

API User's Guide

Java API

Java Application Programming Interface

User's Guide

How the API Is Organized

There are three levels to the API:

- All Packages
- All Classes (within a package)
- This Class (selected class).

Level 1 - All Packages

This level of the API provides links to the packages in the 1.0 release.

Level 2 - This Package

This level provides links to the classes and interfaces in a given package. There are three categories in the listing:

- Interfaces
- Classes
- Exceptions

Level 3 - This Class/Interface

This level begins with an index, followed by the detailed API. There are three categories at the class level.

- Variables
- Constructors
- Methods

A category is omitted when a class has no applicable entries.

Within these categories there is additional color coding as follows:

- Instance Variables
- Static Variables
- Constructors
- Instance Methods
- Static Methods

How to Locate Items

- To Browse A Package
Select a package from the list of All Packages. This list is the home page for the the Java API.
- To Locate a Class
Use the searchable index tool.
Or, select its package.
Select the class from the alphabetical index.
- To Browse a Class
Use the Next/Previous anchors to browse alphabetically.
Or, traverse the links within the class.
- To Locate a Method
Use the searchable index tool.
Or, scroll through the alphabetical class index to locate a method.

A Closer Look at the Class-Level API

Take a look at class String in the package java.lang. The navigational anchors are at the top. This is followed by the fully qualified class name and a representation of its position in the class hierarchy.

The next entries are links to the superclass and the interfaces, if any. This is followed by a description of the class, taken from the class comment. Notice how the programmer has embedded some code samples using html tags.

The author also chose to include a See Also entry to another class. Following the class-level entries for See Also, Version, and Author, the index begins.

The Index

Each class/interface begins with an index of its variables, constructors and methods, sorted alphabetically. The entry consists of the declaration and short description. The description is the first sentence of the doc comment for that item. The index entries are linked to their corresponding entries in the application programming interface which immediately follows.

The Detailed API

The index is followed by the complete API for each entry. Within the three categories: Variables, Constructors, and Methods, the entries are presented in the order they appear in the source. This is done to preserve the logical groupings established by the programmer.

Where Are All the Links in the API?

- At the top of each class/interface there are navigational anchors to the other levels and to Previous and Next (class or interface).
- There are links in the class type of every method and variable definition.
- At the top of each class/interface there is a drawing of the tree structure down to the current class/interface, in which each superclass is a link.
- Every method contains a list of exceptions that it may throw. These are linked to the appropriate class.
- The superclass and interface references at the beginning of the class are links.
- Every See Also is a link.
- When a method overrides a method in the superclass, the API has the entry "Overrides: foo in class bar." Both foo (the method name) and bar (the class name) are links.

Packages

API User's Guide

Package Index

Applet API Packages

[java.applet](#)
[java.awt](#)
[java.awt.image](#)
[java.awt.peer](#)
[java.io](#)
[java.lang](#)
[java.net](#)
[java.util](#)

Other Packages

[sun.tools.debug](#)



This documentation was ported to *MS Window's Help* by Bill Bercik.
Bill may be reached at: bill@dippybird.com

Stop by his web site and get the latest update to the *Information Portafilter* for Java at:
<http://www.dippybird.com/java.html>



Package java.applet

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.applet

Interface Index

[AppletContext](#)

[AppletStub](#)

[AudioClip](#)

Class Index

[Applet](#)

Package java.awt

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.awt

Interface Index

[LayoutManager](#)

[MenuContainer](#)

Class Index

[BorderLayout](#)

[Button](#)

[Canvas](#)

[CardLayout](#)

[Checkbox](#)

[CheckboxGroup](#)

[CheckboxMenuItem](#)

[Choice](#)

[Color](#)

[Component](#)

[Container](#)

[Dialog](#)

[Dimension](#)

[Event](#)

[FileDialog](#)

[FlowLayout](#)

[Font](#)

[FontMetrics](#)

[Frame](#)

[Graphics](#)

[GridBagConstraints](#)

[GridBagLayout](#)

[GridLayout](#)

[Image](#)

[Insets](#)

[Label](#)

[List](#)

[MediaTracker](#)

[Menu](#)

[MenuBar](#)

[MenuComponent](#)

[MenuItem](#)

[Panel](#)

[Point](#)

[Polygon](#)

Rectangle
Scrollbar
TextArea
TextComponent
TextField
Toolkit
Window

Exception Index

AWTException

Error Index

AWTError

Package java.awt.image

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.awt.image

Interface Index

[ImageConsumer](#)

[ImageObserver](#)

[ImageProducer](#)

Class Index

[ColorModel](#)

[CropImageFilter](#)

[DirectColorModel](#)

[FilteredImageSource](#)

[ImageFilter](#)

[IndexColorModel](#)

[MemoryImageSource](#)

[PixelGrabber](#)

[RGBImageFilter](#)

Package java.awt.peer

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.awt.peer

Interface Index

[ButtonPeer](#)
[CanvasPeer](#)
[CheckboxMenuItemPeer](#)
[CheckboxPeer](#)
[ChoicePeer](#)
[ComponentPeer](#)
[ContainerPeer](#)
[DialogPeer](#)
[FileDialogPeer](#)
[FramePeer](#)
[LabelPeer](#)
[ListPeer](#)
[MenuBarPeer](#)
[MenuComponentPeer](#)
[MenuItemPeer](#)
[MenuPeer](#)
[PanelPeer](#)
[ScrollbarPeer](#)
[TextAreaPeer](#)
[TextComponentPeer](#)
[TextFieldPeer](#)
[WindowPeer](#)

Package java.io

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.io

Interface Index

[DataInput](#)
[DataOutput](#)
[FilenameFilter](#)

Class Index

[BufferedInputStream](#)
[BufferedOutputStream](#)
[ByteArrayInputStream](#)
[ByteArrayOutputStream](#)
[DataInputStream](#)
[DataOutputStream](#)
[File](#)
[FileDescriptor](#)
[FileInputStream](#)
[FileOutputStream](#)
[FilterInputStream](#)
[FilterOutputStream](#)
[InputStream](#)
[LineNumberInputStream](#)
[OutputStream](#)
[PipedInputStream](#)
[PipedOutputStream](#)
[PrintStream](#)
[PushbackInputStream](#)
[RandomAccessFile](#)
[SequenceInputStream](#)
[StreamTokenizer](#)
[StringBufferInputStream](#)

Exception Index

[EOFException](#)
[FileNotFoundException](#)
[IOException](#)
[InterruptedIOException](#)
[UTFDataFormatException](#)

Package java.lang

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.lang

Interface Index

[Cloneable](#)

[Runnable](#)

Class Index

[Boolean](#)

[Character](#)

[Class](#)

[ClassLoader](#)

[Compiler](#)

[Double](#)

[Float](#)

[Integer](#)

[Long](#)

[Math](#)

[Number](#)

[Object](#)

[Process](#)

[Runtime](#)

[SecurityManager](#)

[String](#)

[StringBuffer](#)

[System](#)

[Thread](#)

[ThreadGroup](#)

[Throwable](#)

Exception Index

[ArithmeticException](#)

[ArrayIndexOutOfBoundsException](#)

[ArrayStoreException](#)

[ClassCastException](#)

[ClassNotFoundException](#)

[CloneNotSupportedException](#)

[Exception](#)

[IllegalAccessException](#)

[IllegalArgumentException](#)

IllegalMonitorStateException
IllegalThreadStateException
IndexOutOfBoundsException
InstantiationException
InterruptedException
NegativeArraySizeException
NoSuchMethodException
NullPointerException
NumberFormatException
RuntimeException
SecurityException
StringIndexOutOfBoundsException

Error Index

AbstractMethodError
ClassCircularityError
ClassFormatError
Error
IllegalAccessError
IncompatibleClassChangeError
InstantiationException
InternalError
LinkageError
NoClassDefFoundError
NoSuchFieldError
NoSuchMethodError
OutOfMemoryError
StackOverflowError
ThreadDeath
UnknownError
UnsatisfiedLinkError
VerifyError
VirtualMachineError

Package java.net

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.net

Interface Index

[ContentHandlerFactory](#)
[SocketImplFactory](#)
[URLStreamHandlerFactory](#)

Interface Index

[ContentHandler](#)
[DatagramPacket](#)
[DatagramSocket](#)
[InetAddress](#)
[ServerSocket](#)
[Socket](#)
[SocketImpl](#)
[URL](#)
[URLConnection](#)
[URLEncoder](#)
[URLStreamHandler](#)

Interface Index

[MalformedURLException](#)
[ProtocolException](#)
[SocketException](#)
[UnknownHostException](#)
[UnknownServiceException](#)

Package java.util

[All Packages](#) [Class Hierarchy](#) [Index](#)

package java.util

Interface Index

[Enumeration](#)

[Observer](#)

Interface Index

[BitSet](#)

[Date](#)

[Dictionary](#)

[Hashtable](#)

[Observable](#)

[Properties](#)

[Random](#)

[Stack](#)

[StringTokenizer](#)

[Vector](#)

Interface Index

[EmptyStackException](#)

[NoSuchElementException](#)

Package sun.tools.debug

[All Packages](#) [Class Hierarchy](#) [Index](#)

package sun.tools.debug

Interface Index

[DebuggerCallback](#)

Interface Index

[RemoteArray](#)
[RemoteBoolean](#)
[RemoteByte](#)
[RemoteChar](#)
[RemoteClass](#)
[RemoteDebugger](#)
[RemoteDouble](#)
[RemoteField](#)
[RemoteFloat](#)
[RemoteInt](#)
[RemoteLong](#)
[RemoteObject](#)
[RemoteShort](#)
[RemoteStackFrame](#)
[RemoteStackVariable](#)
[RemoteString](#)
[RemoteThread](#)
[RemoteThreadGroup](#)
[RemoteValue](#)
[StackFrame](#)

Index of all Fields and Methods

[All Packages](#) [Class Hierarchy](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

Index of all Fields and Methods

A

ABORT. Static variable in interface java.awt.image.[ImageObserver](#)

An image which was being tracked asynchronously was aborted before production was complete.

ABORTED. Static variable in class java.awt.[MediaTracker](#)

Flag indicating the download of some media was aborted.

abs(double). Static method in class java.lang.[Math](#)

Returns the absolute double value of a.

abs(float). Static method in class java.lang.[Math](#)

Returns the absolute float value of a.

abs(int). Static method in class java.lang.[Math](#)

Returns the absolute integer value of a.

abs(long). Static method in class java.lang.[Math](#)

Returns the absolute long value of a.

AbstractMethodError([String](#)). Constructor for class java.lang.[AbstractMethodError](#)

Constructs an [AbstractMethodError](#) with no detail message.

AbstractMethodError([String](#)). Constructor for class java.lang.[AbstractMethodError](#)

Constructs an [AbstractMethodError](#) with the specified detail message.

accept([String](#)). Method in class java.net.[ServerSocket](#)

Accepts a connection.

accept([File](#), [String](#)). Method in interface java.io.[FilenameFilter](#)

Determines whether a name should be included in a file list.

accept([SocketImpl](#)). Method in class java.net.[SocketImpl](#)

Accepts a connection.

acos(double). Static method in class java.lang.[Math](#)

Returns the arc cosine of a, in the range of 0.0 through Pi.

action([Event](#), [Object](#)). Method in class java.awt.[Component](#)

Called if an action occurs in the Component.

ACTION_EVENT. Static variable in class java.awt.[Event](#)

An action event.

activeCount([Thread](#)). Static method in class java.lang.[Thread](#)

Returns the current number of active Threads in this Thread group.

activeCount([ThreadGroup](#)). Method in class java.lang.[ThreadGroup](#)

Returns an estimate of the number of active Threads in the Thread group.

activeGroupCount([ThreadGroup](#)). Method in class java.lang.[ThreadGroup](#)

Returns an estimate of the number of active groups in the Thread group.

add([Component](#)). Method in class java.awt.[Container](#)

Adds the specified component to this container.

add([Component](#), int). Method in class java.awt.[Container](#)

Adds the specified component to this container at the given position.

add(int, int). Method in class java.awt.[Rectangle](#)

Adds a point to a rectangle.

add([Menu](#)). Method in class java.awt.[MenuBar](#)

Adds the specified menu to the menu bar.
add(MenuItem). Method in class java.awt.Menu
Adds the specified item to this menu.
add(Point). Method in class java.awt.Rectangle
Adds a point to a rectangle.
add(Rectangle). Method in class java.awt.Rectangle
Adds a rectangle to a rectangle.
add(String). Method in class java.awt.Menu
Adds an item with the specified label to this menu.
add(String, Component). Method in class java.awt.Container
Adds the specified component to this container.
addConsumer(ImageConsumer). Method in class java.awt.image.FilteredImageSource
Adds an ImageConsumer to the list of consumers interested in data for this image.
addConsumer(ImageConsumer). Method in interface java.awt.image.ImageProducer
This method is used to register an ImageConsumer with the ImageProducer for access to the image data during a later reconstruction of the Image.
addConsumer(ImageConsumer). Method in class java.awt.image.MemoryImageSource
Adds an ImageConsumer to the list of consumers interested in data for this image.
addElement(Object). Method in class java.util.Vector
Adds the specified object as the last element of the vector.
addHelpMenu(Menu). Method in interface java.awt.peer.MenuBarPeer
addImage(Image, int). Method in class java.awt.MediaTracker
Adds an image to the list of images being tracked.
addImage(Image, int, int, int). Method in class java.awt.MediaTracker
Adds a scaled image to the list of images being tracked.
addItem(MenuItem). Method in interface java.awt.peer.MenuPeer
addItem(String). Method in class java.awt.Choice
Adds an item to this Choice.
addItem(String). Method in class java.awt.List
Adds the specified item to the end of scrolling list.
addItem(String, int). Method in interface java.awt.peer.ChoicePeer
addItem(String, int). Method in class java.awt.List
Adds the specified item to the end of scrolling list.
addItem(String, int). Method in interface java.awt.peer.ListPeer
addLayoutComponent(String, Component). Method in class java.awt.BorderLayout
Adds the specified named component to the layout.
addLayoutComponent(String, Component). Method in class java.awt.CardLayout
Adds the specified component with the specified name to the layout.
addLayoutComponent(String, Component). Method in class java.awt.FlowLayout
Adds the specified component to the layout.
addLayoutComponent(String, Component). Method in class java.awt.GridBagLayout
Adds the specified component with the specified name to the layout.
addLayoutComponent(String, Component). Method in class java.awt.GridLayout
Adds the specified component with the specified name to the layout.
addLayoutComponent(String, Component). Method in interface java.awt.LayoutManager
Adds the specified component with the specified name to the layout.
addMenu(Menu). Method in interface java.awt.peer.MenuBarPeer
addNotify(). Method in class java.awt.Button
Creates the peer of the button.
addNotify(). Method in class java.awt.Canvas
Creates the peer of the canvas.
addNotify(). Method in class java.awt.Checkbox
Creates the peer of the Checkbox.
addNotify(). Method in class java.awt.CheckboxMenuItem
Creates the peer of the checkbox item.
addNotify(). Method in class java.awt.Choice

Creates the Choice's peer.

addNotify(). Method in class java.awt.Component
Notifies the Component to create a peer.

addNotify(). Method in class java.awt.Container
Notifies the container to create a peer.

addNotify(). Method in class java.awt.Dialog
Creates the frame's peer.

addNotify(). Method in class java.awt.FileDialog
Creates the frame's peer.

addNotify(). Method in class java.awt.Frame
Creates the Frame's peer.

addNotify(). Method in class java.awt.Label
Creates the peer for this label.

addNotify(). Method in class java.awt.List
Creates the peer for the list.

addNotify(). Method in class java.awt.Menu
Creates the menu's peer.

addNotify(). Method in class java.awt.MenuBar
Creates the menu bar's peer.

addNotify(). Method in class java.awt.MenuItem
Creates the menu item's peer.

addNotify(). Method in class java.awt.Panel
Creates the Panel's peer.

addNotify(). Method in class java.awt.Scrollbar
Creates the Scrollbar's peer.

addNotify(). Method in class java.awt.TextArea
Creates the TextArea's peer.

addNotify(). Method in class java.awt.TextField
Creates the TextField's peer.

addNotify(). Method in class java.awt.Window
Creates the Window's peer.

addObserver(Observer). Method in class java.util.Observable
Adds an observer to the observer list.

addPoint(int, int). Method in class java.awt.Polygon
Appends a point to a polygon.

address. Variable in class java.net.SocketImpl
The internet address where the socket will make a connection.

addSeparator(). Method in class java.awt.Menu
Adds a separator line, or a hyphen, to the menu at the current position.

addSeparator(). Method in interface java.awt.peer.MenuPeer

AdjustForGravity(GridBagConstraints, Rectangle). Method in class java.awt.GridBagLayout

after(Date). Method in class java.util.Date
Checks whether this date comes after the specified date.

