(java.io.Reader stream)
```

Method Detail

ArcAttributes

```
public final java.util.ArrayList ArcAttributes()
                                          throws ParseException
```

Throws:
ParseException

Assignment

```
public final void Assignment()
                     throws ParseException
```

Throws:
ParseException

Canvas

```
public final void Canvas()
                  throws ParseException
```

Throws:
ParseException

CanvasAttributes

```
public final java.util.ArrayList CanvasAttributes()
```

throws ParseException

Throws:
ParseException

CircleAttributes

```
public final java.util.ArrayList CircleAttributes()
                                        throws ParseException
```

Throws:
ParseException

Clear

```
public final void Clear()
                throws ParseException
```

Throws:
ParseException

Command

```
public final void Command()
                 throws ParseException
```

Throws:
ParseException

Debug

```
public final void Debug()
```

```
                throws ParseException
```

Throws:

ParseException

Declaration

```
public final void Declaration()
                        throws ParseException
```

Throws:

ParseException

DeclareCanvas

```
public final void DeclareCanvas()
                         throws ParseException
```

Throws:

ParseException

DeclareColor

```
public final void DeclareColor()
                        throws ParseException
```

Throws:

ParseException

DeclareRGB

```
public final void DeclareRGB(Token type,
```

```
                                Token id)
                         throws ParseException
```

Throws:

ParseException

DeclareStandardColor

```
public final void DeclareStandardColor(Token type,
                                        Token id)
                                 throws ParseException
```

Throws:

ParseException

---

disable_tracing

```
public final void disable_tracing()
```

Documentation

```
public final void Documentation()
                           throws ParseException
```

Throws:

ParseException

Draw

```
public final void Draw()
                  throws ParseException
```

Throws:
ParseException

DrawArc

```
public final void DrawArc()
                  throws ParseException
```

Throws:
ParseException

DrawAttributes

```
public final java.util.ArrayList DrawAttributes()
                                        throws ParseException
```

Throws:
ParseException

DrawCircle

```
public final void DrawCircle()
                    throws ParseException
```

Throws:
ParseException

DrawElipse

```
public final void DrawElipse()
                    throws ParseException
```

Throws:
ParseException

DrawLine

```
public final void DrawLine()
                  throws ParseException
```

Throws:
ParseException

DrawPolygon

```
public final void DrawPolygon()
                    throws ParseException
```

Throws:
ParseException

DrawRectangle

```
public final void DrawRectangle()
                      throws ParseException
```

Throws:
ParseException

DrawShape

```
public final void DrawShape()
                   throws ParseException
```

Throws:
ParseException

DrawSquare

```
public final void DrawSquare()
                    throws ParseException
```

Throws:
ParseException

DrawText

```
public final void DrawText()
                  throws ParseException
```

Throws:
ParseException

ElipseAttributes

```
public final java.util.ArrayList ElipseAttributes()
                                          throws ParseException
```

Throws:
ParseException

---

enable_tracing

```
public final void enable_tracing()
```

Error

```
public final void Error()
                  throws ParseException
```

Throws:

ParseException

---

generateParseException

```
public ParseException generateParseException()
```

GetColor

```
public final java.awt.Color GetColor()
                             throws ParseException
```

Throws:

ParseException

---

getNextToken

```
public final Token getNextToken()
```

GetRGB

```
public final java.awt.Color GetRGB()
                            throws ParseException
```

Throws:

ParseException

---

getScript

```
public JGSLScript getScript()
```

GetStandardColor

```
public final java.awt.Color GetStandardColor()
                                    throws ParseException
```

Throws:

ParseException

---

getToken

```
public final Token getToken(int index)
```

LineAttributes

```
public final java.util.ArrayList LineAttributes()
                                      throws ParseException
```

Throws:

ParseException

Log

```
public final void Log()
```

```
                throws ParseException
```

Throws:
ParseException

---

main

```
public static void main(java.lang.String[] args)
                  throws ParseException
```

Throws:
ParseException

Message

```
public final void Message()
                   throws ParseException
```

Throws:
ParseException

---

parseScript

```
public final void parseScript()
                       throws ParseException
```

THE JGSL GRAMMAR STARTS HERE *

Throws:
ParseException

PolygonAttributes

```
public final java.util.ArrayList PolygonAttributes()
                                                 throws ParseException
```

Throws:
ParseException

PolygonParameters

```
public final void PolygonParameters(java.util.ArrayList parameters)
                             throws ParseException
```

Throws:
ParseException

RectangleAttributes

```
public final java.util.ArrayList RectangleAttributes()
                                                   throws ParseException
```

Throws:
ParseException

ReInit

```
public void ReInit(java.io.InputStream stream)
```

ReInit

```
public void ReInit(JGSL_ParserTokenManager tm)
```

ReInit

```
public void ReInit(java.io.Reader stream)
```

Script

```
public final void Script()
                  throws ParseException
```

Throws:

ParseException

ScriptBody

```
public final void ScriptBody()
                    throws ParseException
```

Throws:

ParseException

SquareAttributes

```
public final java.util.ArrayList SquareAttributes()
                                          throws ParseException
```

Throws:

ParseException

TextAttributes

```
public final java.util.ArrayList TextAttributes()
```

throws ParseException

Throws:
ParseException

Wait

```
public final void Wait()
                throws ParseException
```

Throws:
ParseException

Warning

```
public final void Warning()
                  throws ParseException
```

Throws:
ParseException

---

jgsl.parser

Interface JGSL_ParserConstants

All Known Implementing Classes:

JGSL_Parser, JGSL_ParserTokenManager

```
public interface JGSL_ParserConstants
```

Field Summary	
static int	**AND**
static int	**ARC**
static int	**ASSIGN**
static int	**BACKGROUND**
static int	**BANG**
static int	**BEGIN**
static int	**BLACK**
static int	**BLUE**
static int	**BORDER**
static int	**CANVAS**

static int	**CIRCLE**
static int	**CLEAR**
static int	**COLON**
static int	**COLOR**
static int	**COMMA**
static int	**DARK_GRAY**
static int	**DEBUG**
static int	**DECIMAL_LITERAL**
static int	**DECLARE**
static int	**DEFAULT**
static int	**DIGIT**
static int	**DOC**
static int	**DRAW**
static int	**ELIPSE**
static int	**ELSE**
static int	**ELSEIF**

static int	**END**
static int	**EOF**
static int	**EQ**
static int	**ERROR**
static int	**FALSE**
static int	**FILL**
static int	**FLOATING_POINT_LITERAL**
static int	**FOREGROUND**
static int	**FORMAL_COMMENT**
static int	**GE**
static int	**GRADIENT**
static int	**GRAY**
static int	**GREEN**
static int	**GT**
static int	**HEX_LITERAL**
static int	**IDENTIFIER**

static int	**IF**
static int	**IN_FORMAL_COMMENT**
static int	**IN_MULTI_LINE_COMMENT**
static int	**IN_SINGLE_LINE_COMMENT**
static int	**INTEGER_LITERAL**
static int	**JGSL**
static int	**LE**
static int	**LETTER**
static int	**LIGHT_GRAY**
static int	**LINE**
static int	**LOG**
static int	**LOOP**
static int	**LPAREN**
static int	**LT**
static int	**MAGENTA**
static int	**MINUS**

static int	**MOD**
static int	**MULTI_LINE_COMMENT**
static int	**NE**
static int	**NOT**
static int	**OCTAL_LITERAL**
static int	**OR**
static int	**ORANGE**
static int	**PINK**
static int	**PLUS**
static int	**POLYGON**
static int	**READ**
static int	**RECTANGLE**
static int	**RED**
static int	**REPEAT**
static int	**RPAREN**
static int	**SC_AND**

static int	**SC_OR**
static int	**SEMICOLON**
static int	**SINGLE_LINE_COMMENT**
static int	**SLASH**
static int	**SQUARE**
static int	**STAR**
static int	**STRING_LITERAL**
static int	**TEXT**
static int	**THEN**
static java.lang.String[]	**tokenImage**
static int	**TRUE**
static int	**VERSION**
static int	**WAIT**
static int	**WARNING**
static int	**WHITE**
static int	**WRITE**

static int	**YELLOW**

Field Detail

AND