ALLBITS. Static variable in interface java.awt.image.ImageObserver
A static image which was previously drawn is now complete and can be drawn again in its final form.

allowsMultipleSelections(). Method in class java.awt.List
Returns true if this list allows multiple selections.

allowUserInteraction. Variable in class java.net.URLConnection

ALT_MASK. Static variable in class java.awt.Event
The alt modifier constant.

anchor. Variable in class java.awt.GridBagConstraints

and(BitSet). Method in class java.util.BitSet
Logically ANDs this bit set with the specified set of bits.

append(boolean). Method in class java.lang.StringBuffer
Appends a boolean to the end of this buffer.

append(char). Method in class java.lang.StringBuffer

Appends a character to the end of this buffer.

append(char[]). Method in class java.lang.StringBuffer

Appends an array of characters to the end of this buffer.

append(char[], int, int). Method in class java.lang.StringBuffer

Appends a part of an array of characters to the end of this buffer.

append(double). Method in class java.lang.StringBuffer

Appends a double to the end of this buffer.

append(float). Method in class java.lang.StringBuffer

Appends a float to the end of this buffer.

append(int). Method in class java.lang.StringBuffer

Appends an integer to the end of this buffer.

append(long). Method in class java.lang.StringBuffer

Appends a long to the end of this buffer.

append(Object). Method in class java.lang.StringBuffer

Appends an object to the end of this buffer.

append(String). Method in class java.lang.StringBuffer

Appends a String to the end of this buffer.

appendText(String). Method in class java.awt.TextArea

Appends the given text to the end.

Applet(). Constructor for class java.applet.Applet

appletResize(int, int). Method in interface java.applet.AppletStub

Called when the applet wants to be resized.

arg. Variable in class java.awt.Event

An arbitrary argument.

ArithmeticException(). Constructor for class java.lang.ArithmeticException

Constructs an ArithmeticException with no detail message.

ArithmeticException(String). Constructor for class java.lang.ArithmeticException

Constructs an ArithmeticException with the specified detail message.

ArrangeGrid(Container). Method in class java.awt.GridBagLayout

arraycopy(Object, int, Object, int, int). Static method in class java.lang.System

Copies an array from the source array, beginning at the specified position, to the specified position of the destination array.

ArrayIndexOutOfBoundsException(). Constructor for class java.lang.

ArrayIndexOutOfBoundsException

Constructs an ArrayIndexOutOfBoundsException with no detail message.

ArrayIndexOutOfBoundsException(int). Constructor for class java.lang.

ArrayIndexOutOfBoundsException

Constructs a new ArrayIndexOutOfBoundsException class initialized to the specific index.

ArrayIndexOutOfBoundsException(String). Constructor for class java.lang.

ArrayIndexOutOfBoundsException

Constructs an ArrayIndexOutOfBoundsException class with the specified detail message.

ArrayStoreException(). Constructor for class java.lang.ArrayStoreException

Constructs a ArrayStoreException with no detail message.

ArrayStoreException(String). Constructor for class java.lang.ArrayStoreException

Constructs a ArrayStoreException with the specified detail message.

arrayTypeName(int). Method in class sun.tools.debug.RemoteArray

Return the element type as a string.

asin(double). Static method in class java.lang.Math

Returns the arc sine of a, in the range of -Pi/2 through Pi/2.

atan(double). Static method in class java.lang.Math

Returns the arc tangent of a, in the range of -Pi/2 through Pi/2.

atan2(double, double). Static method in class java.lang.Math

Converts rectangular coordinates (a, b) to polar (r, theta).

available(). Method in class java.io.BufferedInputStream

Returns the number of bytes that can be read without blocking.

available(). Method in class java.io.ByteArrayInputStream

Returns the number of available bytes in the buffer.
available(). Method in class java.io.FileInputStream
Returns the number of bytes that can be read without blocking.
available(). Method in class java.io.FilterInputStream
Returns the number of bytes that can be read without blocking.
available(). Method in class java.io.InputStream
Returns the number of bytes that can be read without blocking.
available(). Method in class java.io.LineNumberInputStream
Returns the number of bytes that can be read without blocking.
available(). Method in class java.io.PushbackInputStream
Returns the number of bytes that can be read.
available(). Method in class java.net.SocketImpl
Returns the number of bytes that can be read without blocking.
available(). Method in class java.io.StringBufferInputStream
Returns the number of available bytes in the buffer.
AWTError(String). Constructor for class java.awt.AWTError
AWTException(String). Constructor for class java.awt.AWTException
Constructs an AWTException with the specified detail message.

B

before(Date). Method in class java.util.Date
Checks whether this date comes before the specified date.
bind(InetAddress, int). Method in class java.net.SocketImpl
Binds the socket to the specified port on the specified host.
BitSet(). Constructor for class java.util.BitSet
Creates an empty set.
BitSet(int). Constructor for class java.util.BitSet
Creates an empty set with the specified size.
black. Static variable in class java.awt.Color
The color black.
blue. Static variable in class java.awt.Color
The color blue.
BOLD. Static variable in class java.awt.Font
The bold style constant.
Boolean(boolean). Constructor for class java.lang.Boolean
Constructs a Boolean object initialized to the specified boolean value.
Boolean(String). Constructor for class java.lang.Boolean
Constructs a Boolean object initialized to the value specified by the String parameter.
booleanValue(). Method in class java.lang.Boolean
Returns the value of this Boolean object as a boolean.
BorderLayout(). Constructor for class java.awt.BorderLayout
Constructs a new BorderLayout.
BorderLayout(int, int). Constructor for class java.awt.BorderLayout
Constructs a BorderLayout with the specified gaps.
BOTH. Static variable in class java.awt.GridBagConstraints
bottom. Variable in class java.awt.Insets
The inset from the bottom.
bounds(). Method in class java.awt.Component
Returns the current bounds of this component.
breakpointEvent(RemoteThread). Method in interface sun.tools.debug.DebuggerCallback
A breakpoint has been hit in the specified thread.
brighter(). Method in class java.awt.Color

Returns a brighter version of this color.

buf. Variable in class java.io.BufferedInputStream
The buffer where data is stored.

buf. Variable in class java.io.BufferedOutputStream
The buffer where data is stored.

buf. Variable in class java.io.ByteArrayInputStream
The buffer where data is stored.

buf. Variable in class java.io.ByteArrayOutputStream
The buffer where data is stored.

buffer. Variable in class java.io.StringBufferInputStream
The buffer where data is stored.

BufferedInputStream(InputStream). Constructor for class java.io. BufferedInputStream
Creates a new buffered stream with a default buffer size.

BufferedInputStream(InputStream, int). Constructor for class java.io. BufferedInputStream
Creates a new buffered stream with the specified buffer size.

BufferedOutputStream(OutputStream). Constructor for class java.io. BufferedOutputStream
Creates a new buffered stream with a default buffer size.

BufferedOutputStream(OutputStream, int). Constructor for class java.io. BufferedOutputStream
Creates a new buffered stream with the specified buffer size.

Button(()). Constructor for class java.awt.Button
Constructs a Button with no label.

Button(String). Constructor for class java.awt.Button
Constructs a Button with a string label.

ByteArrayInputStream(byte[]). Constructor for class java.io. ByteArrayInputStream
Creates an ByteArrayInputStream from the specified array of bytes.

ByteArrayInputStream(byte[], int, int). Constructor for class java.io. ByteArrayInputStream
Creates an ByteArrayInputStream from the specified array of bytes.

ByteArrayOutputStream(()). Constructor for class java.io.ByteArrayOutputStream
Creates a new ByteArrayOutputStream.

ByteArrayOutputStream(int). Constructor for class java.io.ByteArrayOutputStream
Creates a new ByteArrayOutputStream with the specified initial size.

bytesTransferred. Variable in class java.io.InterruptedIOException
Reports how many bytes had been transferred as part of the IO operation before it was interrupted.

bytesWidth(byte[], int, int). Method in class java.awt.FontMetrics
Returns the width of the specified array of bytes in this Font.

C

canFilterIndexColorModel. Variable in class java.awt.image.RGBImageFilter
This boolean indicates whether or not it is acceptable to apply the color filtering of the filterRGB method to the color table entries of an IndexColorModel object in lieu of pixel by pixel filtering.

canRead(()). Method in class java.io.File
Returns a boolean indicating whether or not a readable file exists.

Canvas(()). Constructor for class java.awt.Canvas

canWrite(()). Method in class java.io.File
Returns a boolean indicating whether or not a writable file exists.

capacity(()). Method in class java.lang.StringBuffer
Returns the current capacity of the String buffer.

capacity(()). Method in class java.util.Vector
Returns the current capacity of the vector.

capacityIncrement. Variable in class java.util.Vector
The size of the increment.

CardLayout(()). Constructor for class java.awt.CardLayout

Creates a new card layout.

CardLayout(int, int). Constructor for class java.awt.CardLayout

Creates a card layout with the specified gaps.

catchExceptions(). Method in class sun.tools.debug.RemoteClass

Enter the debugger when an instance of this class is thrown.

ceil(double). Static method in class java.lang.Math

Returns the "ceiling" or smallest whole number greater than or equal to a.

CENTER. Static variable in class java.awt.FlowLayout

The right alignment variable.

CENTER. Static variable in class java.awt.GridBagConstraints

CENTER. Static variable in class java.awt.Label

The center alignment.

Character(char). Constructor for class java.lang.Character

Constructs a Character object with the specified value.

charAt(int). Method in class java.lang.String

Returns the character at the specified index.

charAt(int). Method in class java.lang.StringBuffer

Returns the character at the specified index.

charsWidth(char[], int, int). Method in class java.awt.FontMetrics

Returns the width of the specified character array in this Font.

charValue(). Method in class java.lang.Character

Returns the value of this Character object.

charWidth(char). Method in class java.awt.FontMetrics

Returns the width of the specified character in this Font.

charWidth(int). Method in class java.awt.FontMetrics

Returns the width of the specified character in this Font.

checkAccept(String, int). Method in class java.lang.SecurityManager

Checks to see if a socket connection to the specified port on the specified host has been accepted.

checkAccess(). Method in class java.lang.Thread

Checks whether the current Thread is allowed to modify this Thread.

checkAccess(). Method in class java.lang.ThreadGroup

Checks to see if the current Thread is allowed to modify this group.

checkAccess(Thread). Method in class java.lang.SecurityManager

Checks to see if the specified Thread is allowed to modify the Thread group.

checkAccess(ThreadGroup). Method in class java.lang.SecurityManager

Checks to see if the specified Thread group is allowed to modify this group.

checkAll(). Method in class java.awt.MediaTracker

Checks to see if all images have finished loading but does not start loading the images if they are not already loading.

checkAll(boolean). Method in class java.awt.MediaTracker

Checks to see if all images have finished loading.

Checkbox(). Constructor for class java.awt.Checkbox

Constructs a Checkbox with no label, no Checkbox group, and initialized to a false state.

Checkbox(String). Constructor for class java.awt.Checkbox

Constructs a Checkbox with the specified label, no Checkbox group, and initialized to a false state.

Checkbox(String, CheckboxGroup, boolean). Constructor for class java.awt. Checkbox

Constructs a Checkbox with the specified label, specified Checkbox group, and specified boolean state.

CheckboxGroup(). Constructor for class java.awt.CheckboxGroup

Creates a new CheckboxGroup.

CheckboxMenuItem(String). Constructor for class java.awt.CheckboxMenuItem

Creates the checkbox item with the specified label.

checkConnect(String, int). Method in class java.lang.SecurityManager

Checks to see if a socket has connected to the specified port on the the specified host.

checkConnect(String, int, Object). Method in class java.lang.SecurityManager

Checks to see if the current execution context and the indicated execution context are both allowed

to connect to the indicated host and port.

checkCreateClassLoader(). Method in class java.lang.SecurityManager

Checks to see if the ClassLoader has been created.

checkDelete(String). Method in class java.lang.SecurityManager

Checks to see if a file with the specified system dependent file name can be deleted.

checkError(). Method in class java.io.PrintStream

Flushes the print stream and returns whether or not there was an error on the output stream.

checkExec(String). Method in class java.lang.SecurityManager

Checks to see if the system command is executed by trusted code.

checkExit(int). Method in class java.lang.SecurityManager

Checks to see if the system has exited the virtual machine with an exit code.

checkID(int). Method in class java.awt.MediaTracker

Checks to see if all images tagged with the indicated ID have finished loading, but does not start loading the images if they are not already loading.

checkID(int, boolean). Method in class java.awt.MediaTracker

Checks to see if all images tagged with the indicated ID have finished loading.

checkImage(Image, ImageObserver). Method in class java.awt.Component

Returns the status of the construction of a screen representation of the specified image.

checkImage(Image, int, int, ImageObserver). Method in class java.awt.Component

Returns the status of the construction of a scaled screen representation of the specified image.

checkImage(Image, int, int, ImageObserver). Method in interface java.awt.peer.ComponentPeer

checkImage(Image, int, int, ImageObserver). Method in class java.awt.Toolkit

Returns the status of the construction of the indicated method at the indicated width and height for the default screen.

checkLink(String). Method in class java.lang.SecurityManager

Checks to see if the specified linked library exists.

checkListen(int). Method in class java.lang.SecurityManager

Checks to see if a server socket is listening to the specified local port that it is bounded to.

checkPackageAccess(String). Method in class java.lang.SecurityManager

Checks to see if an applet can access a package.

checkPackageDefinition(String). Method in class java.lang.SecurityManager

Checks to see if an applet can define classes in a package.

checkPropertiesAccess(). Method in class java.lang.SecurityManager

Checks to see who has access to the System properties.

checkPropertyAccess(String). Method in class java.lang.SecurityManager

Checks to see who has access to the System property named by *key*.

checkPropertyAccess(String, String). Method in class java.lang.SecurityManager

Checks to see who has access to the System property named by *key* and *def*.

checkRead(FileDescriptor). Method in class java.lang.SecurityManager

Checks to see if an input file with the specified file descriptor object gets created.

checkRead(String). Method in class java.lang.SecurityManager

Checks to see if an input file with the specified system dependent file name gets created.

checkRead(String, Object). Method in class java.lang.SecurityManager

Checks to see if the current context or the indicated context are both allowed to read the given file name.

checkSetFactory(). Method in class java.lang.SecurityManager

Checks to see if an applet can set a networking-related object factory.

checkTopLevelWindow(Object). Method in class java.lang.SecurityManager

Checks to see if top-level windows can be created by the caller.

checkWrite(FileDescriptor). Method in class java.lang.SecurityManager

Checks to see if an output file with the specified file descriptor object gets created.

checkWrite(String). Method in class java.lang.SecurityManager

Checks to see if an output file with the specified system dependent file name gets created.

Choice(). Constructor for class java.awt.Choice

Constructs a new Choice.

ClassCastException(). Constructor for class java.lang.ClassCastException

Constructs a ClassCastException with no detail message.

ClassCastException(String). Constructor for class java.lang.ClassCastException
Constructs a ClassCastException with the specified detail message.

ClassCircularityError(). Constructor for class java.lang.ClassCircularityError
Constructs a ClassCircularityError with no detail message.

ClassCircularityError(String). Constructor for class java.lang.ClassCircularityError
Constructs a ClassCircularityError with the specified detail message.

classDepth(String). Method in class java.lang.SecurityManager
Return the position of the stack frame containing the first occurrence of the named class.

ClassFormatError(). Constructor for class java.lang.ClassFormatError
Constructs a ClassFormatError with no detail message.

ClassFormatError(String). Constructor for class java.lang.ClassFormatError
Constructs a ClassFormatError with the specified detail message.

ClassLoader(). Constructor for class java.lang.ClassLoader
Constructs a new Class loader and initializes it.

ClassLoaderDepth(). Method in class java.lang.SecurityManager

ClassNotFoundException(). Constructor for class java.lang.ClassNotFoundException
Constructs a ClassNotFoundException with no detail message.

ClassNotFoundException(String). Constructor for class java.lang.ClassNotFoundException
Constructs a ClassNotFoundException with the specified detail message.

clear(). Method in class java.util.Hashtable
Clears the hash table so that it has no more elements in it.

clear(). Method in class java.awt.List
Clears the list.

clear(). Method in interface java.awt.peer.ListPeer

clear(int). Method in class java.util.BitSet
Clears a bit.

clearBreakpoint(int). Method in class sun.tools.debug.RemoteClass
Clear a breakpoint at a specific address in a class.

clearBreakpointLine(int). Method in class sun.tools.debug.RemoteClass
Clear a breakpoint at a specified line.

clearBreakpointMethod(RemoteField). Method in class sun.tools.debug.RemoteClass
Clear a breakpoint at the start of a specified method.

clearChanged(). Method in class java.util.Observable
Clears an observable change.

clearRect(int, int, int, int). Method in class java.awt.Graphics
Clears the specified rectangle by filling it with the current background color of the current drawing surface.

clickCount. Variable in class java.awt.Event
The number of consecutive clicks.

clipRect(int, int, int, int). Method in class java.awt.Graphics
Clips to a rectangle.

clone(). Method in class java.util.BitSet
Clones the BitSet.

clone(). Method in class java.awt.GridBagConstraints
Creates a clone of the object.

clone(). Method in class java.util.Hashtable
Creates a clone of the hashtable.

clone(). Method in class java.awt.image.ImageFilter
Clones this object.

clone(). Method in class java.awt.Insets
Creates a clone of the object.

clone(). Method in class java.lang.Object
Creates a clone of the object.

clone(). Method in class java.util.Vector
Clones this vector.

CloneNotSupportedException(). Constructor for class java.lang. CloneNotSupportedException

Constructs an CloneNotSupportedException with no detail message.

CloneNotSupportedException(String). Constructor for class java.lang. CloneNotSupportedException

Constructs an CloneNotSupportedException with the specified detail message.

close(). Method in class java.net. DatagramSocket

Closes the datagram socket.

close(). Method in class java.io. FileInputStream

Closes the input stream.

close(). Method in class java.io. FileOutputStream

Closes the stream.

close(). Method in class java.io. FilterInputStream

Closes the input stream.

close(). Method in class java.io. FilterOutputStream

Closes the stream.

close(). Method in class java.io. InputStream

Closes the input stream.

close(). Method in class java.io. OutputStream

Closes the stream.

close(). Method in class java.io. PipedInputStream

Closes the input stream.

close(). Method in class java.io. PipedOutputStream

Closes the stream.

close(). Method in class java.io. PrintStream

Closes the stream.

close(). Method in class java.io. RandomAccessFile

Closes the file.

close(). Method in class sun.tools.debug. RemoteDebugger

Closes the connection to the remote debugging agent.

close(). Method in class java.io. SequenceInputStream

Closes the input stream; flipping to the next stream, if an EOF is reached.

close(). Method in class java.net. ServerSocket

Closes the server socket.

close(). Method in class java.net. Socket

Closes the socket.

close(). Method in class java.net. SocketImpl

Closes the socket.

Color(float, float, float). Constructor for class java.awt. Color

Creates a color with the specified red, green, and blue values in the range (0.0 - 1.0).

Color(int). Constructor for class java.awt. Color

Creates a color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7.

Color(int, int, int). Constructor for class java.awt. Color

Creates a color with the specified red, green, and blue values in the range (0 - 255).

ColorModel(int). Constructor for class java.awt.image. ColorModel

Constructs a ColorModel which describes a pixel of the specified number of bits.

columnWeights. Variable in class java.awt. GridBagLayout

columnWidths. Variable in class java.awt. GridBagLayout

command(Object). Static method in class java.lang. Compiler

commentChar(int). Method in class java.io. StreamTokenizer

Specifies that this character starts a single line comment.

compareTo(String). Method in class java.lang. String

Compares this String to another specified String.

compileClass(Class). Static method in class java.lang. Compiler

compileClasses(String). Static method in class java.lang. Compiler

COMPLETE. Static variable in class java.awt. MediaTracker

Flag indicating the download of media completed successfully.

COMPLETESCANLINES. Static variable in interface java.awt.image.ImageConsumer

The pixels will be delivered in (multiples of) complete scanlines at a time.

comptable. Variable in class java.awt.GridBagLayout

concat(String). Method in class java.lang.String

Concatenates the specified string to the end of this String.

connect(()). Method in class java.net.URLConnection

URLConnection objects go through two phases: first they are created, then they are connected.

connect(InetAddress, int). Method in class java.net.SocketImpl

Connects the socket to the specified address on the specified port.

connect(PipedInputStream). Method in class java.io.PipedOutputStream

Connect this output stream to a receiver.

connect(PipedOutputStream). Method in class java.io.PipedInputStream

Connects this input stream to a sender.

connect(String, int). Method in class java.net.SocketImpl

Connects the socket to the specified port on the specified host.

connected. Variable in class java.net.URLConnection

consumer. Variable in class java.awt.image.ImageFilter

The consumer of the particular image data stream for which this instance of the ImageFilter is filtering data.

cont(()). Method in class sun.tools.debug.RemoteThread

Resume this thread from a breakpoint, unless it previously suspended.

contains(Object). Method in class java.util.Hashtable

Returns true if the specified object is an element of the hashtable.

contains(Object). Method in class java.util.Vector

Returns true if the specified object is a value of the collection.

containsKey(Object). Method in class java.util.Hashtable

Returns true if the collection contains an element for the key.

ContentHandler(()). Constructor for class java.net.ContentHandler

controlDown(()). Method in class java.awt.Event

Checks if the control key is down.

copyArea(int, int, int, int, int, int). Method in class java.awt.Graphics

Copies an area of the screen.

copyInto(Object[]). Method in class java.util.Vector

Copies the elements of this vector into the specified array.

copyValueOf(char[]). Static method in class java.lang.String

Returns a String that is equivalent to the specified character array.

copyValueOf(char[], int, int). Static method in class java.lang.String

Returns a String that is equivalent to the specified character array.

cos(double). Static method in class java.lang.Math

Returns the trigonometric cosine of an angle.

count. Variable in class java.io.BufferedInputStream

The number of bytes in the buffer.

count. Variable in class java.io.BufferedOutputStream

The number of bytes in the buffer.

count. Variable in class java.io.ByteArrayInputStream

The number of characters to use in the buffer.

count. Variable in class java.io.ByteArrayOutputStream

The number of bytes in the buffer.

count. Variable in class java.io.StringBufferInputStream

The number of characters to use in the buffer.

countComponents(()). Method in class java.awt.Container

Returns the number of components in this panel.

countItems(()). Method in class java.awt.Choice

Returns the number of items in this Choice.

countItems(()). Method in class java.awt.List

Returns the number of items in the list.

countItems(). Method in class java.awt.Menu

Returns the number of elements in this menu.

countMenus(). Method in class java.awt.MenuBar

Counts the number of menus on the menu bar.

countObservers(). Method in class java.util.Observable

Counts the number of observers.

countStackFrames(). Method in class java.lang.Thread

Returns the number of stack frames in this Thread.

countTokens(). Method in class java.util.StringTokenizer

Returns the next number of tokens in the String using the current delimiter set.

create(). Method in class java.awt.Graphics

Creates a new Graphics Object that is a copy of the original Graphics Object.

create(boolean). Method in class java.net.SocketImpl

Creates a socket with a boolean that specifies whether this is a stream socket or a datagram socket.

create(int, int, int, int). Method in class java.awt.Graphics

Creates a new Graphics Object with the specified parameters, based on the original Graphics Object.

createButton(Button). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Button.

createCanvas(Canvas). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Canvas.

createCheckbox(Checkox). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Checkbox.

createCheckboxMenuItem(CheckboxMenuItem). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new CheckboxMenuItem.

createChoice(Choice). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Choice.

createContentHandler(String). Method in interface java.net.ContentHandlerFactory

Creates a new ContentHandler to read an object from a URLStreamHandler.

createDialog(Dialog). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Dialog.

createFileDialog(FileDialog). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new FileDialog.

createFrame(Frame). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Frame.

createImage(ImageProducer). Method in class java.awt.Component

Creates an image from the specified image producer.

createImage(ImageProducer). Method in interface java.awt.peer.ComponentPeer

createImage(ImageProducer). Method in class java.awt.Toolkit

Creates an image with the specified image producer.

createImage(int, int). Method in class java.awt.Component

Creates an off-screen drawable Image to be used for double buffering.

createImage(int, int). Method in interface java.awt.peer.ComponentPeer

createLabel(Label). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Label.

createList(List). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new List.

createMenu(Menu). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Menu.

createMenuBar(MenuBar). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new MenuBar.

createMenuItem(MenuItem). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new MenuItem.

createPanel(Panel). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Panel.

createScrollbar(Scrollbar). Method in class java.awt.Toolkit

Uses the specified Peer interface to create a new Scrollbar.

createSocketImpl(). Method in interface [java.net.SocketImplFactory](#)
Creates a new SocketImpl instance.

createTextArea(TextArea). Method in class [java.awt.Toolkit](#)
Uses the specified Peer interface to create a new TextArea.

createTextField(TextField). Method in class [java.awt.Toolkit](#)
Uses the specified Peer interface to create a new TextField.

createURLStreamHandler(String). Method in interface [java.net.URLStreamHandlerFactory](#)
Creates a new URLStreamHandler instance with the specified protocol.

createWindow(Window). Method in class [java.awt.Toolkit](#)
Uses the specified Peer interface to create a new Window.

CropImageFilter(int, int, int, int). Constructor for class [java.awt.image.CropImageFilter](#)
Constructs a CropImageFilter that extracts the absolute rectangular region of pixels from its source Image as specified by the x, y, w, and h parameters.

CROSSHAIR_CURSOR. Static variable in class [java.awt.Frame](#)

CTRL_MASK. Static variable in class [java.awt.Event](#)
The control modifier constant.

currentClassLoader(). Method in class [java.lang.SecurityManager](#)
The current ClassLoader on the execution stack.

currentThread(). Static method in class [java.lang.Thread](#)
Returns a reference to the currently executing Thread object.

currentTimeMillis(). Static method in class [java.lang.System](#)
Returns the current time in milliseconds GMT since the epoch (00:00:00 UTC, January 1, 1970).

cyan. Static variable in class [java.awt.Color](#)
The color cyan.

D

darker(). Method in class [java.awt.Color](#)
Returns a darker version of this color.

darkGray. Static variable in class [java.awt.Color](#)
The color dark gray.

DatagramPacket(byte[], int). Constructor for class [java.net.DatagramPacket](#)
This constructor is used to create a DatagramPacket object used for receiving datagrams.

DatagramPacket(byte[], int, InetAddress, int). Constructor for class [java.net.DatagramPacket](#)
This constructor is used to construct the DatagramPacket to be sent.

DatagramSocket(). Constructor for class [java.net.DatagramSocket](#)
Creates a datagram socket

DatagramSocket(int). Constructor for class [java.net.DatagramSocket](#)
Creates a datagram socket

DataInputStream(InputStream). Constructor for class [java.io.DataInputStream](#)
Creates a new DataInputStream.

DataOutputStream(OutputStream). Constructor for class [java.io.DataOutputStream](#)
Creates a new DataOutputStream.

Date(). Constructor for class [java.util.Date](#)
Creates today's date/time.

Date(int, int, int). Constructor for class [java.util.Date](#)
Creates a date.

Date(int, int, int, int, int). Constructor for class [java.util.Date](#)
Creates a date.

Date(int, int, int, int, int, int). Constructor for class [java.util.Date](#)
Creates a date.

Date(long). Constructor for class [java.util.Date](#)
Creates a date.

Date(String). Constructor for class java.util.Date
Creates a date from a string according to the syntax accepted by parse().

DEFAULT_CURSOR. Static variable in class java.awt.Frame

defaultConstraints. Variable in class java.awt. GridBagLayout

defaults. Variable in class java.util.Properties

defineClass(byte[], int, int). Method in class java.lang.ClassLoader
Converts an array of bytes to an instance of class Class.

delete(). Method in class java.io.File
Deletes the specified file.

deleteObserver(Observer). Method in class java.util.Observable
Deletes an observer from the observer list.

deleteObservers(). Method in class java.util.Observable
Deletes observers from the observer list.

dellItem(int). Method in class java.awt.List
Delete an item from the list.

dellItem(int). Method in interface java.awt.peer.MenuPeer

dellItems(int, int). Method in class java.awt.List
Delete multiple items from the list.

dellItems(int, int). Method in interface java.awt.peer.ListPeer

deliverEvent(Event). Method in class java.awt.Component
Delivers an event to this component or one of its sub components.

deliverEvent(Event). Method in class java.awt.Container
Delivers an event.

delMenu(int). Method in interface java.awt.peer.MenuBarPeer

description(). Method in class sun.tools.debug. RemoteArray
Return a description of the array.

description(). Method in class sun.tools.debug. RemoteClass
Return a (somewhat verbose) description.

description(). Method in class sun.tools.debug. RemoteObject
Return a description of the object.

description(). Method in class sun.tools.debug. RemoteString
Return the string value, or "null"

description(). Method in class sun.tools.debug. RemoteValue
Return a description of the RemoteValue.

deselect(int). Method in class java.awt.List
Deselects the item at the specified index.

deselect(int). Method in interface java.awt.peer.ListPeer

destroy(). Method in class java.applet.Applet
Cleans up whatever resources are being held.

destroy(). Method in class java.lang.Process
Kills the subprocess.

destroy(). Method in class java.lang.Thread
Destroy a thread, without any cleanup, i.e.

destroy(). Method in class java.lang.ThreadGroup
Destroys a Thread group.