```
static final int AND
```

See Also:

Constant Field Values

ARC

```
static final int ARC
```

See Also:

Constant Field Values

ASSIGN

```
static final int ASSIGN
```

See Also:

Constant Field Values

BACKGROUND

```
static final int BACKGROUND
```

See Also:

Constant Field Values

BANG

```
static final int BANG
```

See Also:

Constant Field Values

BEGIN

```
static final int BEGIN
```

See Also:

Constant Field Values

BLACK

```
static final int BLACK
```

See Also:

Constant Field Values

BLUE

```
static final int BLUE
```

See Also:

Constant Field Values

BORDER

```
static final int BORDER
```

See Also:

Constant Field Values

CANVAS

```
static final int CANVAS
```

See Also:

Constant Field Values

CIRCLE

```
static final int CIRCLE
```

See Also:

Constant Field Values

CLEAR

```
static final int CLEAR
```

See Also:

Constant Field Values

COLON

```
static final int COLON
```

See Also:

Constant Field Values

COLOR

```
static final int COLOR
```

See Also:

Constant Field Values

COMMA

```
static final int COMMA
```

See Also:

Constant Field Values

DARK_GRAY

```
static final int DARK_GRAY
```

See Also:

Constant Field Values

DEBUG

```
static final int DEBUG
```

See Also:

Constant Field Values

DECIMAL_LITERAL

```
static final int DECIMAL_LITERAL
```

See Also:
Constant Field Values

DECLARE

```
static final int DECLARE
```

See Also:
Constant Field Values

DEFAULT

```
static final int DEFAULT
```

See Also:
Constant Field Values

DIGIT

```
static final int DIGIT
```

See Also:
Constant Field Values

DOC

```
static final int DOC
```

See Also:
Constant Field Values

DRAW

```
static final int DRAW
```

See Also:
Constant Field Values

ELIPSE

```
static final int ELIPSE
```

See Also:
Constant Field Values

ELSE

```
static final int ELSE
```

See Also:
Constant Field Values

ELSEIF

```
static final int ELSEIF
```

See Also:
Constant Field Values

END

```
static final int END
```

See Also:

Constant Field Values

EOF

```
static final int EOF
```

See Also:

Constant Field Values

EQ

```
static final int EQ
```

See Also:

Constant Field Values

ERROR

```
static final int ERROR
```

See Also:

Constant Field Values

FALSE

```
static final int FALSE
```

See Also:

Constant Field Values

FILL

```
static final int FILL
```

See Also:

Constant Field Values

FLOATING_POINT_LITERAL

```
static final int FLOATING_POINT_LITERAL
```

See Also:

Constant Field Values

FOREGROUND

```
static final int FOREGROUND
```

See Also:

Constant Field Values

FORMAL_COMMENT

```
static final int FORMAL_COMMENT
```

See Also:

Constant Field Values

GE

```
static final int GE
```

See Also:

Constant Field Values

GRADIENT

```
static final int GRADIENT
```

See Also:

Constant Field Values

GRAY

```
static final int GRAY
```

See Also:

Constant Field Values

GREEN

```
static final int GREEN
```

See Also:

Constant Field Values

GT

```
static final int GT
```

See Also:

Constant Field Values

HEX_LITERAL

```
static final int HEX_LITERAL
```

See Also:

Constant Field Values

IDENTIFIER

```
static final int IDENTIFIER
```

See Also:

Constant Field Values

IF

```
static final int IF
```

See Also:

Constant Field Values

IN_FORMAL_COMMENT

```
static final int IN_FORMAL_COMMENT
```

See Also:

Constant Field Values

IN_MULTI_LINE_COMMENT

```
static final int IN_MULTI_LINE_COMMENT
```

See Also:

Constant Field Values

IN_SINGLE_LINE_COMMENT

```
static final int IN_SINGLE_LINE_COMMENT
```

See Also:

Constant Field Values

INTEGER_LITERAL

```
static final int INTEGER_LITERAL
```

See Also:

Constant Field Values

JGSL

```
static final int JGSL
```

See Also:

Constant Field Values

LE

```
static final int LE
```

See Also:

Constant Field Values

LETTER

```
static final int LETTER
```

See Also:

Constant Field Values

LIGHT_GRAY

```
static final int LIGHT_GRAY
```

See Also:

Constant Field Values

LINE

```
static final int LINE
```

See Also:

Constant Field Values

LOG

```
static final int LOG
```

See Also:

Constant Field Values

LOOP

```
static final int LOOP
```

See Also:

Constant Field Values

LPAREN

```
static final int LPAREN
```

See Also:

Constant Field Values

LT

```
static final int LT
```

See Also:

Constant Field Values

MAGENTA

```
static final int MAGENTA
```

See Also:

Constant Field Values

MINUS

```
static final int MINUS
```

See Also:
Constant Field Values

MOD

```
static final int MOD
```

See Also:
Constant Field Values

MULTI_LINE_COMMENT

```
static final int MULTI_LINE_COMMENT
```

See Also:
Constant Field Values

NE

```
static final int NE
```

See Also:
Constant Field Values

NOT

```
static final int NOT
```

See Also:
Constant Field Values

OCTAL_LITERAL

```
static final int OCTAL_LITERAL
```

See Also:
Constant Field Values

OR

```
static final int OR
```

See Also:
Constant Field Values

ORANGE

```
static final int ORANGE
```

See Also:
Constant Field Values

PINK

```
static final int PINK
```

See Also:
Constant Field Values

PLUS

```
static final int PLUS
```

See Also:

Constant Field Values

POLYGON

```
static final int POLYGON
```

See Also:

Constant Field Values

READ

```
static final int READ
```

See Also:

Constant Field Values

RECTANGLE

```
static final int RECTANGLE
```

See Also:

Constant Field Values

RED

```
static final int RED
```

See Also:

Constant Field Values

REPEAT

```
static final int REPEAT
```

See Also:

Constant Field Values

RPAREN

```
static final int RPAREN
```

See Also:

Constant Field Values

SC_AND

```
static final int SC_AND
```

See Also:

Constant Field Values

SC_OR

```
static final int SC_OR
```

See Also:

Constant Field Values

SEMICOLON

```
static final int SEMICOLON
```

See Also:
Constant Field Values

SINGLE_LINE_COMMENT

```
static final int SINGLE_LINE_COMMENT
```

See Also:
Constant Field Values

SLASH

```
static final int SLASH
```

See Also:
Constant Field Values

SQUARE

```
static final int SQUARE
```

See Also:
Constant Field Values

STAR

```
static final int STAR
```

See Also:

Constant Field Values

STRING_LITERAL

```
static final int STRING_LITERAL
```

See Also:

Constant Field Values

TEXT

```
static final int TEXT
```

See Also:

Constant Field Values

THEN

```
static final int THEN
```

See Also:

Constant Field Values

---

tokenImage

```
static final java.lang.String[] tokenImage
```

TRUE

```
static final int TRUE
```

See Also:

Constant Field Values

VERSION

```
static final int VERSION
```

See Also:

Constant Field Values

WAIT

```
static final int WAIT
```

See Also:

Constant Field Values

WARNING

```
static final int WARNING
```

See Also:

Constant Field Values

WHITE

```
static final int WHITE
```

See Also:

Constant Field Values

WRITE

```
static final int WRITE
```

See Also:

Constant Field Values

YELLOW

```
static final int YELLOW
```

See Also:

Constant Field Values

---

jgsl.parser

Class JGSL_ParserTokenManager

```
java.lang.Object
  └ jgsl.parser.JGSL_ParserTokenManager
```

All Implemented Interfaces:

JGSL_ParserConstants

```
public class JGSL_ParserTokenManager
```

extends java.lang.Object
implements JGSL_ParserConstants

Field Summary	
java.io.PrintStream	**debugStream**
static int[]	**jjnewLexState**
static java.lang.String[]	**jjstrLiteralImages**
static java.lang.String[]	**lexStateNames**

Fields inherited from interface jgsl.parser.JGSL_ParserConstants

AND, ARC, ASSIGN, BACKGROUND, BANG, BEGIN, BLACK, BLUE, BORDER, CANVAS, CIRCLE, CLEAR, COLON, COLOR, COMMA, DARK_GRAY, DEBUG, DECIMAL_LITERAL, DECLARE, DEFAULT, DIGIT, DOC, DRAW, ELIPSE, ELSE, ELSEIF, END, EOF, EQ, ERROR, FALSE, FILL, FLOATING_POINT_LITERAL, FOREGROUND, FORMAL_COMMENT, GE, GRADIENT, GRAY, GREEN, GT, HEX_LITERAL, IDENTIFIER, IF, IN_FORMAL_COMMENT, IN_MULTI_LINE_COMMENT, IN_SINGLE_LINE_COMMENT, INTEGER_LITERAL, JGSL, LE, LETTER, LIGHT_GRAY, LINE, LOG, LOOP, LPAREN, LT, MAGENTA, MINUS, MOD, MULTI_LINE_COMMENT, NE, NOT, OCTAL_LITERAL, OR, ORANGE, PINK, PLUS, POLYGON, READ, RECTANGLE, RED, REPEAT, RPAREN, SC_AND, SC_OR, SEMICOLON, SINGLE_LINE_COMMENT, SLASH, SQUARE, STAR, STRING_LITERAL, TEXT, THEN, tokenImage, TRUE, VERSION, WAIT, WARNING, WHITE, WRITE, YELLOW

Constructor Summary

JGSL_ParserTokenManager(JavaCharStream stream)
JGSL_ParserTokenManager(JavaCharStream stream, int lexState)

Method Summary

Token	**getNextToken**()
void	**ReInit**(JavaCharStream stream)
void	**ReInit**(JavaCharStream stream, int lexState)
void	**setDebugStream**(java.io.PrintStream ds)
void	**SwitchTo**(int lexState)

Methods inherited from class java.lang.Object

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

Field Detail

debugStream

public java.io.PrintStream **debugStream**

---

jjnewLexState

public static final int[] **jjnewLexState**

---

jjstrLiteralImages

public static final java.lang.String[] **jjstrLiteralImages**

---

lexStateNames

public static final java.lang.String[] **lexStateNames**

Constructor Detail

JGSL_ParserTokenManager

public **JGSL_ParserTokenManager**(JavaCharStream stream)

JGSL_ParserTokenManager

```
public JGSL_ParserTokenManager(JavaCharStream stream,
                               int lexState)
```

Method Detail

getNextToken

```
public Token getNextToken()
```

ReInit

```
public void ReInit(JavaCharStream stream)
```

ReInit

```
public void ReInit(JavaCharStream stream,
                   int lexState)
```

---

setDebugStream

```
public void setDebugStream(java.io.PrintStream ds)
```

SwitchTo

```
public void SwitchTo(int lexState)
```

---

jgsl.model

Class JGSLColor

```
java.lang.Object
  └ jgsl.model.JGSLColor
```

All Implemented Interfaces:

Argument, Type, Value

```
public class JGSLColor
```

extends java.lang.Object
implements Type, Value, Argument

Declare an instance of a color type.

Version:

$Id: JGSLColor.java,v 1.3 2005/05/21 01:42:07 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

JGSLColor(java.lang.String name, java.awt.Color color)

Method Summary

java.awt.Color	**getColor**()
java.lang.Class	**getJavaClass**() Get the java Class meta-data for this type

java.lang.String	**getJavaType**() Get the Java type as a String
java.lang.String	**getJavaValue**() Get the Java representation of this value
java.lang.String	**getName**() Get the name of the argument

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

Constructor Detail

JGSLColor

```
public JGSLColor(java.lang.String name,
                 java.awt.Color color)
```

Method Detail

getColor

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

---

getJavaClass

```
public java.lang.Class getJavaClass()
```

Get the java Class meta-data for this type

Specified by:

getJavaClass in interface Type

Returns:

The Class mete-data for this type

---

getJavaType

```
public java.lang.String getJavaType()
```

Get the Java type as a String

Specified by:

getJavaType in interface Type

Returns:

a String containing the type

---

getJavaValue

```
public java.lang.String getJavaValue()
```

Get the Java representation of this value

Specified by:

getJavaValue in interface Value

Returns:

A String containing the Java representation of this value

---

getName

```
public java.lang.String getName()
```

Get the name of the argument

Specified by:

getName in interface Argument

Returns:

String containing the name

---

jgsl.model

Class JGSLDouble

```
java.lang.Object
  └ jgsl.model.JGSLDouble
```

All Implemented Interfaces:

Argument, Type, Value

```
public class JGSLDouble
```

extends java.lang.Object
implements Type, Value, Argument

Version:

$Id: JGSLDouble.java,v 1.2 2005/05/16 00:54:18 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

JGSLDouble(java.lang.String name, java.lang.Double value)
JGSLDouble(java.lang.String name, java.lang.String value)

Method Summary

java.lang.Class	**getJavaClass**() Get the java Class meta-data for this type

java.lang.String	**getJavaType**() Get the Java type as a String
java.lang.String	**getJavaValue**() Get the Java representation of this value
java.lang.String	**getName**() Get the name of the argument
java.lang.Double	**getValue**()

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

Constructor Detail

JGSLDouble

```
public JGSLDouble(java.lang.String name,
                  java.lang.Double value)
```

JGSLDouble

```
public JGSLDouble(java.lang.String name,
                  java.lang.String value)
```

Method Detail

getJavaClass

```
public java.lang.Class getJavaClass()
```

Get the java Class meta-data for this type

Specified by:
`getJavaClass` in interface `Type`

Returns:
The Class mete-data for this type

---

getJavaType

```
public java.lang.String getJavaType()
```

Get the Java type as a String

Specified by:
`getJavaType` in interface `Type`

Returns:
a String containing the type

---

getJavaValue

```
public java.lang.String getJavaValue()
```

Get the Java representation of this value

Specified by:
`getJavaValue` in interface `Value`

Returns:
A String containing the Java representation of this value

---

getName

```
public java.lang.String getName()
```

Get the name of the argument

Specified by:

`getName` in interface `Argument`

Returns:

String containing the name

---

getValue

```
public java.lang.Double getValue()
```

---

jgsl.io

Class JGSLFileFilter

```
java.lang.Object
  └javax.swing.filechooser.FileFilter
      └jgsl.io.JGSLFileFilter
```

```
public class JGSLFileFilter
```

extends javax.swing.filechooser.FileFilter

FileFilter for .jgsl files

Version:

$Id: JGSLFileFilter.java,v 1.5 2005/05/16 00:54:16 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`JGSLFileFilter()`

Method Summary

boolean	`accept(java.io.File f)`
java.lang.String	`getDescription()`

Methods inherited from class java.lang.Object

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

Constructor Detail

JGSLFileFilter

public **JGSLFileFilter**()

Method Detail

accept

public boolean **accept**(java.io.File f)

Specified by:
accept in class javax.swing.filechooser.FileFilter

---

getDescription

public java.lang.String **getDescription**()

Specified by:
getDescription in class javax.swing.filechooser.FileFilter

---

jgsl.model

Class JGSLInteger

```
java.lang.Object
  └ jgsl.model.JGSLInteger
```

All Implemented Interfaces:

Argument, Type, Value

```
public class JGSLInteger
```

extends java.lang.Object
implements Type, Value, Argument

Version:

$Id: JGSLInteger.java,v 1.2 2005/05/16 00:54:18 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`JGSLInteger(java.lang.String name, java.lang.Integer value)`
`JGSLInteger(java.lang.String name, java.lang.String value)`

Method Summary

`java.lang.Class`	`getJavaClass()` Get the java Class meta-data for this type

java.lang.String	**getJavaType**() Get the Java type as a String
java.lang.String	**getJavaValue**() Get the Java representation of this value
java.lang.String	**getName**() Get the name of the argument
java.lang.Integer	**getValue**()

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

Constructor Detail

JGSLInteger

```
public JGSLInteger(java.lang.String name,
                   java.lang.Integer value)
```

JGSLInteger

```
public JGSLInteger(java.lang.String name,
                   java.lang.String value)
```

Method Detail

getJavaClass

```
public java.lang.Class getJavaClass()
```

Get the java Class meta-data for this type

Specified by:
`getJavaClass` in interface `Type`

Returns:
The Class mete-data for this type

---

getJavaType

```
public java.lang.String getJavaType()
```

Get the Java type as a String

Specified by:
`getJavaType` in interface `Type`

Returns:
a String containing the type

---

getJavaValue

```
public java.lang.String getJavaValue()
```

Get the Java representation of this value

Specified by:
`getJavaValue` in interface `Value`

Returns:
A String containing the Java representation of this value

---

getName

```
public java.lang.String getName()
```

Get the name of the argument

Specified by:

`getName` in interface `Argument`

Returns:

String containing the name

---

getValue

```
public java.lang.Integer getValue()
```

---

jgsl.model

Class JGSLScript

```
java.lang.Object
  └ jgsl.model.JGSLScript
```

All Implemented Interfaces:

java.io.Serializable, ParseStatus, Script

```
public class JGSLScript
```

extends java.lang.Object
implements java.io.Serializable, Script, ParseStatus

A JGSLScript contains an ordered colllection of objects that implement the statement interface.

Version:

$Id: JGSLScript.java,v 1.7 2005/05/21 19:21:36 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

JGSLScript()

Method Summary

void	**add**(Statement s)

void	**addDocumentation**(java.lang.String d)
void	**addError**(ScriptError se) Add a ScriptError to the parse status
void	**addMessage**(ScriptMessage sm) Add a ScriptMessage to the parse status
void	**addWarning**(ScriptWarning sw) Add a ScriptWarning to the parse status
java.lang.String	**generateImplementation**() Generate the implementation class and return the name of the class
java.lang.String	**getClassFileName**()
java.lang.String	**getClassName**()
java.lang.String	**getDocumentation**() Returns the JGSL script documentation as specified in the DOC keyword by the script author.
int	**getErrorCount**()
java.lang.String	**getFullClassName**()
java.lang.String	**getJava**() Return the Java implementation of this script
java.lang.String	**getJavaForInit**()
int	**getMessageCount**()
java.lang.String	**getParseStatus**()

java.lang.String	**getScriptName**() Get the script name
int	**getWarningCount**()
boolean	**hasErrors**() Return the error state of the script
boolean	**hasMessages**() Return the message state of the script * @return true of the script contains messages or false otherwise
boolean	**hasWarnings**() Return the warning state of the script
void	**setScriptName**(java.lang.String scriptName) Set the scipt name
java.lang.String	**toString**()

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

Constructor Detail

JGSLScript