Dialog(Frame, boolean). Constructor for class java.awt.Dialog
Constructs an initially invisible Dialog.

Dialog(Frame, String, boolean). Constructor for class java.awt.Dialog
Constructs an initially invisible Dialog with a title.

Dictionary(). Constructor for class java.util.Dictionary

digit(char, int). Static method in class java.lang. Character
Returns the numeric value of the character digit using the specified radix.

Dimension(). Constructor for class java.awt.Dimension
Constructs a Dimension with a 0 width and 0 height.

Dimension(Dimension). Constructor for class java.awt.Dimension
Constructs a Dimension and initializes it to the specified value.

Dimension(int, int). Constructor for class java.awt.Dimension

Constructs a Dimension and initializes it to the specified width and specified height.

DirectColorModel(int, int, int, int). Constructor for class java.awt.image. DirectColorModel

Constructs a DirectColorModel from the given masks specifying which bits in the pixel contain the red, green and blue color components.

DirectColorModel(int, int, int, int, int). Constructor for class java.awt.image. DirectColorModel

Constructs a DirectColorModel from the given masks specifying which bits in the pixel contain the alpha, red, green and blue color components.

disable(). Static method in class java.lang.Compiler

disable(). Method in class java.awt.Component

Disables a component.

disable(). Method in interface java.awt.peer.ComponentPeer

disable(). Method in class java.awt.MenuItem

Makes this menu item unselectable by the user.

disable(). Method in interface java.awt.peer.MenuItemPeer

dispose(). Method in interface java.awt.peer. ComponentPeer

dispose(). Method in class java.awt.Frame

Disposes of the Frame.

dispose(). Method in class java.awt.Graphics

Disposes of this graphics context.

dispose(). Method in interface java.awt.peer.MenuComponentPeer

dispose(). Method in class java.awt.Window

Disposes of the Window.

doInput. Variable in class java.net.URLConnection

doOutput. Variable in class java.net.URLConnection

Double(double). Constructor for class java.lang.Double

Constructs a Double wrapper for the specified double value.

Double(String). Constructor for class java.lang.Double

Constructs a Double object initialized to the value specified by the String parameter.

doubleToLongBits(double). Static method in class java.lang.Double

Returns the bit representation of a double-float value

doubleValue(). Method in class java.lang.Double

Returns the double value of this Double.

doubleValue(). Method in class java.lang.Float

Returns the double value of this Float.

doubleValue(). Method in class java.lang.Integer

Returns the value of this Integer as a double.

doubleValue(). Method in class java.lang.Long

Returns the value of this Long as a double.

doubleValue(). Method in class java.lang.Number

Returns the value of the number as a double.

DOWN. Static variable in class java.awt.Event

The down arrow key.

down(int). Method in class sun.tools.debug.RemoteThread

Change the current stackframe to be one or more frames lower (as in, toward the current program counter).

draw3DRect(int, int, int, int, boolean). Method in class java.awt.Graphics

Draws a highlighted 3-D rectangle.

drawArc(int, int, int, int, int, int). Method in class java.awt.Graphics

Draws an arc bounded by the specified rectangle from startAngle to endAngle.

drawBytes(byte[], int, int, int, int). Method in class java.awt.Graphics

Draws the specified bytes using the current font and color.

drawChars(char[], int, int, int, int). Method in class java.awt.Graphics

Draws the specified characters using the current font and color.

drawImage(Image, int, int, Color, ImageObserver). Method in class java.awt. Graphics

Draws the specified image at the specified coordinate (x, y), with the given solid background Color.

drawImage(Image, int, int, ImageObserver). Method in class java.awt.Graphics
Draws the specified image at the specified coordinate (x, y).

drawImage(Image, int, int, int, int, Color, ImageObserver). Method in class java.awt.Graphics
Draws the specified image inside the specified rectangle, with the given solid background Color.

drawImage(Image, int, int, int, int, ImageObserver). Method in class java.awt. Graphics
Draws the specified image inside the specified rectangle.

drawLine(int, int, int, int). Method in class java.awt.Graphics
Draws a line between the coordinates (x1,y1) and (x2,y2).

drawOval(int, int, int, int). Method in class java.awt.Graphics
Draws an oval inside the specified rectangle using the current color.

drawPolygon(int[], int[], int). Method in class java.awt.Graphics
Draws a polygon defined by an array of x points and y points.

drawPolygon(Polygon). Method in class java.awt.Graphics
Draws a polygon defined by the specified point.

drawRect(int, int, int, int). Method in class java.awt.Graphics
Draws the outline of the specified rectangle using the current color.

drawRoundRect(int, int, int, int, int, int). Method in class java.awt.Graphics
Draws an outlined rounded corner rectangle using the current color.

drawString(String, int, int). Method in class java.awt.Graphics
Draws the specified String using the current font and color.

DumpConstraints(GridBagConstraints). Method in class java.awt.GridBagLayout
Print the layout constraints.

DumpLayoutInfo(GridBagLayoutInfo). Method in class java.awt.GridBagLayout
Print the layout information.

dumpStack(()). Method in class sun.tools.debug.RemoteThread
Dump the stack.

dumpStack(()). Static method in class java.lang.Thread
A debugging procedure to print a stack trace for the current Thread.

E

E. Static variable in class java.lang.Math
The float representation of the value E.

E_RESIZE_CURSOR. Static variable in class java.awt.Frame

EAST. Static variable in class java.awt. GridBagConstraints

echoCharSet(()). Method in class java.awt.TextField
Returns true if this TextField has a character set for echoing.

elementAt(int). Method in class java.util.Vector
Returns the element at the specified index.

elementCount. Variable in class java.util.Vector
The number of elements in the buffer.

elementData. Variable in class java.util.Vector
The buffer where elements are stored.

elements(()). Method in class java.util.Dictionary
Returns an enumeration of the elements.

elements(()). Method in class java.util.Hashtable
Returns an enumeration of the elements.

elements(()). Method in class java.util.Vector
Returns an enumeration of the elements.

empty(()). Method in class java.util.Stack
Returns true if the stack is empty.

EmptyStackException(()). Constructor for class java.util.EmptyStackException
Constructs a new EmptyStackException with no detail message.

enable(). Static method in class java.lang.Compiler

enable(). Method in class java.awt.Component

Enables a component.

enable(). Method in interface java.awt.peer.ComponentPeer

enable(). Method in class java.awt.MenuItem

Makes this menu item selectable by the user.

enable(). Method in interface java.awt.peer.MenuItemPeer

enable(boolean). Method in class java.awt.Component

Conditionally enables a component.

enable(boolean). Method in class java.awt.MenuItem

Conditionally enables a component.

encode(String). Static method in class java.net.URLConnection

Translates String into x-www-form-urlencoded format.

END. Static variable in class java.awt.Event

The end key.

endsWith(String). Method in class java.lang.String

Determines whether the String ends with some suffix.

ensureCapacity(int). Method in class java.lang.StringBuffer

Ensures that the capacity of the buffer is at least equal to the specified minimum.

ensureCapacity(int). Method in class java.util.Vector

Ensures that the vector has at least the specified capacity.

enumerate(Thread[]). Static method in class java.lang.Thread

Copies, into the specified array, references to every active Thread in this Thread's group.

enumerate(Thread[]). Method in class java.lang.ThreadGroup

Copies, into the specified array, references to every active Thread in this Thread group.

enumerate(Thread[], boolean). Method in class java.lang.ThreadGroup

Copies, into the specified array, references to every active Thread in this Thread group.

enumerate(ThreadGroup[]). Method in class java.lang.ThreadGroup

Copies, into the specified array, references to every active Thread group in this Thread group.

enumerate(ThreadGroup[], boolean). Method in class java.lang.ThreadGroup

Copies, into the specified array, references to every active Thread group in this Thread group.

EOFException(). Constructor for class java.io.EOFException

Constructs an EOFException with no detail message.

EOFException(String). Constructor for class java.io.EOFException

Constructs an EOFException with the specified detail message.

eofIsSignificant(boolean). Method in class java.io.StreamTokenizer

If the flag is true, end-of-lines are significant (TT_EOL will be returned by nexttoken).

equals(Object). Method in class java.util.BitSet

Compares this object against the specified object.

equals(Object). Method in class java.lang.Boolean

Compares this object against the specified object.

equals(Object). Method in class java.lang.Character

Compares this object against the specified object.

equals(Object). Method in class java.awt.Color

Compares this object against the specified object.

equals(Object). Method in class java.util.Date

Compares this object against the specified object.

equals(Object). Method in class java.lang.Double

Compares this object against the specified object.

equals(Object). Method in class java.io.File

Compares this object against the specified object.

equals(Object). Method in class java.lang.Float

Compares this object against some other object.

equals(Object). Method in class java.awt.Font

Compares this object to the specified object.

equals(Object). Method in class java.net.InetAddress

Compares this object against the specified object.
equals(Object). Method in class java.lang.Integer
Compares this object to the specified object.
equals(Object). Method in class java.lang.Long
Compares this object against the specified object.
equals(Object). Method in class java.lang.Object
Compares two Objects for equality.
equals(Object). Method in class java.awt.Point
Checks whether two pointers are equal.
equals(Object). Method in class java.awt.Rectangle
Checks whether two rectangles are equal.
equals(Object). Method in class java.lang.String
Compares this String to the specified object.
equals(Object). Method in class java.net.URL
Compares two URLs.
equalsIgnoreCase(String). Method in class java.lang.String
Compares this String to another object.
err. Static variable in class java.io.FileDescriptor
Handle to standard error.
err. Static variable in class java.lang.System
Standard error stream.
ERROR. Static variable in interface java.awt.image.ImageObserver
An image which was being tracked asynchronously has encountered an error.
Error(). Constructor for class java.lang.Error
Constructs an Error with no specified detail message.
Error(String). Constructor for class java.lang.Error
Constructs an Error with the specified detail message.
ERRORED. Static variable in class java.awt.MediaTracker
Flag indicating the download of some media encountered an error.
Event(Object, int, Object). Constructor for class java.awt.Event
Constructs an event with the specified target component, event type, and argument.
Event(Object, long, int, int, int, int, int). Constructor for class java.awt. Event
Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and an argument set to null.
Event(Object, long, int, int, int, int, int, Object). Constructor for class java.awt.Event
Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and argument.
evt. Variable in class java.awt.Event
The next event.
Exception(). Constructor for class java.lang.Exception
Constructs an Exception with no specified detail message.
Exception(String). Constructor for class java.lang.Exception
Constructs a Exception with the specified detail message.
exceptionEvent(RemoteThread, String). Method in interface sun.tools.debug. DebuggerCallback
An exception has occurred.
exec(String). Method in class java.lang.Runtime
Executes the system command specified in the parameter.
exec(String, String[]). Method in class java.lang.Runtime
Executes the system command specified in the parameter.
exec(String[]). Method in class java.lang.Runtime
Executes the system command specified by cmdarray[0] with arguments specified by the strings in the rest of the array.
exec(String[], String[]). Method in class java.lang.Runtime
Executes the system command specified by cmdarray[0] with arguments specified by the strings in the rest of the array.
exists(). Method in class java.io.File

Returns a boolean indicating whether or not a file exists.

exit(int). Method in class java.lang.Runtime

Exits the virtual machine with an exit code.

exit(int). Static method in class java.lang.System

Exits the virtual machine with an exit code.

exitValue(). Method in class java.lang.Process

Returns the exit value for the subprocess.

exp(double). Static method in class java.lang.Math

Returns the exponential number e(2.718...) raised to the power of a.

F

F1. Static variable in class java.awt.Event

The F1 function key.

F10. Static variable in class java.awt.Event

The F10 function key.

F11. Static variable in class java.awt.Event

The F11 function key.

F12. Static variable in class java.awt.Event

The F12 function key.

F2. Static variable in class java.awt.Event

The F2 function key.

F3. Static variable in class java.awt.Event

The F3 function key.

F4. Static variable in class java.awt.Event

The F4 function key.

F5. Static variable in class java.awt.Event

The F5 function key.

F6. Static variable in class java.awt.Event

The F6 function key.

F7. Static variable in class java.awt.Event

The F7 function key.

F8. Static variable in class java.awt.Event

The F8 function key.

F9. Static variable in class java.awt.Event

The F9 function key.

FALSE. Static variable in class java.lang.Boolean

Assigns this Boolean to be false.

fd. Variable in class java.net.SocketImpl

The file descriptor object

File(File, String). Constructor for class java.io.File

Creates a File object (given a directory File object).

File(String). Constructor for class java.io.File

Creates a File object.

File(String, String). Constructor for class java.io.File

Creates a File object from the specified directory.

FileDescriptor(). Constructor for class java.io.FileDescriptor

FileDialog(Frame, String). Constructor for class java.awt.FileDialog

Creates a file dialog for loading a file.

FileDialog(Frame, String, int). Constructor for class java.awt.FileDialog

Creates a file dialog with the specified title and mode.

FileInputStream(File). Constructor for class java.io.FileInputStream

Creates an input file from the specified File object.

FileInputStream(FileDescriptor). Constructor for class java.io.FileInputStream

FileInputStream(String). Constructor for class java.io.FileInputStream

Creates an input file with the specified system dependent file name.

FileNotFoundException(). Constructor for class java.io.FileNotFoundException

Constructs a FileNotFoundException with no detail message.

FileNotFoundException(String). Constructor for class java.io. FileNotFoundException

Constructs a FileNotFoundException with the specified detail message.

FileOutputStream(File). Constructor for class java.io.FileOutputStream

Creates an output file with the specified File object.

FileOutputStream(FileDescriptor). Constructor for class java.io. FileOutputStream

FileOutputStream(String). Constructor for class java.io.FileOutputStream

Creates an output file with the specified system dependent file name.

fill. Variable in class java.awt.GridBagConstraints

fill3DRect(int, int, int, int, boolean). Method in class java.awt.Graphics

Paints a highlighted 3-D rectangle using the current color.

fillArc(int, int, int, int, int, int). Method in class java.awt.Graphics

Fills an arc using the current color.

fillInStackTrace(). Method in class java.lang.Throwable

Fills in the execution stack trace.

fillOval(int, int, int, int). Method in class java.awt.Graphics

Fills an oval inside the specified rectangle using the current color.

fillPolygon(int[], int[], int). Method in class java.awt.Graphics

Fills a polygon with the current color using an even-odd fill rule (otherwise known as an alternating rule).

fillPolygon(Polygon). Method in class java.awt.Graphics

Fills the specified polygon with the current color using an even-odd fill rule (otherwise known as an alternating rule).

fillRect(int, int, int, int). Method in class java.awt.Graphics

Fills the specified rectangle with the current color.

fillRoundRect(int, int, int, int, int, int). Method in class java.awt.Graphics

Draws a rounded rectangle filled in with the current color.