```
public JGSLScript()
```

Method Detail

add

```
public void add(Statement s)
```

---

addDocumentation

```
public void addDocumentation(java.lang.String d)
```

---

addError

```
public void addError(ScriptError se)
```

Add a ScriptError to the parse status

Specified by:
addError in interface ParseStatus

Parameters:
se -

---

addMessage

```
public void addMessage(ScriptMessage sm)
```

Add a ScriptMessage to the parse status

Specified by:
addMessage in interface ParseStatus

Parameters:
sm -

---

addWarning

```
public void addWarning(ScriptWarning sw)
```

Add a ScriptWarning to the parse status

Specified by:

addWarning in interface ParseStatus

Parameters:

sw -

---

generateImplementation

```
public java.lang.String generateImplementation()
```

Generate the implementation class and return the name of the class

Returns:

returns a String containing the full name of the implementation class

---

getClassFileName

```
public java.lang.String getClassFileName()
```

---

getClassName

```
public java.lang.String getClassName()
```

---

getDocumentation

```
public java.lang.String getDocumentation()
```

Returns the JGSL script documentation as specified in the DOC keyword by the script author.

Specified by:

getDocumentation in interface Script

Returns:

The script documentation

---

getErrorCount

```
public int getErrorCount()
```

---

getFullClassName

```
public java.lang.String getFullClassName()
```

---

getJava

```
public java.lang.String getJava()
```

Return the Java implementation of this script

Specified by:
`getJava` in interface `Script`

Returns:
the Java language implementation of this script

---

getJavaForInit

```
public java.lang.String getJavaForInit()
```

---

getMessageCount

```
public int getMessageCount()
```

---

getParseStatus

```
public java.lang.String getParseStatus()
```

---

getScriptName

```
public java.lang.String getScriptName()
```

Get the script name

Specified by:
`getScriptName` in interface `Script`

Returns:
String containing the script name

---

getWarningCount

```
public int getWarningCount()
```

---

hasErrors

```
public boolean hasErrors()
```

Return the error state of the script

Specified by:
`hasErrors` in interface `ParseStatus`

Returns:
true of the script contains errors or false otherwise

---

hasMessages

```
public boolean hasMessages()
```

Return the message state of the script * @return true of the script contains messages or false otherwise

Specified by:
`hasMessages` in interface `ParseStatus`

Returns:
true of the script contains messages or false otherwise

---

hasWarnings

```
public boolean hasWarnings()
```

Return the warning state of the script

Specified by:
`hasWarnings` in interface `ParseStatus`

Returns:
true of the script contains warnings or false otherwise

---

setScriptName

```
public void setScriptName(java.lang.String scriptName)
```

Set the scipt name

Specified by:
`setScriptName` in interface `Script`

Parameters:
`scriptName` - name of the script file

---

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

---

jgsl.model

Class JGSLString

```
java.lang.Object
  └ jgsl.model.JGSLString
```

All Implemented Interfaces:

Argument, Type, Value

```
public class JGSLString
```

extends java.lang.Object
implements Type, Value, Argument

Version:

$Id: JGSLString.java,v 1.2 2005/05/16 00:54:18 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

JGSLString(java.lang.String name, java.lang.String value)

Method Summary

java.lang.Class	**getJavaClass**() Get the java Class meta-data for this type
java.lang.String	**getJavaType**() Get the Java type as a String

java.lang.String	**getJavaValue**() Get the Java representation of this value
java.lang.String	**getName**() Get the name of the argument
java.lang.String	**getValue**()

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

Constructor Detail

JGSLString

```
public JGSLString(java.lang.String name,
                  java.lang.String value)
```

Method Detail

getJavaClass

```
public java.lang.Class getJavaClass()
```

Get the java Class meta-data for this type

Specified by:

getJavaClass in interface Type

Returns:

The Class mete-data for this type

---

getJavaType

```
public java.lang.String getJavaType()
```

Get the Java type as a String

Specified by:
`getJavaType` in interface `Type`

Returns:
a String containing the type

---

getJavaValue

```
public java.lang.String getJavaValue()
```

Get the Java representation of this value

Specified by:
`getJavaValue` in interface `Value`

Returns:
A String containing the Java representation of this value

---

getName

```
public java.lang.String getName()
```

Get the name of the argument

Specified by:
`getName` in interface `Argument`

Returns:
String containing the name

---

getValue

```
public java.lang.String getValue()
```

---

jgsl.view.swing

Class JGSLSwingFrame

```
java.lang.Object
  └ jgsl.view.swing.JGSLSwingFrame
```

All Implemented Interfaces:

java.awt.event.ActionListener, java.awt.event.ItemListener, java.util.EventListener, javax.swing.event.ListSelectionListener

---

```
public class JGSLSwingFrame
```

extends java.lang.Object
implements java.awt.event.ActionListener, java.awt.event.ItemListener, javax.swing.event.ListSelectionListener

The JGSLSwingFrame class is the main class for the interactive GUI.

Version:

$Id: JGSLSwingFrame.java,v 1.5 2005/05/21 01:42:11 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

JGSLSwingFrame(javax.swing.JFrame frame) Constructs a new frame that is initially invisible.

Method Summary

void	**actionPerformed**(java.awt.event.ActionEvent actionEvent)
javax.swing.JPanel	**getMainPanel**() Return reference to the main panel.
void	**itemStateChanged**(java.awt.event.ItemEvent itemEvent)
static void	**main**(java.lang.String[] args) Program entry point.

static void	**startJGSL**(java.lang.String[] args) Start the JGSL.
void	**valueChanged**(javax.swing.event.ListSelectionEvent listSelectionEvent)

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

Constructor Detail

JGSLSwingFrame

```
public JGSLSwingFrame(javax.swing.JFrame frame)
               throws java.awt.HeadlessException
```

Constructs a new frame that is initially invisible.

This constructor sets the component's locale property to the value returned by `JComponent.getDefaultLocale`.

Throws:

`java.awt.HeadlessException` - if GraphicsEnvironment.isHeadless() returns true.

See Also:

`GraphicsEnvironment.isHeadless()`, `Component.setSize(int, int)`, `Component.setVisible(boolean)`, `JComponent.getDefaultLocale()`

Method Detail

actionPerformed

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

Specified by:

`actionPerformed` in interface `java.awt.event.ActionListener`

---

getMainPanel

```
public javax.swing.JPanel getMainPanel()
```

Return reference to the main panel.

---

itemStateChanged

```
public void itemStateChanged(java.awt.event.ItemEvent itemEvent)
```

Specified by:
`itemStateChanged` in interface `java.awt.event.ItemListener`

---

main

```
public static void main(java.lang.String[] args)
```

Program entry point.

---

startJGSL

```
public static void startJGSL(java.lang.String[] args)
```

Start the JGSL. This method is needed to work around the strange startup requirements by the IntelliJ IDEA GUI builder.

---

valueChanged

```
public void valueChanged(javax.swing.event.ListSelectionEvent listSelectionEvent)
```

Specified by:
`valueChanged` in interface `javax.swing.event.ListSelectionListener`

---

jgsl.util

Class JGSLToImage

```
java.lang.Object
  └ jgsl.util.JGSLToImage
```

```
public class JGSLToImage
```

extends java.lang.Object

TODO - write java docs

Version:

$Id: JGSLToImage.java,v 1.1 2005/05/21 01:42:10 zenarchitect Exp $

Author:

$Author: zenarchitect $

Field Summary

static int	**BMP**
static int	**GIF**
static int	**JPEG**
static int	**PNG**

Constructor Summary

JGSLToImage(javax.swing.JFrame comp)

Method Summary	
void	**save**(java.lang.String fileName, int outType)
void	**save**(java.lang.String fileName, java.lang.String outType)

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

Field Detail

BMP

```
public static final int BMP
```

See Also:

Constant Field Values

GIF

```
public static final int GIF
```

See Also:

Constant Field Values

JPEG

```
public static final int JPEG
```

See Also:

Constant Field Values

PNG

```
public static final int PNG
```

See Also:

Constant Field Values

Constructor Detail

JGSLToImage

```
public JGSLToImage(javax.swing.JFrame comp)
```

Method Detail

save

```
public void save(java.lang.String fileName,
                 int outType)
          throws java.io.IOException
```

Throws:

`java.io.IOException`

---

save

```
public void save(java.lang.String fileName,
                 java.lang.String outType)
          throws java.io.IOException
```

Throws:

`java.io.IOException`

---

jgsl.view.swing

Class JGSLViewer

```
java.lang.Object
  └ jgsl.view.swing.JGSLViewer
```

```
public class JGSLViewer
```

extends java.lang.Object

The JGSLViewer class is executed as a main class by the JGSL at runtime. It takes the suppled class name and creates and instance using reflection. This class is an subclass of BaseFrame thus represents a compiled JGSL script. It assumed the the JVM and CLASSPATH are set properly by the SwingScriptViewer class.

Version:

$Id: JGSLViewer.java,v 1.5 2005/05/21 01:42:11 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`JGSLViewer()`

Method Summary

`static void`	`main(java.lang.String[] args)` Create the BaseFrame subclass and window closing actions then display the window to the user.

Methods inherited from class java.lang.Object

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

Constructor Detail

JGSLViewer

```
public JGSLViewer()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
                throws java.lang.Exception
```

Create the BaseFrame subclass and window closing actions then display the window to the user.

Parameters:

`args` - A single argument specifying the JGSL Java class name.

Throws:

`java.lang.Exception`

---

jgsl.model

Class Logical

```
java.lang.Object
  └ jgsl.model.Logical
```

All Implemented Interfaces:

java.io.Serializable, Statement

```
public class Logical
```

extends java.lang.Object
implements Statement, java.io.Serializable

A logical statement is one in which the product of a logical comparison resutls in TRUE or FALSE.

Version:

$Id: Logical.java,v 1.2 2005/05/16 00:54:18 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

Logical()

Method Summary

java.lang.String	**getJava**() This method returns the Java language equivalent of the JGSL statement.

java.lang.String	**getType**() Return the type of statement.
void	**setJGSL**(java.lang.String jgsl) Set the JGSL statement body

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

Constructor Detail

Logical

```
public Logical()
```

Method Detail

getJava

```
public java.lang.String getJava()
```

This method returns the Java language equivalent of the JGSL statement.

Specified by:
getJava in interface Statement

Returns:
Java language statement from the JGSL

---

getType

```
public java.lang.String getType()
```

Return the type of statement. The String form of the class name.

Specified by:
`getType` in interface `Statement`

---

setJGSL

```
public void setJGSL(java.lang.String jgsl)
```

Set the JGSL statement body

Specified by:
`setJGSL` in interface `Statement`

---

jgsl.io

Interface Message

All Known Implementing Classes:

ScriptError, ScriptMessage, ScriptWarning

```
public interface Message
```

Interface for message types

Version:

$Id: Message.java,v 1.2 2005/05/16 00:54:16 zenarchitect Exp $

Author:

ZenArchitect

Nested Class Summary

static class	**Message.MessageType** Message types

Method Summary

java.lang.String	**getDetailMessage**() Return a detailed message
java.lang.String	**getMessage**() Retuns the message
Message.MessageType	**getType**() Returns the type of message

Method Detail

getDetailMessage

`java.lang.String` **`getDetailMessage`**`()`

Return a detailed message

Returns:
A string containing a detailed message

---

getMessage

`java.lang.String` **`getMessage`**`()`

Retuns the message

Returns:
A string containing a simple message

---

getType

`Message.MessageType` **`getType`**`()`

Returns the type of message

Returns:
MessageType

---

jgsl.io

Enum Message.MessageType

```
java.lang.Object
  └ java.lang.Enum<Message.MessageType>
      └ jgsl.io.Message.MessageType
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<Message.MessageType>

Enclosing interface:

Message

```
public static enum Message.MessageType
```

extends java.lang.Enum<Message.MessageType>

Message types

Enum Constant Summary

ERROR
MESSAGE
WARNING

Method Summary

static Message.MessageType	**valueOf**(java.lang.String name) Returns the enum constant of this type with the specified name.

static Message.MessageType[]	**values**() Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum
compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

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

Enum Constant Detail

ERROR

public static final Message.MessageType **ERROR**

MESSAGE

public static final Message.MessageType **MESSAGE**

WARNING

public static final Message.MessageType **WARNING**

Method Detail

valueOf

public static Message.MessageType **valueOf**(java.lang.String name)

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

`name` - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

---

values

```
public static final Message.MessageType[] values()
```

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```
for(Message.MessageType c : Message.MessageType.values())
        System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they're declared

---

---

jgsl.parser

Class ParseException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └jgsl.parser.ParseException
```

All Implemented Interfaces:

java.io.Serializable

`public class` **`ParseException`**

extends java.lang.Exception

This exception is thrown when parse errors are encountered. You can explicitly create objects of this exception type by calling the method generateParseException in the generated parser.

You can modify this class to customize your error reporting mechanisms so long as you retain the public fields.

See Also:

Serialized Form

Field Summary

Token	**currentToken** This is the last token that has been consumed successfully.
int[][]	**expectedTokenSequences** Each entry in this array is an array of integers.

java.lang.String[]	**tokenImage** This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred.

Constructor Summary

ParseException() The following constructors are for use by you for whatever purpose you can think of.
ParseException(java.lang.String message)
ParseException(Token currentTokenVal, int[][] expectedTokenSequencesVal, java.lang.String[] tokenImageVal) This constructor is used by the method "generateParseException" in the generated parser.

Method Summary

java.lang.String	**getMessage**() This method has the standard behavior when this object has been created using the standard constructors.

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

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

Field Detail

currentToken

public Token **currentToken**

This is the last token that has been consumed successfully. If this object has been created due to a

parse error, the token followng this token will (therefore) be the first error token.

---

expectedTokenSequences

```
public int[][] expectedTokenSequences
```

Each entry in this array is an array of integers. Each array of integers represents a sequence of tokens (by their ordinal values) that is expected at this point of the parse.

---

tokenImage

```
public java.lang.String[] tokenImage
```

This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred. This array is defined in the generated ...Constants interface.

Constructor Detail

ParseException

```
public ParseException()
```

The following constructors are for use by you for whatever purpose you can think of. Constructing the exception in this manner makes the exception behave in the normal way - i.e., as documented in the class "Throwable". The fields "errorToken", "expectedTokenSequences", and "tokenImage" do not contain relevant information. The JavaCC generated code does not use these constructors.

ParseException

```
public ParseException(java.lang.String message)
```

ParseException

```
public ParseException(Token currentTokenVal,
                      int[][] expectedTokenSequencesVal,
                      java.lang.String[] tokenImageVal)
```

This constructor is used by the method "generateParseException" in the generated parser. Calling this constructor generates a new object of this type with the fields "currentToken", "expectedTokenSequences", and "tokenImage" set. The boolean flag "specialConstructor" is also set to true to indicate that this constructor was used to create this object. This constructor calls its super class with the empty string to force the "toString" method of parent class "Throwable" to print the error message in the form: ParseException:

Method Detail

getMessage

```
public java.lang.String getMessage()
```

This method has the standard behavior when this object has been created using the standard constructors. Otherwise, it uses "currentToken" and "expectedTokenSequences" to generate a parse error message and returns it. If this object has been created due to a parse error, and you do not catch it (it gets thrown from the parser), then this method is called during the printing of the final stack trace, and hence the correct error message gets displayed.

Overrides:

`getMessage` in class `java.lang.Throwable`

---

jgsl.io

Interface ParseStatus

All Known Implementing Classes:

JGSLScript

```
public interface ParseStatus
```

ParseStatus interface for collecting and reporting errors and error counts.

Version:

$Id: ParseStatus.java,v 1.2 2005/05/16 00:54:16 zenarchitect Exp $

Author:

zenarchitect

Method Summary

void	**addError**(ScriptError se) Add a ScriptError to the parse status
void	**addMessage**(ScriptMessage sm) Add a ScriptMessage to the parse status
void	**addWarning**(ScriptWarning sw) Add a ScriptWarning to the parse status
boolean	**hasErrors**() Return the error state of the script
boolean	**hasMessages**() Return the message state of the script
boolean	**hasWarnings**() Return the warning state of the script

Method Detail

addError

`void` **`addError`**`(`[`ScriptError`]` se)`

Add a ScriptError to the parse status

Parameters:

`se` -

---

addMessage

`void` **`addMessage`**`(`[`ScriptMessage`]` sm)`

Add a ScriptMessage to the parse status

Parameters:

`sm` -

---

addWarning

`void` **`addWarning`**`(`[`ScriptWarning`]` sw)`

Add a ScriptWarning to the parse status

Parameters:

`sw` -

---

hasErrors

`boolean` **`hasErrors`**`()`

Return the error state of the script

Returns:
true of the script contains errors or false otherwise

---

hasMessages

`boolean` **`hasMessages`**`()`

Return the message state of the script

Returns:
true of the script contains messages or false otherwise

---

hasWarnings

`boolean` **`hasWarnings`**`()`

Return the warning state of the script

Returns:
true of the script contains warnings or false otherwise

---

jgsl.model

Interface Script

All Known Implementing Classes:

JGSLScript