FilteredImageSource(ImageProducer, ImageFilter). Constructor for class

java.awt.image.FilteredImageSource

Constructs an ImageProducer object from an existing ImageProducer and a filter object.

filterIndexColorModel(IndexColorModel). Method in class java.awt.image. RGBImageFilter

Filters an IndexColorModel object by running each entry in its color tables through the filterRGB function that RGBImageFilter subclasses must provide.

FilterInputStream(InputStream). Constructor for class java.io.FilterInputStream

Creates an input stream filter.

FilterOutputStream(OutputStream). Constructor for class java.io. FilterOutputStream

Creates an output stream filter.

filterRGB(int, int, int). Method in class java.awt.image.RGBImageFilter

Subclasses must specify a method to convert a single input pixel in the default RGB ColorModel to a single output pixel.

filterRGBPixels(int, int, int, int, int[], int, int). Method in class java.awt.image.RGBImageFilter

Filters a buffer of pixels in the default RGB ColorModel by passing them one by one through the filterRGB method.

finalize(). Method in class java.net.DatagramSocket

Code to perform when this object is garbage collected.

finalize(). Method in class java.io.FileInputStream

Closes the stream when garbage is collected.

finalize(). Method in class java.io.FileOutputStream

Closes the stream when garbage is collected.

finalize(). Method in class java.awt.Graphics

Disposes of this graphics context once it is no longer referenced.

finalize(). Method in class java.lang.Object

Code to perform when this object is garbage collected.

findClass(String). Method in class sun.tools.debug.RemoteDebugger
Find a specified class.

findSystemClass(String). Method in class java.lang.ClassLoader
Loads a system Class.

first(Container). Method in class java.awt.CardLayout
Flip to the first card.

firstElement(Vector). Method in class java.util.Vector
Returns the first element of the sequence.

Float(double). Constructor for class java.lang.Float
Constructs a Float wrapper for the specified double value.

Float(float). Constructor for class java.lang.Float
Constructs a Float wrapper for the specified float value.

Float(String). Constructor for class java.lang.Float
Constructs a Float object initialized to the value specified by the String parameter.

floatToIntBits(float). Static method in class java.lang.Float
Returns the bit representation of a single-float value

floatValue(Double). Method in class java.lang.Double
Returns the float value of this Double.

floatValue(Float). Method in class java.lang.Float
Returns the float value of this Float object.

floatValue(Integer). Method in class java.lang.Integer
Returns the value of this Integer as a float.

floatValue(Long). Method in class java.lang.Long
Returns the value of this Long as a float.

floatValue(Number). Method in class java.lang.Number
Returns the value of the number as a float.

floor(double). Static method in class java.lang.Math
Returns the "floor" or largest whole number less than or equal to a.

FlowLayout(int). Constructor for class java.awt.FlowLayout
Constructs a new Flow Layout with a centered alignment.

FlowLayout(int, int). Constructor for class java.awt.FlowLayout
Constructs a new Flow Layout with the specified alignment.

FlowLayout(int, int, int). Constructor for class java.awt.FlowLayout
Constructs a new Flow Layout with the specified alignment and gap values.

flush(BufferedOutputStream). Method in class java.io.BufferedOutputStream
Flushes the stream.

flush(DataOutputStream). Method in class java.io.DataOutputStream
Flushes the stream.

flush(FilterOutputStream). Method in class java.io.FilterOutputStream
Flushes the stream.

flush(Image). Method in class java.awt.Image
Flushes all resources being used by this Image object.

flush(OutputStream). Method in class java.io.OutputStream
Flushes the stream.

flush(PrintStream). Method in class java.io.PrintStream
Flushes the stream.

font. Variable in class java.awt.FontMetrics
The actual font.

Font(String, int, int). Constructor for class java.awt.Font
Creates a new font with the specified name, style and point size.

FontMetrics(Font). Constructor for class java.awt.FontMetrics
Creates a new FontMetrics object with the specified font.

forDigit(int, int). Static method in class java.lang.Character
Returns the character value for the specified digit in the specified radix.

forName(String). Static method in class java.lang.Class

Returns the runtime Class descriptor for the specified Class.

Frame(). Constructor for class java.awt.Frame

Constructs a new Frame that is initially invisible.

Frame(String). Constructor for class java.awt.Frame

Constructs a new, initially invisible Frame with the specified title.

FRAMEBITS. Static variable in interface java.awt.image.ImageObserver

Another complete frame of a multi-frame image which was previously drawn is now available to be drawn again.

freeMemory(). Method in class sun.tools.debug.RemoteDebugger

Report the free memory available to the Java interpreter being debugged.

freeMemory(). Method in class java.lang.Runtime

Returns the number of free bytes in system memory.

fromHex(String). Static method in class sun.tools.debug.RemoteValue

Convert hexadecimal strings to ints.

G

gc(). Method in class java.lang.Runtime

Runs the garbage collector.

gc(). Static method in class java.lang.System

Runs the garbage collector.

gc(RemoteObject[]). Method in class sun.tools.debug.RemoteDebugger

Free all objects referenced by the debugger.

get(). Method in class sun.tools.debug.RemoteBoolean

Return the boolean's value.

get(). Method in class sun.tools.debug.RemoteByte

Return the byte's value.

get(). Method in class sun.tools.debug.RemoteChar

Return the char's value.

get(). Method in class sun.tools.debug.RemoteDouble

Return the double's value.

get(). Method in class sun.tools.debug.RemoteFloat

Return the float's value.

get(). Method in class sun.tools.debug.RemoteInt

Return the int's value.

get(). Method in class sun.tools.debug.RemoteLong

Return the long's value.

get(). Method in class sun.tools.debug.RemoteShort

Return the short's value.

get(int). Method in class java.util.BitSet

Gets a bit.

get(Integer). Method in class sun.tools.debug.RemoteDebugger

Get an object from the remote object cache.

get(Object). Method in class java.util.Dictionary

Gets the object associated with the specified key in the Dictionary.

get(Object). Method in class java.util.Hashtable

Gets the object associated with the specified key in the hashtable.

getAbsolutePath(). Method in class java.io.File

Gets the absolute path of the file.

getAddress(). Method in class java.net.DatagramPacket

getAddress(). Method in class java.net.InetAddress

Returns the raw IP address in network byte order.

getAlignment(). Method in class java.awt.Label

Gets the current alignment of this label.

getAllByName(String). Static method in class java.net.InetAddress

Given a hostname, returns an array of all the corresponding InetAddresses.

getAllowUserInteraction(). Method in class java.net.URLConnection

getAlpha(int). Method in class java.awt.image. ColorModel

The subclass must provide a function which provides the alpha color component for the specified pixel.

getAlpha(int). Method in class java.awt.image.DirectColorModel

Return the alpha transparency value for the specified pixel in the range 0-255.

getAlpha(int). Method in class java.awt.image.IndexColorModel

Returns the alpha transparency value for the specified pixel in the range 0-255.

getAlphaMask(). Method in class java.awt.image.DirectColorModel

Returns the mask indicating which bits in a pixel contain the alpha transparency component.

getAlphas(byte[]). Method in class java.awt.image.IndexColorModel

Copies the array of alpha transparency values into the given array.

getApplet(String). Method in interface java.applet.AppletContext

Gets an applet by name.

getAppletContext(). Method in class java.applet.Applet

Gets a handle to the applet context.

getAppletContext(). Method in interface java.applet.AppletStub

Gets a handler to the applet's context.

getAppletInfo(). Method in class java.applet.Applet

Returns a string containing information about the author, version and copyright of the applet.

getApplets(). Method in interface java.applet.AppletContext

Enumerates the applets in this context.

getAscent(). Method in class java.awt.FontMetrics

Gets the font ascent.

getAudioClip(URL). Method in class java.applet.Applet

Gets an audio clip.

getAudioClip(URL). Method in interface java.applet.AppletContext

Gets an audio clip.

getAudioClip(URL, String). Method in class java.applet.Applet

Gets an audio clip.

getBackground(). Method in class java.awt.Component

Gets the background color.

getBlue(). Method in class java.awt.Color

Gets the blue component.

getBlue(int). Method in class java.awt.image. ColorModel

The subclass must provide a function which provides the blue color component for the specified pixel.

getBlue(int). Method in class java.awt.image.DirectColorModel

Returns the blue color component for the specified pixel in the range 0-255.

getBlue(int). Method in class java.awt.image.IndexColorModel

Returns the blue color component for the specified pixel in the range 0-255.

getBlueMask(). Method in class java.awt.image.DirectColorModel

Returns the mask indicating which bits in a pixel contain the blue color component.

getBlues(byte[]). Method in class java.awt.image.IndexColorModel

Copies the array of blue color components into the given array.

getBoolean(String). Static method in class java.lang.Boolean

Gets a Boolean from the properties.

getBoundingBox(). Method in class java.awt.Polygon

Determines the area spanned by this Polygon.

getByName(String). Static method in class java.net.InetAddress

Returns a network address for the indicated host.

getBytes(int, int, byte[], int). Method in class java.lang.String

Copies characters from this String into the specified byte array.

getChars(int, int, char[], int). Method in class java.lang.String
Copies characters from this String into the specified haracter array.

getChars(int, int, char[], int). Method in class java.lang.StringBuffer
Copies the characters of the specified substring (determined by srcBegin and srcEnd) into the character array, starting at the array's dstBegin location.

getCheckboxGroup(int). Method in class java.awt.Checkbox
Returns the checkbox group.

getClass(int). Method in class java.lang.Object
Returns the Class of this Object.

getClassContext(int). Method in class java.lang.SecurityManager
Gets the context of this Class.

getClassLoader(int). Method in class java.lang.Class
Returns the Class loader of this Class.

getClassLoader(int). Method in class sun.tools.debug.RemoteClass
Return the classloader for this class.

getClasszz(int). Method in class sun.tools.debug.RemoteObject
Returns the object's class.

getClipRect(int). Method in class java.awt.Graphics
Returns the bounding rectangle of the current clipping area.

getCodeBase(int). Method in class java.applet.Applet
Gets the base URL.

getCodeBase(int). Method in interface java.applet.AppletStub
Gets the base URL.

getColor(int). Method in class java.awt.Graphics
Gets the current color.

getColor(String). Static method in class java.awt.Color
Gets the specified Color property.

getColor(String, Color). Static method in class java.awt.Color
Gets the specified Color property of the specified Color.

getColor(String, int). Static method in class java.awt.Color
Gets the specified Color property of the color value.

getColorModel(int). Method in class java.awt.Component
Gets the ColorModel used to display the component on the output device.

getColorModel(int). Method in interface java.awt.peer.ComponentPeer

getColorModel(int). Method in class java.awt.Toolkit
Returns the ColorModel of the screen.

getColumns(int). Method in class java.awt.TextArea
Returns the number of columns in the TextArea.

getColumns(int). Method in class java.awt.TextField
Returns the number of columns in this TextField.

getComponent(int). Method in class java.awt.Container
Gets the nth component in this container.

getComponents(int). Method in class java.awt.Container
Gets all the components in this container.

getConstraints(Component). Method in class java.awt.GridBagLayout
Retrieves the constraints for the specified component.

getContent(int). Method in class java.net.URL
Gets the contents from this opened connection.

getContent(int). Method in class java.net.URLConnection
Gets the object referred to by this URL.

getContent(URLConnection). Method in class java.net.ContentHandler
Given an input stream positioned at the beginning of the representation of an object, reads that stream and recreates the object from it.

getContentEncoding(int). Method in class java.net.URLConnection
Gets the content encoding.

getContentLength(int). Method in class java.net.URLConnection

Gets the content length.

getContentLength(). Method in class java.net.URLConnection

Gets the content type.

getCurrent(). Method in class java.awt.CheckboxGroup

Gets the current choice.

getCurrentFrame(). Method in class sun.tools.debug.RemoteThread

Get the current stack frame.

getCurrentFrameIndex(). Method in class sun.tools.debug.RemoteThread

Return the current stackframe index

getCursorType(). Method in class java.awt.Frame

Return the cursor type

getData(). Method in class java.net.DatagramPacket

getDate(). Method in class java.util.Date

Returns the day of the month.

getDate(). Method in class java.net.URLConnection

Gets the sending date of the object.

getDay(). Method in class java.util.Date

Returns the day of the week.

getDefaultAllowUserInteraction(). Static method in class java.net. URLConnection

getDefaultRequestProperty(String). Static method in class java.net.URLConnection

getDefaultToolkit(). Static method in class java.awt. Toolkit

Returns the default toolkit.

getDefaultUseCaches(). Method in class java.net.URLConnection

Sets/gets the default value of the UseCaches flag.

getDescent(). Method in class java.awt.FontMetrics

Gets the font descent.

getDirectory(). Method in class java.awt.FileDialog

Gets the directory of the Dialog.

getDocumentBase(). Method in class java.applet.Applet

Gets the document URL.

getDocumentBase(). Method in interface java.applet.AppletStub

Gets the document URL.

getDoInput(). Method in class java.net.URLConnection

getDoOutput(). Method in class java.net.URLConnection

getEchoChar(). Method in class java.awt.TextField

Returns the character to be used for echoing.

getElement(int). Method in class sun.tools.debug.RemoteArray

Return an array element.

getElements(). Method in class sun.tools.debug.RemoteArray

Returns a copy of the array as instances of RemoteValue.

getElements(int, int). Method in class sun.tools.debug.RemoteArray

Returns a copy of a portion of the array as instances of RemoteValue.

getElementType(). Method in class sun.tools.debug.RemoteArray

Return the element type as a "TC_" constant, such as "TC_CHAR".

getenv(String). Static method in class java.lang.System

Obsolete.

getErrorsAny(). Method in class java.awt.MediaTracker

Returns a list of all media that have encountered an error.

getErrorsID(int). Method in class java.awt.MediaTracker

Returns a list of media with the specified ID that have encountered an error.

getErrorStream(). Method in class java.lang.Process

Returns the an InputStream connected to the error stream of the child process.

getExceptionCatchList(). Method in class sun.tools.debug.RemoteDebugger

Return the list of the exceptions the debugger will stop on.

getExpiration(). Method in class java.net.URLConnection

Gets the expiration date of the object.

getFamily(). Method in class java.awt.Font
Gets the platform specific family name of the font.

getFD(). Method in class java.io.FileInputStream
Returns the opaque file descriptor object associated with this stream.

getFD(). Method in class java.io.FileOutputStream
Returns the file descriptor associated with this stream.

getFD(). Method in class java.io.RandomAccessFile
Returns the opaque file descriptor object.

getField(int). Method in class sun.tools.debug.RemoteClass
Return the static field, specified by index.

getField(int). Method in class sun.tools.debug.RemoteObject
Return an instance variable, specified by slot number.

getField(String). Method in class sun.tools.debug.RemoteClass
Return the static field, specified by name.

getField(String). Method in class sun.tools.debug.RemoteObject
Return an instance variable, specified by name.

getFields(). Method in class sun.tools.debug.RemoteClass
Return all the static fields for this class.

getFields(). Method in class sun.tools.debug.RemoteObject
Return the instance (non-static) fields of an object.

getFieldValue(int). Method in class sun.tools.debug.RemoteClass
Return the value of a static field, specified by its index

getFieldValue(int). Method in class sun.tools.debug.RemoteObject
Returns the value of an object's instance variable.

getFieldValue(String). Method in class sun.tools.debug.RemoteClass
Return the value of a static field, specified by name.

getFieldValue(String). Method in class sun.tools.debug.RemoteObject
Returns the value of an object's instance variable.

getFile(). Method in class java.awt.FileDialog
Gets the file of the Dialog.

getFile(). Method in class java.net.URL
Gets the file name.

getFileDescriptor(). Method in class java.net.SocketImpl

getFilenameFilter(). Method in class java.awt.FileDialog
Gets the filter.

getFilePointer(). Method in class java.io.RandomAccessFile
Returns the current location of the file pointer.

getFilterInstance(ImageConsumer). Method in class java.awt.image.ImageFilter
Returns a unique instance of an ImageFilter object which will actually perform the filtering for the specified ImageConsumer.

getFont(). Method in class java.awt.Component
Gets the font of the component.

getFont(). Method in class java.awt.FontMetrics
Gets the font.

getFont(). Method in class java.awt.Graphics
Gets the current font.

getFont(). Method in class java.awt.MenuComponent
Gets the font used for this MenuItem.

getFont(). Method in interface java.awt.MenuContainer

getFont(String). Static method in class java.awt.Font
Gets a font from the system properties list.

getFont(String, Font). Static method in class java.awt.Font
Gets the specified font from the system properties list.

getFontList(). Method in class java.awt.Toolkit
Returns the names of the available fonts.

getFontMetrics(). Method in class java.awt.Graphics

Gets the current font metrics.

getFontMetrics(Font). Method in class java.awt.Component

Gets the font metrics for this component.

getFontMetrics(Font). Method in interface java.awt.peer.ComponentPeer

getFontMetrics(Font). Method in class java.awt. Graphics

Gets the current font metrics for the specified font.

getFontMetrics(Font). Method in class java.awt.Toolkit

Returns the screen metrics of the font.

getForeground(). Method in class java.awt.Component

Gets the foreground color.

getGraphics(). Method in class java.awt.Component

Gets a Graphics context for this component.

getGraphics(). Method in interface java.awt.peer.ComponentPeer

getGraphics(). Method in class java.awt.Image

Gets a graphics object to draw into this image.

getGreen(). Method in class java.awt.Color

Gets the green component.

getGreen(int). Method in class java.awt.image.ColorModel

The subclass must provide a function which provides the green color component for the specified pixel.

getGreen(int). Method in class java.awt.image.DirectColorModel

Returns the green color component for the specified pixel in the range 0-255.

getGreen(int). Method in class java.awt.image.IndexColorModel

Returns the green color component for the specified pixel in the range 0-255.

getGreenMask(). Method in class java.awt.image.DirectColorModel

Returns the mask indicating which bits in a pixel contain the green color component.

getGreens(byte[]). Method in class java.awt.image.IndexColorModel

Copies the array of green color components into the given array.

getHeaderField(int). Method in class java.net.URLConnection

Returns the value for the nth header field.

getHeaderField(String). Method in class java.net.URLConnection

Gets a header field by name.

getHeaderFieldDate(String, long). Method in class java.net.URLConnection

Gets a header field by name.

getHeaderFieldInt(String, int). Method in class java.net.URLConnection

Gets a header field by name.

getHeaderFieldKey(int). Method in class java.net.URLConnection

Returns the key for the nth header field.

getHeight(). Method in class java.awt.FontMetrics

Gets the total height of the font.

getHeight(ImageObserver). Method in class java.awt.Image

Gets the actual height of the image.

getHelpMenu(). Method in class java.awt.MenuBar

Gets the help menu on the menu bar.

getHost(). Method in class java.net.URL

Gets the host name.

getHostName(). Method in class java.net.InetAddress

Gets the hostname for this address; also the key in the hashtable.

getHours(). Method in class java.util.Date

Returns the hour.

getHSBColor(float, float, float). Static method in class java.awt.Color

A static Color factory for generating a Color object from HSB values.

getIconImage(). Method in class java.awt.Frame

Returns the icon image for this Frame.

getId(). Method in class sun.tools.debug.RemoteObject

Returns the id of the object.

getIfModifiedSince(). Method in class java.net.URLConnection

getImage(String). Method in class java.awt.Toolkit
Returns an image which gets pixel data from the specified file.

getImage(URL). Method in class java.applet.Applet
Gets an image given a URL.

getImage(URL). Method in interface java.applet.AppletContext
Gets an image.

getImage(URL). Method in class java.awt.Toolkit
Returns an image which gets pixel data from the specified URL.

getImage(URL, String). Method in class java.applet.Applet
Gets an image relative to a URL.

getInCheck(). Method in class java.lang.SecurityManager
Returns whether there is a security check in progress.

getInetAddress(). Method in class java.net.ServerSocket
Gets the address to which the socket is connected.

getInetAddress(). Method in class java.net.Socket
Gets the address to which the socket is connected.

getInetAddress(). Method in class java.net.SocketImpl

getInputStream(). Method in class java.lang.Process
Returns a Stream connected to the output of the child process.

getInputStream(). Method in class java.net.Socket
Gets an InputStream for this socket.

getInputStream(). Method in class java.net.SocketImpl
Gets an InputStream for this socket.

getInputStream(). Method in class java.net.URLConnection
Calls this routine to get an InputStream that reads from the object.

getInstanceField(int). Method in class sun.tools.debug.RemoteClass
Return the instance field, specified by its index.

getInstanceFields(). Method in class sun.tools.debug.RemoteClass
Return all the instance fields for this class.

getInteger(String). Static method in class java.lang.Integer
Gets an Integer property.

getInteger(String, int). Static method in class java.lang.Integer
Gets an Integer property.

getInteger(String, Integer). Static method in class java.lang.Integer
Gets an Integer property.

getInterfaces(). Method in class java.lang.Class
Returns the interfaces of this Class.

getInterfaces(). Method in class sun.tools.debug.RemoteClass
Return the interfaces for this class.

getItem(int). Method in class java.awt.Choice
Returns the String at the specified index in the Choice.

getItem(int). Method in class java.awt.List
Gets the item associated with the specified index.

getItem(int). Method in class java.awt.Menu
Returns the item located at the specified index of this menu.

getLabel(). Method in class java.awt.Button
Gets the label of the button.

getLabel(). Method in class java.awt.Checkbox
Gets the label of the button.

getLabel(). Method in class java.awt.MenuItem
Gets the label for this menu item.

getLastModified(). Method in class java.net.URLConnection
Gets the last modified date of the object.

getLayout(). Method in class java.awt.Container
Gets the layout manager for this container.

getLayoutDimensions(). Method in class java.awt.GridBagLayout
GetLayoutInfo(Container, int). Method in class java.awt.GridBagLayout
getLayoutOrigin(). Method in class java.awt. GridBagLayout
getLayoutWeights(). Method in class java.awt. GridBagLayout
getLeading(). Method in class java.awt.FontMetrics
Gets the standard leading, or line spacing, for the font.
getLength(). Method in class java.net.DatagramPacket
getLineIncrement(). Method in class java.awt.Scrollbar
Gets the line increment for this scrollbar.
getLineNumber(). Method in class java.io.LineNumberInputStream
Returns the current line number.
getLineNumber(). Method in class sun.tools.debug.RemoteStackFrame
Return the source file line number.
getLocalHost(). Static method in class java.net.InetAddress
Returns the local host.
getLocalizedInputStream(InputStream). Method in class java.lang.Runtime
Localize an input stream.
getLocalizedOutputStream(OutputStream). Method in class java.lang.Runtime
Localize an output stream.
getLocalPort(). Method in class java.net.DatagramSocket
Returns the local port that this socket is bound to.
getLocalPort(). Method in class java.net.ServerSocket
Gets the port on which the socket is listening.
getLocalPort(). Method in class java.net.Socket
Gets the local port to which the socket is connected.
getLocalPort(). Method in class java.net.SocketImpl
getLocalVariable(String). Method in class sun.tools.debug.RemoteStackFrame
Return a specific (named) stack variable.
getLocalVariables(). Method in class sun.tools.debug.RemoteStackFrame
Return an array of all valid local variables and method arguments for this stack frame.
getLong(String). Static method in class java.lang.Long
Get a Long property.
getLong(String, Long). Static method in class java.lang.Long
Get a Long property.
getLong(String, long). Static method in class java.lang.Long
Get a Long property.
getMapSize(). Method in class java.awt.image.IndexColorModel
Returns the size of the color component arrays in this IndexColorModel.
getMaxAdvance(). Method in class java.awt.FontMetrics
Gets the maximum advance width of any character in this Font.
getMaxAscent(). Method in class java.awt.FontMetrics
Gets the maximum ascent of all characters in this Font.
getMaxDecent(). Method in class java.awt.FontMetrics
For backward compatibility only.
getMaxDescent(). Method in class java.awt.FontMetrics
Gets the maximum descent of all characters.
getMaximum(). Method in class java.awt.Scrollbar
Returns the maximum value of this Scrollbar.
getMaxPriority(). Method in class java.lang.ThreadGroup
Gets the maximum priority of the group.
getMenu(int). Method in class java.awt.MenuBar
Gets the specified menu.
getMenuBar(). Method in class java.awt.Frame
Gets the menu bar for this Frame.
getMessage(). Method in class java.lang.Throwable
Gets the detail message of the Throwable.

getMethod(String). Method in class sun.tools.debug.RemoteClass
Return the method, specified by name.

getMethodName(RemoteStackFrame). Method in class sun.tools.debug.RemoteStackFrame
Get the method name referenced by this stackframe.

getMethodNames(RemoteClass). Method in class sun.tools.debug.RemoteClass
Return the names of all methods supported by this class.

getMethods(RemoteClass). Method in class sun.tools.debug.RemoteClass
Return the class's methods.

getMinimum(Scrollbar). Method in class java.awt.Scrollbar
Returns the minimum value of this Scrollbar.

GetMinSize(Container, GridBagLayoutInfo). Method in class java.awt. GridBagLayout

getMinutes(Date). Method in class java.util.Date
Returns the minute.

getMode(FileDialog). Method in class java.awt.FileDialog
Gets the mode of the file dialog.

getModifiers(RemoteField). Method in class sun.tools.debug.RemoteField
Returns a string with the field's modifiers, such as "public", "static", "final", etc.

getMonth(Date). Method in class java.util.Date
Returns the month.

getName(Class). Method in class java.lang.Class
Returns the name of this Class.

getName(File). Method in class java.io.File
Gets the name of the file.

getName(Font). Method in class java.awt.Font
Gets the logical name of the font.

getName(RemoteClass). Method in class sun.tools.debug.RemoteClass
Returns the name of the class.

getName(RemoteField). Method in class sun.tools.debug.RemoteField
Returns the name of the field.

getName(RemoteStackVariable). Method in class sun.tools.debug.RemoteStackVariable
Return the name of a stack variable or argument.

getName(RemoteThread). Method in class sun.tools.debug.RemoteThread
Return the name of the thread.

getName(RemoteThreadGroup). Method in class sun.tools.debug.RemoteThreadGroup
Return the threadgroup's name.

getName(Thread). Method in class java.lang.Thread
Gets and returns this Thread's name.

getName(ThreadGroup). Method in class java.lang.ThreadGroup
Gets the name of this Thread group.

getOrientation(Scrollbar). Method in class java.awt.Scrollbar
Returns the orientation for this Scrollbar.

getOutputStream(Process). Method in class java.lang.Process
Returns a Stream connected to the input of the child process.

getOutputStream(Socket). Method in class java.net.Socket
Gets an OutputStream for this socket.

getOutputStream(SocketImpl). Method in class java.net.SocketImpl
Gets an OutputStream for this socket.

getOutputStream(URLConnection). Method in class java.net.URLConnection
Calls this routine to get an OutputStream that writes to the object.

getPageIncrement(Scrollbar). Method in class java.awt.Scrollbar
Gets the page increment for this scrollbar.

getParameter(String). Method in class java.applet.Applet
Gets a parameter of the applet.

getParameter(String). Method in interface java.applet.AppletStub
Gets a parameter of the applet.

getParameterInfo(Applet). Method in class java.applet.Applet

Returns an array of strings describing the parameters that are understood by this applet.

getParent(). Method in class java.awt.Component

Gets the parent of the component.

getParent(). Method in class java.io.File

Gets the name of the parent directory.

getParent(). Method in class java.awt.MenuComponent

Returns the parent container.

getParent(). Method in class java.lang.ThreadGroup

Gets the parent of this Thread group.

getPath(). Method in class java.io.File

Gets the path of the file.

getPC(). Method in class sun.tools.debug.RemoteStackFrame

Get the program counter referenced by this stackframe.

getPeer(). Method in class java.awt.Component

Gets the peer of the component.

getPeer(). Method in class java.awt.MenuComponent

Gets the MenuComponent's peer.

getPixelSize(). Method in class java.awt.image.ColorModel

Returns the number of bits per pixel described by this ColorModel.

getPort(). Method in class java.net.DatagramPacket

getPort(). Method in class java.net.Socket

Gets the remote port to which the socket is connected.

getPort(). Method in class java.net.SocketImpl

getPort(). Method in class java.net.URL

Gets the port number.

getPriority(). Method in class java.lang.Thread

Gets and returns the Thread's priority.

getProperties(). Static method in class java.lang.System

Gets the System properties.

getProperty(String). Method in class java.util.Properties

Gets a property with the specified key.

getProperty(String). Static method in class java.lang.System

Gets the System property indicated by the specified key.

getProperty(String, ImageObserver). Method in class java.awt.Image

Gets a property of the image by name.

getProperty(String, String). Method in class java.util.Properties

Gets a property with the specified key and default.

getProperty(String, String). Static method in class java.lang.System

Gets the System property indicated by the specified key and def.

getProtocol(). Method in class java.net.URL

Gets the protocol name.

getRed(). Method in class java.awt.Color

Gets the red component.

getRed(int). Method in class java.awt.image.ColorModel

The subclass must provide a function which provides the red color component for the specified pixel.

getRed(int). Method in class java.awt.image.DirectColorModel

Returns the red color component for the specified pixel in the range 0-255.

getRed(int). Method in class java.awt.image.IndexColorModel

Returns the red color component for the specified pixel in the range 0-255.

getRedMask(). Method in class java.awt.image.DirectColorModel

Returns the mask indicating which bits in a pixel contain the red color component.

getReds(byte[]). Method in class java.awt.image.IndexColorModel

Copies the array of red color components into the given array.

getRef(). Method in class java.net.URL

Gets the ref.

getRemoteClass(). Method in class sun.tools.debug.RemoteStackFrame

Get the class this stackframe references.

getRequestProperty(String). Method in class java.net.URLConnection

getRGB(). Method in class java.awt.Color

Gets the RGB value representing the color in the default RGB ColorModel.

getRGB(int). Method in class java.awt.image.ColorModel

Returns the color of the pixel in the default RGB color model.

getRGB(int). Method in class java.awt.image.DirectColorModel

Returns the color of the pixel in the default RGB color model.

getRGB(int). Method in class java.awt.image.IndexColorModel

Returns the color of the pixel in the default RGB color model.

getRGBdefault(). Static method in class java.awt.image.ColorModel

Return a ColorModel which describes the default format for integer RGB values used throughout the AWT image interfaces.

getRows(). Method in class java.awt.List

Returns the number of visible lines in this list.

getRows(). Method in class java.awt.TextArea

Returns the number of rows in the TextArea.

getRuntime(). Static method in class java.lang.Runtime

Returns the runtime.