```
public interface Script
```

The script interface provide the set of operations for a script

Version:

$Id: Script.java,v 1.2 2005/05/16 00:54:19 zenarchitect Exp $

Author:

zenarchitect

Method Summary

java.lang.String	**getDocumentation**() Returns the JGSL script documentation as specified in the DOC keyword by the script author.
java.lang.String	**getJava**() Return the Java implementation of this script
java.lang.String	**getScriptName**() Get the script name
void	**setScriptName**(java.lang.String scriptName) Set the scipt name

Method Detail

getDocumentation

```
java.lang.String getDocumentation()
```

Returns the JGSL script documentation as specified in the DOC keyword by the script author.

Returns:
The script documentation

---

getJava

```
java.lang.String getJava()
```

Return the Java implementation of this script

Returns:
the Java language implementation of this script

---

getScriptName

```
java.lang.String getScriptName()
```

Get the script name

Returns:
String containing the script name

---

setScriptName

```
void setScriptName(java.lang.String scriptName)
```

Set the scipt name

Parameters:

`scriptName` - name of the script file

---

jgsl.controller.script

Class ScriptEngine

```
java.lang.Object
  └ jgsl.controller.script.ScriptEngine
```

```
public class ScriptEngine
```

extends java.lang.Object

The ScriptEngine class is the controller for the JGSL application. It contains the command line processor for the command console interface. The *Interactive methods are the controller interfaced for the Interactive GUI.

Version:
$Id: ScriptEngine.java,v 1.8 2005/05/21 01:42:06 zenarchitect Exp $

Author:
zenarchitect

Constructor Summary

ScriptEngine()

Method Summary

JGSLScript	**jarInteractive**(java.io.File scriptFileName, java.io.File jarFileName) The method will parse the script contained in scriptFileName and then create an executable JAR file with name of jarFileName containing the Java class for the JGSL script.
JGSLScript	**parseInteractive**(java.io.File fileName) Parse the script supplied in fileName and return the JGSLScript containing the JGSL object model for the script.
void	**processCommandLine**(java.lang.String[] args) Process the command line arguments and perform the requested actions.
JGSLScript	**viewInteractive**(java.io.File fileName, java.lang.String saveToFileName) This method will parse the script contained in fileName and then display the result in the JGSL viewer.

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

Constructor Detail

ScriptEngine

```
public ScriptEngine()
```

Method Detail

jarInteractive

```
public JGSLScript jarInteractive(java.io.File scriptFileName,
                                 java.io.File jarFileName)
                          throws ScriptEngineException,
                                 ScriptParserException
```

The method will parse the script contained in scriptFileName and then create an executable JAR file with name of jarFileName containing the Java class for the JGSL script.

Parameters:
`scriptFileName` - Name of JGSL script to create the JAR from.
`jarFileName` - Name of JAR file to generate
Returns:
JGSLScript object containing the script object model
Throws:
`ScriptParserException` - If a problem is encountered during parsing a ScriptParser exception will be thrown.
`ScriptEngineException` - If a problem occurs during the creation of the JAR file.
See Also:
`JGSLScript`

---

parseInteractive

```
public JGSLScript parseInteractive(java.io.File fileName)
                           throws ScriptParserException
```

Parse the script supplied in fileName and return the JGSLScript containing the JGSL object model for the script.

Parameters:
`fileName` - Name of the JGSL script file
Returns:
JGSLScript object containing the script object model
Throws:
`ScriptParserException` - If a problem is encountered during parsing a ScriptParser exception will be thrown.
See Also:
`JGSLScript`

---

processCommandLine

```
public void processCommandLine(java.lang.String[] args)
```

```
                    throws ScriptParserException,
                           ScriptEngineException
```

Process the command line arguments and perform the requested actions. The set of available options is listed below.

```
 usage: jgsl.JGSL
      -d,--doc jgsl script filt script doc file   Generate script documenation
      -e,--exec script file                       Execute the script file
      -h,--help                                   Print this message
      -j,--jar jgsl script file JAR file          Generate JAR file for script
      -l,--logLevel user log level                Set user logging level
                                                      to one of: LOG, DEBUG, ERROR,
WARNING
      -p,--parse script file                      Parse the script file and print
the results
      -s,--sysLogLevel system log level           Set system logging
                                                      level to one of: LOG, DEBUG,
ERROR, WARNIN
      -v,--view Type of viewer, supports: swing   Parse, execute and view the script
```

Parameters:

`args` - Array of String references to valid arguments

Throws:

`ScriptParserException` - If parsing is requested and problem is found this script this exception will be thrown.

`ScriptEngineException` - If execution or JAR is requested and a problem occurs this exception will be thrown.

---

viewInteractive

```
public JGSLScript viewInteractive(java.io.File fileName,
                                  java.lang.String saveToFileName)
                           throws ScriptParserException
```

This method will parse the script contained in fileName and then display the result in the JGSL viewer.

Parameters:

`fileName` - Name of the JGSL script file

Returns:

JGSLScript object containing the script object model

Throws:

`ScriptParserException` - If a problem is encountered during parsing a ScriptParser exception will be thrown.

See Also:

`JGSLScript`

---

jgsl.controller.script

Class ScriptEngineException

```
java.lang.Object
  └java.lang.Throwable
      └jgsl.controller.script.ScriptEngineException
```

All Implemented Interfaces:

java.io.Serializable

```
public class ScriptEngineException
```

extends java.lang.Throwable

ScriptEngineException is thrown by the ScriptEngine class to report exception conditions.

Version:

$Id: ScriptEngineException.java,v 1.3 2005/05/16 00:54:15 zenarchitect Exp $

Author:

zenarchitect

See Also:

Serialized Form

Constructor Summary

ScriptEngineException() Constructs a new throwable with null as its detail message.
ScriptEngineException(java.lang.String message) Constructs a new throwable with the specified detail message.
ScriptEngineException(java.lang.String message, java.lang.Throwable cause) Constructs a new throwable with the specified detail message and cause.

`ScriptEngineException(java.lang.Throwable cause)` Constructs a new throwable with the specified cause and a detail message of `(cause==null ? null : cause.toString())` (which typically contains the class and detail message of `cause`).

Method Summary

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, wait, wait, wait`

Constructor Detail

ScriptEngineException

`public` **`ScriptEngineException`**`()`

Constructs a new throwable with `null` as its detail message. The cause is not initialized, and may subsequently be initialized by a call to `Throwable.initCause(java.lang.Throwable)`.

The `Throwable.fillInStackTrace()` method is called to initialize the stack trace data in the newly created throwable.

ScriptEngineException

`public` **`ScriptEngineException`**`(java.lang.String message)`

Constructs a new throwable with the specified detail message. The cause is not initialized, and may subsequently be initialized by a call to `Throwable.initCause(java.lang.`

Throwable).

The Throwable.fillInStackTrace() method is called to initialize the stack trace data in the newly created throwable.

Parameters:

message - the detail message. The detail message is saved for later retrieval by the Throwable.getMessage() method.

ScriptEngineException

```
public ScriptEngineException(java.lang.String message,
                             java.lang.Throwable cause)
```

Constructs a new throwable with the specified detail message and cause.

Note that the detail message associated with cause is *not* automatically incorporated in this throwable's detail message.

The Throwable.fillInStackTrace() method is called to initialize the stack trace data in the newly created throwable.

Parameters:

message - the detail message (which is saved for later retrieval by the Throwable.getMessage() method).

cause - the cause (which is saved for later retrieval by the Throwable.getCause() method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.)

Since:

1.4

ScriptEngineException

```
public ScriptEngineException(java.lang.Throwable cause)
```

Constructs a new throwable with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and

detail message of `cause`). This constructor is useful for throwables that are little more than wrappers for other throwables (for example, `java.security.PrivilegedActionException`).

The `Throwable.fillInStackTrace()` method is called to initialize the stack trace data in the newly created throwable.

Parameters:

`cause` - the cause (which is saved for later retrieval by the `Throwable.getCause()` method). (A `null` value is permitted, and indicates that the cause is nonexistent or unknown.)

Since:

1.4

---

jgsl.controller.script

Class ScriptEngineTest

```
java.lang.Object
  └junit.framework.Assert
      └junit.framework.TestCase
          └jgsl.controller.script.ScriptEngineTest
```

All Implemented Interfaces:

junit.framework.Test

```
public class ScriptEngineTest
```

extends junit.framework.TestCase

ScriptParser JUnit tester.

Version:

$Id: ScriptEngineTest.java,v 1.5 2005/05/21 01:42:06 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`ScriptEngineTest(java.lang.String name)`

Method Summary

void	`setUp()`

static junit.framework.Test	**suite**()
void	**tearDown**()
void	**testGenDocs**()
void	**testGenJar**()
void	**testScriptEngine**()
void	**testScriptSaveToFile**()
void	**testScriptViewer**()

Methods inherited from class junit.framework.TestCase
countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class junit.framework.Assert
assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail

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

Constructor Detail

ScriptEngineTest