getScreenResolution(). Method in class java.awt.Toolkit

Returns the screen resolution in dots-per-inch.

getScreenSize(). Method in class java.awt.Toolkit

Gets the size of the screen.

getSeconds(). Method in class java.util.Date

Returns the second.

getSecurityContext(). Method in class java.lang.SecurityManager

Returns an implementation-dependent Object which encapsulates enough information about the current execution environment to perform some of the security checks later.

getSecurityManager(). Static method in class java.lang.System

Gets the system security interface.

getSelectedIndex(). Method in class java.awt.Choice

Returns the index of the currently selected item.

getSelectedIndex(). Method in class java.awt.List

Get the selected item on the list or -1 if no item is selected.

getSelectedIndexes(). Method in class java.awt.List

Returns the selected indexes on the list.

getSelectedIndexes(). Method in interface java.awt.peer.ListPeer

getSelectedItem(). Method in class java.awt.Choice

Returns a String representation of the current choice.

getSelectedItem(). Method in class java.awt.List

Returns the selected item on the list or null if no item is selected.

getSelectedItems(). Method in class java.awt.List

Returns the selected items on the list.

getSelectedText(). Method in class java.awt.TextComponent

Returns the selected text contained in this TextComponent.

getSelectionEnd(). Method in class java.awt.TextComponent

Returns the selected text's end position.

getSelectionEnd(). Method in interface java.awt.peer.TextComponentPeer

getSelectionStart(). Method in class java.awt.TextComponent

Returns the selected text's start position.

getSelectionStart(). Method in interface java.awt.peer.TextComponentPeer

getSize(). Method in class java.awt.Font

Gets the point size of the font.

getSize(). Method in class sun.tools.debug.RemoteArray

Return the number of elements in the array.

getSource(). Method in class java.awt.Image

Gets the object that produces the pixels for the image.

getSourceFile(). Method in class sun.tools.debug.RemoteClass
Get the source file referenced by this stackframe.

getSourceFileName(). Method in class sun.tools.debug.RemoteClass
Get the name of the source file referenced by this stackframe.

getSourcePath(). Method in class sun.tools.debug.RemoteDebugger
Return the source file path the Agent is currently using.

getStackVariable()(String). Method in class sun.tools.debug.RemoteThread
Return a stack variable from the current stackframe.

getStackVariables(). Method in class sun.tools.debug.RemoteThread
Return the arguments and local variable from the current stackframe.

getState(). Method in class java.awt.Checkbox
Returns the boolean state of the Checkbox.

getState(). Method in class java.awt.CheckboxMenuItem
Returns the state of this MenuItem.

getStaticFields(). Method in class sun.tools.debug.RemoteClass
Return all the static fields for this class.

getStatus(). Method in class sun.tools.debug.RemoteThread
Return the thread status description

getStyle(). Method in class java.awt.Font
Gets the style of the font.

getSuperclass(). Method in class java.lang.Class
Returns the superclass of this Class.

getSuperclass(). Method in class sun.tools.debug.RemoteClass
Return the superclass for this class.

getText(). Method in class java.awt.Label
Gets the text of this label.

getText(). Method in class java.awt.TextComponent
Returns the text contained in this TextComponent.

getText(). Method in interface java.awt.peer.TextComponentPeer

getThreadGroup(). Method in class java.lang.Thread
Gets and returns this Thread group.

getTime(). Method in class java.util.Date
Returns the time in milliseconds since the epoch.

getTimezoneOffset(). Method in class java.util.Date
Return the time zone offset in minutes for the current locale that is appropriate for this time.

getTitle(). Method in class java.awt.Dialog
Gets the title of the Dialog.

getTitle(). Method in class java.awt.Frame
Gets the title of the Frame.

getToolkit(). Method in class java.awt.Component
Gets the toolkit of the component.

getToolkit(). Method in interface java.awt.peer.ComponentPeer

getToolkit(). Method in class java.awt.Window
Returns the toolkit of this frame.

getTransparentPixel(). Method in class java.awt.image.IndexColorModel
Returns the index of the transparent pixel in this IndexColorModel or -1 if there is no transparent pixel.

getType(). Method in class sun.tools.debug.RemoteField
Returns a type string describing the field.

getType(). Method in class sun.tools.debug.RemoteValue
Returns the RemoteValue's type.

getURL(). Method in class java.net.URLConnection
Gets the URL for this connection.

getUseCaches(). Method in class java.net.URLConnection

getValue(). Method in class sun.tools.debug. RemoteStackVariable

Return the value of a stack variable or argument.

getValue(). Method in class java.awt.Scrollbar
Returns the current value of this Scrollbar.

getVisible(). Method in class java.awt.Scrollbar
Returns the visible amount of the Scrollbar.

getVisibleIndex(). Method in class java.awt.List
Gets the index of the item that was last made visible by the method makeVisible.

getWarningString(). Method in class java.awt.Window
Gets the warning string for this window.

getWidth(ImageObserver). Method in class java.awt.Image
Gets the actual width of the image.

getWidths(). Method in class java.awt.FontMetrics
Gets the widths of the first 256 characters in the Font.

getYear(). Method in class java.util.Date
Returns the year after 1900.

GOT_FOCUS. Static variable in class java.awt.Event
A component gained the focus.

gotFocus(Event, Object). Method in class java.awt.Component
Indicates that this component has received the input focus.

grabPixels(). Method in class java.awt.image.PixelGrabber
Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered.

grabPixels(long). Method in class java.awt.image.PixelGrabber
Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered or until the specified timeout has elapsed.

Graphics(). Constructor for class java.awt.Graphics
Constructs a new Graphics Object.

gray. Static variable in class java.awt.Color
The color gray.

green. Static variable in class java.awt.Color
The color green.

GridBagConstraints(). Constructor for class java.awt.GridBagConstraints

GridBagLayout(). Constructor for class java.awt.GridBagLayout
Creates a gridbag layout.

gridheight. Variable in class java.awt.GridBagConstraints

GridLayout(int, int). Constructor for class java.awt.GridLayout
Creates a grid layout with the specified rows and columns.

GridLayout(int, int, int, int). Constructor for class java.awt.GridLayout
Creates a grid layout with the specified rows, columns, horizontal gap, and vertical gap.

gridwidth. Variable in class java.awt.GridBagConstraints

gridx. Variable in class java.awt.GridBagConstraints

gridy. Variable in class java.awt.GridBagConstraints

grow(int, int). Method in class java.awt.Rectangle
Grows the rectangle horizontally and vertically.

guessContentTypeFromName(String). Static method in class java.net.URLConnection
A useful utility routine that tries to guess the content-type of an object based upon its extension.

guessContentTypeFromStream(InputStream). Static method in class java.net.URLConnection
This method is used to check for files that have some type that can be determined by inspection.

H

HAND_CURSOR. Static variable in class java.awt.Frame

handleEvent(Event). Method in class java.awt.Component

Handles the event.

handleEvent(Event). Method in interface java.awt.peer.ComponentPeer

hasChanged(). Method in class java.util.Observable

Returns a true boolean if an observable change has occurred.

hashCode(). Method in class java.util.BitSet

Gets the hashCode.

hashCode(). Method in class java.lang.Boolean

Returns a hashCode for this Boolean.

hashCode(). Method in class java.lang.Character

Returns a hashCode for this Character.

hashCode(). Method in class java.awt.Color

Computes the hash code.

hashCode(). Method in class java.util.Date

Computes a hashCode.

hashCode(). Method in class java.lang.Double

Returns a hashCode for this Double.

hashCode(). Method in class java.io.File

Computes a hashCode for the file.

hashCode(). Method in class java.lang.Float

Returns a hashCode for this Float.

hashCode(). Method in class java.awt.Font

Returns a hashCode for this font.

hashCode(). Method in class java.net.InetAddress

Returns a hashCode for this InetAddress.

hashCode(). Method in class java.lang.Integer

Returns a hashCode for this Integer.

hashCode(). Method in class java.lang.Long

Computes a hashCode for this Long.

hashCode(). Method in class java.lang.Object

Returns a hashCode for this Object.

hashCode(). Method in class java.awt.Point

Returns the hashCode for this Point.

hashCode(). Method in class java.awt.Rectangle

Returns the hashCode for this Rectangle.

hashCode(). Method in class java.lang.String

Returns a hashCode for this String.

hashCode(). Method in class java.net.URL

Creates an integer suitable for hash table indexing.

Hashtable(). Constructor for class java.util.Hashtable

Constructs a new, empty hashtable.

Hashtable(int). Constructor for class java.util.Hashtable

Constructs a new, empty hashtable with the specified initial capacity.

Hashtable(int, float). Constructor for class java.util.Hashtable

Constructs a new, empty hashtable with the specified initial capacity and the specified load factor.

hasMoreElements(). Method in interface java.util.Enumeration

Returns true if the enumeration contains more elements; false if its empty.

hasMoreElements(). Method in class java.util.StringTokenizer

Returns true if the Enumeration has more elements.

hasMoreTokens(). Method in class java.util.StringTokenizer

Returns true if more tokens exist.

height. Variable in class java.awt.Dimension

The height dimension.

HEIGHT. Static variable in interface java.awt.image.ImageObserver

The height of the base image is now available and can be taken from the height argument to the imageUpdate callback method.

height. Variable in class java.awt.Rectangle

The height of the rectangle.

hide(). Method in class java.awt.Component
Hides the component.

hide(). Method in interface java.awt.peer.ComponentPeer

HOME. Static variable in class java.awt.Event
The home key.

HORIZONTAL. Static variable in class java.awt.GridBagConstraints

HORIZONTAL. Static variable in class java.awt.Scrollbar
The horizontal Scrollbar variable.

HSBtoRGB(float, float, float). Static method in class java.awt.Color

Returns the RGB value defined by the default RGB ColorModel, of the color corresponding to the given HSB color components.

I

id. Variable in class java.awt.Event
The type of this event.

IEEEremainder(double, double). Static method in class java.lang.Math
Returns the remainder of f1 divided by f2 as defined by IEEE 754.

ifModifiedSince. Variable in class java.net.URLConnection

ignoreExceptions(). Method in class sun.tools.debug. RemoteClass
Don't enter the debugger when an instance of this class is thrown.

IllegalAccessError(). Constructor for class java.lang.IllegalAccessError
Constructs an IllegalAccessError with no detail message.

IllegalAccessError(String). Constructor for class java.lang.IllegalAccessError
Constructs an IllegalAccessError with the specified detail message.

IllegalAccessException(). Constructor for class java.lang. IllegalAccessException
Constructs a IllegalAccessException without a detail message.

IllegalAccessException(String). Constructor for class java.lang. IllegalAccessException
Constructs a IllegalAccessException with a detail message.

IllegalArgumentExcepion(). Constructor for class java.lang. IllegalArgumentExcepion
Constructs an IllegalArgumentExcepion with no detail message.

IllegalArgumentExcepion(String). Constructor for class java.lang. IllegalArgumentExcepion
Constructs an IllegalArgumentExcepion with the specified detail message.

IllegalMonitorStateException(). Constructor for class java.lang. IllegalMonitorStateException
Constructs an IllegalMonitorStateException with no detail message.

IllegalMonitorStateException(String). Constructor for class java.lang. IllegalMonitorStateException
Constructs an IllegalMonitorStateException with the specified detail message.

IllegalThreadStateException(). Constructor for class java.lang. IllegalThreadStateException
Constructs an IllegalThreadStateException with no detail message.

IllegalThreadStateException(String). Constructor for class java.lang. IllegalThreadStateException
Constructs an IllegalThreadStateException with the specified detail message.

Image(). Constructor for class java.awt.Image

IMAGEABORTED. Static variable in interface java.awt.image.ImageConsumer
The image creation process was deliberately aborted.

imageComplete(int). Method in interface java.awt.image.ImageConsumer

The imageComplete method is called when the ImageProducer is finished delivering all of the pixels that the source image contains, or when a single frame of a multi-frame animation has been completed, or when an error in loading or producing the image has occurred.

imageComplete(int). Method in class java.awt.image.ImageFilter

Filters the information provided in the imageComplete method of the ImageConsumer interface.

imageComplete(int). Method in class java.awt.image.PixelGrabber

The imageComplete method is part of the ImageConsumer API which this class must implement to

retrieve the pixels.

IMAGEERROR. Static variable in interface java.awt.image.ImageConsumer

An error was encountered while producing the image.

ImageFilter(). Constructor for class java.awt.image.ImageFilter

imageUpdate(Image, int, int, int, int, int). Method in class java.awt.Component

Repaints the component when the image has changed.

imageUpdate(Image, int, int, int, int, int). Method in interface java.awt.image. ImageObserver

This method is called when information about an image which was previously requested using an asynchronous interface becomes available.

in. Static variable in class java.io.FileDescriptor

Handle to standard input.

in. Variable in class java.io.FilterInputStream

The actual input stream.

in. Static variable in class java.lang.System

Standard input stream.

inCheck. Variable in class java.lang.SecurityManager

inClass(String). Method in class java.lang. SecurityManager

Returns true if the specified String is in this Class.

inClassLoader(()). Method in class java.lang.SecurityManager

Returns a boolean indicating whether or not the current ClassLoader is equal to null.

IncompatibleClassChangeError(()). Constructor for class java.lang. IncompatibleClassChangeError

Constructs an IncompatibleClassChangeError with no detail message.

IncompatibleClassChangeError(String). Constructor for class java.lang. IncompatibleClassChangeError

Constructs an IncompatibleClassChangeError with the specified detail message.

IndexColorModel(int, int, byte[], byte[], byte[]). Constructor for class java.awt.image.IndexColorModel

Constructs an IndexColorModel from the given arrays of red, green, and blue components.

IndexColorModel(int, int, byte[], byte[], byte[], byte[]). Constructor for class

java.awt.image.IndexColorModel

Constructs an IndexColorModel from the given arrays of red, green, blue and alpha components.

IndexColorModel(int, int, byte[], byte[], byte[], int). Constructor for class java.awt.image.IndexColorModel

Constructs an IndexColorModel from the given arrays of red, green, and blue components.

IndexColorModel(int, int, byte[], int, boolean). Constructor for class java.awt.image.IndexColorModel

Constructs an IndexColorModel from a single arrays of packed red, green, blue and optional alpha components.

IndexColorModel(int, int, byte[], int, boolean, int). Constructor for class java.awt.image.IndexColorModel

Constructs an IndexColorModel from a single arrays of packed red, green, blue and optional alpha components.

indexOf(int). Method in class java.lang.String

Returns the index within this String of the first occurrence of the specified character.

indexOf(int, int). Method in class java.lang.String

Returns the index within this String of the first occurrence of the specified character, starting the search at fromIndex.

indexOf(Object). Method in class java.util.Vector

Searches for the specified object, starting from the first position and returns an index to it.

indexOf(Object, int). Method in class java.util.Vector

Searches for the specified object, starting at the specified position and returns an index to it.

indexOf(String). Method in class java.lang.String

Returns the index within this String of the first occurrence of the specified substring.

indexOf(String, int). Method in class java.lang.String

Returns the index within this String of the first occurrence of the specified substring.

IndexOutOfBoundsException(()). Constructor for class java.lang. IndexOutOfBoundsException

Constructs an IndexOutOfBoundsException with no detail message.