```
public ScriptEngineTest(java.lang.String name)
```

Method Detail

setUp

```
public void setUp()
           throws java.lang.Exception
```

Overrides:
`setUp` in class `junit.framework.TestCase`
Throws:
`java.lang.Exception`

---

suite

```
public static junit.framework.Test suite()
```

---

tearDown

```
public void tearDown()
              throws java.lang.Exception
```

Overrides:
`tearDown` in class `junit.framework.TestCase`
Throws:
`java.lang.Exception`

---

testGenDocs

```
public void testGenDocs()
                  throws java.lang.Exception,
```

ScriptEngineException

Throws:

java.lang.Exception
ScriptEngineException

---

testGenJar

```
public void testGenJar()
                throws java.lang.Exception,
                       ScriptEngineException
```

Throws:

java.lang.Exception
ScriptEngineException

---

testScriptEngine

```
public void testScriptEngine()
                    throws java.lang.Exception,
                           ScriptEngineException
```

Throws:

java.lang.Exception
ScriptEngineException

---

testScriptSaveToFile

```
public void testScriptSaveToFile()
                        throws java.lang.Exception,
                               ScriptEngineException
```

Throws:

java.lang.Exception

ScriptEngineException

---

testScriptViewer

```
public void testScriptViewer()
                    throws java.lang.Exception,
                           ScriptEngineException
```

Throws:

java.lang.Exception
ScriptEngineException

---

jgsl.io

Class ScriptError

```
java.lang.Object
  └ jgsl.io.ScriptError
```

All Implemented Interfaces:

Message

```
public class ScriptError
```

extends java.lang.Object
implements Message

A single script error with line and column info.

Version:

$Id: ScriptError.java,v 1.2 2005/05/16 00:54:16 zenarchitect Exp $

Author:

zenarchitect

Nested Class Summary

Nested classes/interfaces inherited from interface jgsl.io.Message
Message.MessageType

Constructor Summary

ScriptError(java.lang.String message)

ScriptError(java.lang.String message, int lineNumber, int colNumber)

Method Summary

java.lang.String	**getDetailMessage**() Return a detailed message
java.lang.String	**getMessage**() Retuns the message
Message.MessageType	**getType**() Returns the type of message

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

Constructor Detail

ScriptError

```
public ScriptError(java.lang.String message)
```

ScriptError

```
public ScriptError(java.lang.String message,
                   int lineNumber,
                   int colNumber)
```

Method Detail

getDetailMessage

```
public java.lang.String getDetailMessage()
```

Return a detailed message

Specified by:
`getDetailMessage` in interface `Message`

Returns:
A string containing a detailed message

---

getMessage

```
public java.lang.String getMessage()
```

Retuns the message

Specified by:
`getMessage` in interface `Message`

Returns:
A string containing a simple message

---

getType

```
public Message.MessageType getType()
```

Returns the type of message

Specified by:
`getType` in interface `Message`

Returns:
MessageType

---

jgsl.io

Class ScriptMessage

```
java.lang.Object
  └ jgsl.io.ScriptMessage
```

All Implemented Interfaces:

Message

```
public class ScriptMessage
```

extends java.lang.Object
implements Message

A general script message generated by the parser.

Version:

$Id: ScriptMessage.java,v 1.2 2005/05/16 00:54:16 zenarchitect Exp $

Author:

zenarchitect

Nested Class Summary

Nested classes/interfaces inherited from interface jgsl.io.Message
Message.MessageType

Constructor Summary

ScriptMessage(java.lang.String message)

Method Summary

java.lang.String	**getDetailMessage**() Return a detailed message
java.lang.String	**getMessage**() Retuns the message
Message.MessageType	**getType**() Returns the type of message

Methods inherited from class java.lang.Object

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

Constructor Detail

ScriptMessage

```
public ScriptMessage(java.lang.String message)
```

Method Detail

getDetailMessage

```
public java.lang.String getDetailMessage()
```

Return a detailed message

Specified by:
getDetailMessage in interface Message

Returns:
A string containing a detailed message

---

getMessage

```
public java.lang.String getMessage()
```

Retuns the message

Specified by:
`getMessage` in interface `Message`

Returns:
A string containing a simple message

---

getType

```
public Message.MessageType getType()
```

Returns the type of message

Specified by:
`getType` in interface `Message`

Returns:
MessageType

---

jgsl.io

Class ScriptParser

```
java.lang.Object
  └ jgsl.io.ScriptParser
```

```
public class ScriptParser
```

extends java.lang.Object

Parse the specified script file using the JGSL_Parser and report the status of the parse.

Version:

$Id: ScriptParser.java,v 1.6 2005/05/16 00:54:16 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

ScriptParser()

Method Summary

JGSLScript	**execScript**(java.io.File scriptFile)
java.lang.String	**parseScript**(java.io.File scriptFile)

Methods inherited from class java.lang.Object

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

Constructor Detail

ScriptParser

```
public ScriptParser()
```

Method Detail

execScript

```
public JGSLScript execScript(java.io.File scriptFile)
                      throws ScriptParserException
```

Throws:

ScriptParserException

---

parseScript

```
public java.lang.String parseScript(java.io.File scriptFile)
                             throws ScriptParserException
```

Throws:

ScriptParserException

---

jgsl.io

Class ScriptParserException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └jgsl.io.ScriptParserException
```

All Implemented Interfaces:

java.io.Serializable

```
public class ScriptParserException
```

extends java.lang.Exception

Report script parsing exceptions.

Version:

$Id: ScriptParserException.java,v 1.2 2005/05/16 00:54:16 zenarchitect Exp $

Author:

Joe Chavez

See Also:

Serialized Form

Constructor Summary

ScriptParserException(java.lang.String message)

Constructs a new exception with the specified detail message.

Method Summary

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, wait, wait, wait

Constructor Detail

ScriptParserException

```
public ScriptParserException(java.lang.String message)
```

Constructs a new exception with the specified detail message. The cause is not initialized, and may subsequently be initialized by a call to `Throwable.initCause(java.lang.Throwable)`.

Parameters:

`message` - the detail message. The detail message is saved for later retrieval by the `Throwable.getMessage()` method.

---

jgsl.io

Class ScriptParserTest

```
java.lang.Object
  └junit.framework.Assert
      └junit.framework.TestCase
          └jgsl.io.ScriptParserTest
```

All Implemented Interfaces:

junit.framework.Test

```
public class ScriptParserTest
```

extends junit.framework.TestCase

ScriptParser Tester.

Since:

`02/23/2005`

Version:

1.0

Author:

Constructor Summary

`ScriptParserTest(java.lang.String name)`

Method Summary

void	**setUp**()
static junit.framework.Test	**suite**()
void	**tearDown**()
void	**testParseScript**()

Methods inherited from class junit.framework.TestCase
countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class junit.framework.Assert
assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail

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

Constructor Detail

ScriptParserTest

```
public ScriptParserTest(java.lang.String name)
```

Method Detail

setUp

```
public void setUp()
           throws java.lang.Exception
```

Overrides:
`setUp` in class `junit.framework.TestCase`
Throws:
`java.lang.Exception`

---

suite

```
public static junit.framework.Test suite()
```

---

tearDown

```
public void tearDown()
              throws java.lang.Exception
```

Overrides:
`tearDown` in class `junit.framework.TestCase`
Throws:
`java.lang.Exception`

---

testParseScript

```
public void testParseScript()
                    throws java.lang.Exception
```

Throws:
`java.lang.Exception`

---

jgsl.io

Class ScriptParserUtil

```
java.lang.Object
  └ jgsl.io.ScriptParserUtil
```

```
public class ScriptParserUtil
```

extends java.lang.Object

Parse a Sting into and int

Version:

$Id: ScriptParserUtil.java,v 1.2 2005/05/16 00:54:17 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

`ScriptParserUtil()`

Method Summary

`static int`	`parseInt(java.lang.String val)`

Methods inherited from class java.lang.Object

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

Constructor Detail

ScriptParserUtil

```
public ScriptParserUtil()
```

Method Detail

parseInt

```
public static int parseInt(java.lang.String val)
                    throws ScriptParserException
```

Throws:

ScriptParserException

---

jgsl.view

Interface ScriptViewer

All Known Implementing Classes:

SwingScriptViewer

```
public interface ScriptViewer
```

Interface for JGSL script viewer windows.

Version:

$Id: ScriptViewer.java,v 1.3 2005/05/21 01:42:12 zenarchitect Exp $

Author:

zenarchitect

Method Summary

void	**renderScript**(java.lang.String fullClassName, java.lang.String saveToFileType) Render the script code in fullClassName to a GUI window.

Method Detail

renderScript

```
void renderScript(java.lang.String fullClassName,
                  java.lang.String saveToFileType)
```

Render the script code in fullClassName to a GUI window.

Parameters:

`fullClassName` -

---

jgsl.io

Class ScriptWarning

```
java.lang.Object
  └ jgsl.io.ScriptWarning
```

All Implemented Interfaces:

Message

```
public class ScriptWarning
```

extends java.lang.Object
implements Message

Record a script warning message.

Version:

$Id: ScriptWarning.java,v 1.2 2005/05/16 00:54:17 zenarchitect Exp $

Author:

zenarchitect

Nested Class Summary

Nested classes/interfaces inherited from interface jgsl.io.Message
Message.MessageType

Constructor Summary

ScriptWarning(java.lang.String message)

```
ScriptWarning(java.lang.String message, int lineNumber,
int colNumber)
```

Method Summary

java.lang.String	**getDetailMessage**() Return a detailed message
java.lang.String	**getMessage**() Retuns the message
Message.MessageType	**getType**() Returns the type of message

Methods inherited from class java.lang.Object

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

Constructor Detail

ScriptWarning

```
public ScriptWarning(java.lang.String message)
```

ScriptWarning

```
public ScriptWarning(java.lang.String message,
                     int lineNumber,
                     int colNumber)
```

Method Detail

getDetailMessage

```
public java.lang.String getDetailMessage()
```

Return a detailed message

Specified by:
`getDetailMessage` in interface `Message`

Returns:
A string containing a detailed message

---

getMessage

```
public java.lang.String getMessage()
```

Retuns the message

Specified by:
`getMessage` in interface `Message`

Returns:
A string containing a simple message

---

getType

```
public Message.MessageType getType()
```

Returns the type of message

Specified by:
`getType` in interface `Message`

Returns:
MessageType

---

jgsl.model

Interface Statement

All Known Implementing Classes:

AbstractStatement, Assignment, Command, Declaration, Documentation, Logical

---

```
public interface Statement
```

The Statement interface provides the set of operations common to all JGSL statements.

Version:

$Id: Statement.java,v 1.2 2005/05/16 00:54:19 zenarchitect Exp $

Author:

zenarchitect

Method Summary

java.lang.String	**getJava**() This method returns the Java language equivalent of the JGSL statement.
java.lang.String	**getType**() Return the type of statement.
void	**setJGSL**(java.lang.String jgsl) Set the JGSL statement body

Method Detail

getJava

```
java.lang.String getJava()
```

This method returns the Java language equivalent of the JGSL statement.

Returns:

Java language statement from the JGSL

---

getType

```
java.lang.String getType()
```

Return the type of statement. The String form of the class name.

---

setJGSL

```
void setJGSL(java.lang.String jgsl)
```

Set the JGSL statement body

---

jgsl.view.swing

Class SwingScriptViewer

```
java.lang.Object
  └ jgsl.view.swing.SwingScriptViewer
```

All Implemented Interfaces:

ScriptViewer

```
public class SwingScriptViewer
```

extends java.lang.Object
implements ScriptViewer

The SwingScriptViewer class creates an JVM that executes the JGSLViewer class with an argument of the compiled JGSL script Java class.

Version:

$Id: SwingScriptViewer.java,v 1.6 2005/05/21 01:42:11 zenarchitect Exp $

Author:

zenarchitect

Constructor Summary

SwingScriptViewer()

Method Summary

void	**renderScript**(java.lang.String fullClassName, java.lang.String saveToFileName) Reder the script by creating a Process object with the properly JGSL runtime class path.

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

Constructor Detail

SwingScriptViewer

```
public SwingScriptViewer()
```

Method Detail

renderScript

```
public void renderScript(java.lang.String fullClassName,
                         java.lang.String saveToFileName)
```

Reder the script by creating a Process object with the properly JGSL runtime class path. The runtime classpath includes the compile JGSL script in the form of a Java class. Also required on the classpath are the jgsl_rt.jar and log4j-1.2.9.jar files.

Specified by:

renderScript in interface ScriptViewer

Parameters:

fullClassName -

---

jgsl.parser

Class Token

```
java.lang.Object
  └ jgsl.parser.Token
```

```
public class Token
```

extends java.lang.Object

Describes the input token stream.

Field Summary	
int	**beginColumn** beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.
int	**beginLine** beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.
int	**endColumn** beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.
int	**endLine** beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.
java.lang.String	**image** The string image of the token.
int	**kind** An integer that describes the kind of this token.

Token	**next** A reference to the next regular (non-special) token from the input stream.
Token	**specialToken** This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (non-special) token.

Constructor Summary

Token()

Method Summary

static Token	**newToken**(int ofKind) Returns a new Token object, by default.
java.lang.String	**toString**() Returns the image.

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

Field Detail

beginColumn

public int **beginColumn**

beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

---

beginLine

public int **beginLine**

beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

---

endColumn

```
public int endColumn
```

beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

---

endLine

```
public int endLine
```

beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

---

image

```
public java.lang.String image
```

The string image of the token.

---

kind

```
public int kind
```

An integer that describes the kind of this token. This numbering system is determined by JavaCCParser, and a table of these numbers is stored in the file ...Constants.java.

---

next

`public Token next`

A reference to the next regular (non-special) token from the input stream. If this is the last token from the input stream, or if the token manager has not read tokens beyond this one, this field is set to null. This is true only if this token is also a regular token. Otherwise, see below for a description of the contents of this field.

---

specialToken

`public Token specialToken`

This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (non-special) token. If there are no such special tokens, this field is set to null. When there are more than one such special token, this field refers to the last of these special tokens, which in turn refers to the next previous special token through its specialToken field, and so on until the first special token (whose specialToken field is null). The next fields of special tokens refer to other special tokens that immediately follow it (without an intervening regular token). If there is no such token, this field is null.

Constructor Detail

Token

`public Token()`

Method Detail

newToken

`public static final Token newToken(int ofKind)`

Returns a new Token object, by default. However, if you want, you can create and return subclass objects based on the value of ofKind. Simply add the cases to the switch for all those special cases. For example, if you have a subclass of Token called IDToken that you want to create if ofKind is ID, simlpy add something like :

case MyParserConstants.ID : return new IDToken();

to the following switch statement. Then you can cast matchedToken variable to the appropriate type and use it in your lexical actions.

---

toString

```
public java.lang.String toString()
```

Returns the image.

Overrides:

`toString` in class `java.lang.Object`

---

jgsl.parser

Class TokenMgrError

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Error
          └jgsl.parser.TokenMgrError
```

All Implemented Interfaces:

java.io.Serializable

```
public class TokenMgrError
```

extends java.lang.Error

See Also:

Serialized Form

Constructor Summary

TokenMgrError()
TokenMgrError(boolean EOFSeen, int lexState, int errorLine, int errorColumn, java.lang.String errorAfter, char curChar, int reason)
TokenMgrError(java.lang.String message, int reason)

Method Summary

java.lang.String	**getMessage**() You can also modify the body of this method to customize your error messages.

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

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

Constructor Detail

TokenMgrError

```
public TokenMgrError()
```

TokenMgrError

```
public TokenMgrError(boolean EOFSeen,
                     int lexState,
                     int errorLine,
                     int errorColumn,
                     java.lang.String errorAfter,
                     char curChar,
                     int reason)
```

TokenMgrError

```
public TokenMgrError(java.lang.String message,
                     int reason)
```

Method Detail

getMessage

```
public java.lang.String getMessage()
```

You can also modify the body of this method to customize your error messages. For example, cases like LOOP_DETECTED and INVALID_LEXICAL_STATE are not of end-users concern, so you can return something like :

"Internal Error : Please file a bug report .... "

from this method for such cases in the release version of your parser.

Overrides:

`getMessage` in class `java.lang.Throwable`

---

jgsl.model

Interface Type

All Known Implementing Classes:

JGSLColor, JGSLDouble, JGSLInteger, JGSLString

```
public interface Type
```

Version:

$Id: Type.java,v 1.2 2005/05/16 00:54:19 zenarchitect Exp $

Author:

zenarchitect

Method Summary

java.lang.Class	**getJavaClass**() Get the java Class meta-data for this type
java.lang.String	**getJavaType**() Get the Java type as a String

Method Detail

getJavaClass

```
java.lang.Class getJavaClass()
```

Get the java Class meta-data for this type

Returns:

The Class mete-data for this type

---

getJavaType

`java.lang.String` **`getJavaType`**`()`

Get the Java type as a String

Returns:
a String containing the type

---

jgsl.model

Interface Value

All Known Implementing Classes:

JGSLColor, JGSLDouble, JGSLInteger, JGSLString

```
public interface Value
```

Version:

$Id: Value.java,v 1.2 2005/05/16 00:54:19 zenarchitect Exp $

Author:

zenarchitect

Method Summary

java.lang.String	**getJavaValue**() Get the Java representation of this value

Method Detail

getJavaValue

```
java.lang.String getJavaValue()
```

Get the Java representation of this value

Returns:

A String containing the Java representation of this value