IndexOutOfBoundsException(String). Constructor for class java.lang. IndexOutOfBoundsException

Constructs a IndexOutOfBoundsException with the specified detail message.

init(()). Method in class java.applet.Applet

Initializes the applet.

InputStream(). Constructor for class java.io.InputStream

inScope(). Method in class sun.tools.debug.RemoteStackVariable
Return whether variable is in scope.

insert(int, boolean). Method in class java.lang.StringBuffer
Inserts a boolean into the String buffer.

insert(int, char). Method in class java.lang.StringBuffer
Inserts a character into the String buffer.

insert(int, char[]). Method in class java.lang.StringBuffer
Inserts an array of characters into the String buffer.

insert(int, double). Method in class java.lang.StringBuffer
Inserts a double into the String buffer.

insert(int, float). Method in class java.lang.StringBuffer
Inserts a float into the String buffer.

insert(int, int). Method in class java.lang.StringBuffer
Inserts an integer into the String buffer.

insert(int, long). Method in class java.lang.StringBuffer
Inserts a long into the String buffer.

insert(int, Object). Method in class java.lang.StringBuffer
Inserts an object into the String buffer.

insert(int, String). Method in class java.lang.StringBuffer
Inserts a String into the String buffer.

insertElementAt(Object, int). Method in class java.util.Vector
Inserts the specified object as an element at the specified index.

insertText(String, int). Method in class java.awt.TextArea
Inserts the specified text at the specified position.

insertText(String, int). Method in interface java.awt.peer.TextAreaPeer

insets. Variable in class java.awt.GridBagConstraints

insets(). Method in class java.awt.Container
Returns the insets of the container.

insets(). Method in interface java.awt.peer.ContainerPeer

Insets(int, int, int, int). Constructor for class java.awt.Insets
Constructs and initializes a new Inset with the specified top, left, bottom, and right insets.

inside(int, int). Method in class java.awt.Component
Checks whether a specified x,y location is "inside" this Component.

inside(int, int). Method in class java.awt.Polygon
Determines whether the point (x,y) is inside the Polygon.

inside(int, int). Method in class java.awt.Rectangle
Checks if the specified point lies inside a rectangle.

InstantiationException(). Constructor for class java.lang.InstantiationException
Constructs an InstantiationException with no detail message.

InstantiationException(String). Constructor for class java.lang.InstantiationException
Constructs an InstantiationException with the specified detail message.

InstantiationExceptionException(). Constructor for class java.lang.InstantiationExceptionException
Constructs an InstantiationExceptionException with no detail message.

InstantiationExceptionException(String). Constructor for class java.lang.InstantiationExceptionException
Constructs an InstantiationExceptionException with the specified detail message.

intBitsToFloat(int). Static method in class java.lang.Float
Returns the single-float corresponding to a given bit representation.

Integer(int). Constructor for class java.lang.Integer
Constructs an Integer object initialized to the specified int value.

Integer(String). Constructor for class java.lang.Integer
Constructs an Integer object initialized to the value specified by the String parameter.

intern(). Method in class java.lang.String
Returns a String that is equal to this String but which is guaranteed to be from the unique String pool.

InternalError(). Constructor for class java.lang.InternalError
Constructs an InternalError with no detail message.

InternalError(String). Constructor for class java.lang.InternalError

Constructs an InternalError with the specified detail message.

interrupt(). Method in class java.lang.Thread

Send an interrupt to a thread.

interrupted(). Static method in class java.lang.Thread

Ask if you have been interrupted.

InterruptedException(String). Constructor for class java.lang.InterruptedException

Constructs an InterruptedException with no detail message.

InterruptedException(String). Constructor for class java.lang.InterruptedException

Constructs an InterruptedException with the specified detail message.

IOException(String). Constructor for class java.io.IOException

Constructs an IOException with no detail message.

IOException(String). Constructor for class java.io.IOException

Constructs an IOException with the specified detail message.

intersection(Rectangle). Method in class java.awt.Rectangle

Computes the intersection of two rectangles.

intersects(Rectangle). Method in class java.awt.Rectangle

Checks if two rectangles intersect.

intValue(Double). Method in class java.lang.Double

Returns the integer value of this Double (by casting to an int).

intValue(Float). Method in class java.lang.Float

Returns the integer value of this Float (by casting to an int).

intValue(Integer). Method in class java.lang.Integer

Returns the value of this Integer as an int.

intValue(Long). Method in class java.lang.Long

Returns the value of this Long as an int.

intValue(Number). Method in class java.lang.Number

Returns the value of the number as an int.

invalidate(Component). Method in class java.awt.Component

Invalidates a component.

IOException(String). Constructor for class java.io.IOException

Constructs an IOException with no detail message.

IOException(String). Constructor for class java.io.IOException

Constructs an IOException with the specified detail message.

ipadx. Variable in class java.awt.GridBagConstraints

ipady. Variable in class java.awt.GridBagConstraints

isAbsolute(File). Method in class java.io.File

Returns a boolean indicating whether the file name is absolute.

isActive(Applet). Method in class java.applet.Applet

Returns true if the applet is active.

isActive(AppletStub). Method in interface java.applet.AppletStub

Returns true if the applet is active.

isAlive(Thread). Method in class java.lang.Thread

Returns a boolean indicating if the Thread is active.

isBold(Font). Method in class java.awt.Font

Returns true if the font is bold.

isConsumer(ImageConsumer). Method in class java.awt.image.FilteredImageSource

Determines whether an ImageConsumer is on the list of consumers currently interested in data for this image.

isConsumer(ImageConsumer). Method in interface java.awt.image.ImageProducer

This method determines if a given ImageConsumer object is currently registered with this ImageProducer as one of its consumers.

isConsumer(ImageConsumer). Method in class java.awt.image.MemoryImageSource

Determine if an ImageConsumer is on the list of consumers currently interested in data for this image.

isDaemon(Thread). Method in class java.lang.Thread

Returns the daemon flag of the Thread.

isDaemon(). Method in class java.lang.ThreadGroup

Returns the daemon flag of the Thread group.

isDigit(char). Static method in class java.lang.Character

Determines if the specified character is a ISO-LATIN-1 digit.

isDirectory(). Method in class java.io.File

Returns a boolean indicating whether or not a directory file exists.

isEditable(). Method in class java.awt.TextComponent

Returns the boolean indicating whether this TextComponent is editable or not.

isEmpty(). Method in class java.util.Dictionary

Returns true if the Dictionary contains no elements.

isEmpty(). Method in class java.util.Hashtable

Returns true if the hashtable contains no elements.

isEmpty(). Method in class java.awt.Rectangle

Determines whether the rectangle is empty.

isEmpty(). Method in class java.util.Vector

Returns true if the collection contains no values.

isEnabled(). Method in class java.awt.Component

Checks if this Component is enabled.

isEnabled(). Method in class java.awt.MenuItem

Checks whether the menu item is enabled.

isErrorAny(). Method in class java.awt.MediaTracker

Checks the error status of all of the images.

isErrorID(int). Method in class java.awt.MediaTracker

Checks the error status of all of the images with the specified ID.

isFile(). Method in class java.io.File

Returns a boolean indicating whether or not a normal file exists.

isInfinite(). Method in class java.lang.Double

Returns true if this Double value is infinitely large in magnitude.

isInfinite(). Method in class java.lang.Float

Returns true if this Float value is infinitely large in magnitude.

isInfinite(double). Static method in class java.lang.Double

Returns true if the specified number is infinitely large in magnitude.

isInfinite(float). Static method in class java.lang.Float

Returns true if the specified number is infinitely large in magnitude.

isInterface(). Method in class java.lang.Class

Returns a boolean indicating whether or not this Class is an interface.

isInterface(). Method in class sun.tools.debug.RemoteClass

Is this RemoteClass an interface?

isInterrupted(). Method in class java.lang.Thread

Ask if another thread has been interrupted.

isItalic(). Method in class java.awt.Font

Returns true if the font is italic.

isLowerCase(char). Static method in class java.lang.Character

Determines if the specified character is ISO-LATIN-1 lower case.

isModal(). Method in class java.awt.Dialog

Returns true if the Dialog is modal.

isNaN(). Method in class java.lang.Double

Returns true if this Double value is the special Not-a-Number (NaN) value.

isNaN(). Method in class java.lang.Float

Returns true if this Float value is Not-a-Number (NaN).

isNaN(double). Static method in class java.lang.Double

Returns true if the specified number is the special Not-a-Number (NaN) value.

isNaN(float). Static method in class java.lang.Float

Returns true if the specified number is the special Not-a-Number (NaN) value.

isObject(). Method in class sun.tools.debug.RemoteValue

Returns whether the RemoteValue is an Object (as opposed to a primitive type, such as int).

isPlain(). Method in class java.awt.Font
Returns true if the font is plain.

isResizable(). Method in class java.awt.Dialog
Returns true if the user can resize the frame.

isResizable(). Method in class java.awt.Frame
Returns true if the user can resize the Frame.

isSelected()(int). Method in class java.awt.List
Returns true if the item at the specified index has been selected; false otherwise.

isShowing(). Method in class java.awt.Component
Checks if this Component is showing on screen.

isSpace()(char). Static method in class java.lang.Character
Determines if the specified character is ISO-LATIN-1 white space according to Java.

isStatic(). Method in class sun.tools.debug.RemoteField
Returns whether the field is static (a class variable or method).

isSuspended(). Method in class sun.tools.debug.RemoteThread
Return whether this thread is suspended.

isTearOff(). Method in class java.awt.Menu
Returns true if this is a tear-off menu.

isUpperCase()(char). Static method in class java.lang.Character
Determines if the specified character is ISO-LATIN-1 upper case.

isValid(). Method in class java.awt.Component
Checks if this Component is valid.

isVisible(). Method in class java.awt.Component
Checks if this Component is visible.

ITALIC. Static variable in class java.awt.Font
The italicized style constant.

itrace()(boolean). Method in class sun.tools.debug.RemoteDebugger
Turn on/off instruction tracing.

J

join(). Method in class java.lang.Thread
Waits forever for this Thread to die.

join()(long). Method in class java.lang.Thread
Waits for this Thread to die.

join()(long, int). Method in class java.lang.Thread
Waits for the Thread to die, with more precise time.

K

key. Variable in class java.awt.Event
The key that was pressed in a keyboard event.

KEY_ACTION. Static variable in class java.awt.Event
The key action keyboard event.

KEY_ACTION_RELEASE. Static variable in class java.awt.Event
The key action keyboard event.

KEY_PRESS. Static variable in class java.awt.Event
The key press keyboard event.

KEY_RELEASE. Static variable in class java.awt.Event

The key release keyboard event.

keyDown(Event, int). Method in class java.awt.Component

Called if a character is pressed.

keys(). Method in class java.util.Dictionary

Returns an enumeration of the Dictionary's keys.

keys(). Method in class java.util.Hashtable

Returns an enumeration of the hashtable's keys.

keyUp(Event, int). Method in class java.awt.Component

Called if a character is released.

L

Label() Constructor for class java.awt.Label

Constructs an empty label.

Label(String). Constructor for class java.awt.Label

Constructs a new label with the specified String of text.

Label(String, int). Constructor for class java.awt.Label

Constructs a new label with the specified String of text and the specified alignment.

last(Container). Method in class java.awt.CardLayout

Flips to the last card of the specified container.

lastElement(). Method in class java.util.Vector

Returns the last element of the sequence.

lastIndexOf(int). Method in class java.lang.String

Returns the index within this String of the last occurrence of the specified character.

lastIndexOf(int, int). Method in class java.lang.String

Returns the index within this String of the last occurrence of the specified character.

lastIndexOf(Object). Method in class java.util.Vector

Searches backwards for the specified object, starting from the last position and returns an index to it.

lastIndexOf(Object, int). Method in class java.util.Vector

Searches backwards for the specified object, starting from the specified position and returns an index to it.

lastIndexOf(String). Method in class java.lang.String

Returns the index within this String of the last occurrence of the specified substring.

lastIndexOf(String, int). Method in class java.lang.String

Returns the index within this String of the last occurrence of the specified substring.

lastModified(). Method in class java.io.File

Returns the last modification time.

layout(). Method in class java.awt.Component

Lays out the component.

layout(). Method in class java.awt.Container

Does a layout on this Container.

layoutContainer(Container). Method in class java.awt.BorderLayout

Lays out the specified container.

layoutContainer(Container). Method in class java.awt.CardLayout

Performs a layout in the specified panel.

layoutContainer(Container). Method in class java.awt.FlowLayout

Lays out the container.

layoutContainer(Container). Method in class java.awt.GridBagLayout

Lays out the container in the specified panel.

layoutContainer(Container). Method in class java.awt.GridLayout

Lays out the container in the specified panel.

layoutContainer(Container). Method in interface java.awt.LayoutManager

Lays out the container in the specified panel.

layoutInfo. Variable in class java.awt.GridBagLayout

LEFT. Static variable in class java.awt.Event

The left arrow key.

LEFT. Static variable in class java.awt.FlowLayout

The left alignment variable.

left. Variable in class java.awt.Insets

The inset from the left.

LEFT. Static variable in class java.awt.Label

The left alignment.

length(). Method in class java.io.File

Returns the length of the file.

length(). Method in class java.io.RandomAccessFile

Returns the length of the file.

length(). Method in class java.lang.String

Returns the length of the String.

length(). Method in class java.lang.StringBuffer

Returns the length (character count) of the buffer.

lightGray. Static variable in class java.awt.Color

The color light gray.

lineno(). Method in class java.io.StreamTokenizer

Return the current line number.

LineNumberInputStream(InputStream). Constructor for class java.io. LineNumberInputStream

Constructs a new LineNumberInputStream initialized with the specified input stream.

LinkageError(). Constructor for class java.lang.LinkageError

Constructs a LinkageError with no detail message.

LinkageError(String). Constructor for class java.lang.LinkageError

Constructs a LinkageError with the specified detail message.

list(). Method in class java.awt.Component

Prints a listing to a print stream.

list(). Method in class java.io.File

Lists the files in a directory.

List(int). Constructor for class java.awt.List

Creates a new scrolling list initialized with no visible Lines or multiple selections.

list(). Method in class java.lang.ThreadGroup

Lists this Thread group.

list(FilenameFilter). Method in class java.io.File

Uses the specified filter to list files in a directory.

List(int, boolean). Constructor for class java.awt.List

Creates a new scrolling list initialized with the specified number of visible lines and a boolean stating whether multiple selections are allowed or not.

list(PrintStream). Method in class java.awt.Component

Prints a listing to the specified print out stream.

list(PrintStream). Method in class java.util.Properties

List properties, for debugging

list(PrintStream, int). Method in class java.awt.Component

Prints out a list, starting at the specified indentation, to the specified print stream.

list(PrintStream, int). Method in class java.awt.Container

Prints out a list, starting at the specified indentation, to the specified out stream.

LIST_DESELECT. Static variable in class java.awt.Event

LIST_SELECT. Static variable in class java.awt.Event

listBreakpoints(). Method in class sun.tools.debug. RemoteDebugger

Return a list of the breakpoints which are currently set.

listClasses(). Method in class sun.tools.debug. RemoteDebugger

List the currently known classes.

listen(int). Method in class java.net.SocketImpl

Listens for connections over a specified amount of time.

listThreadGroups(RemoteThreadGroup). Method in class sun.tools.debug. RemoteDebugger

List threadgroups

listThreads(boolean). Method in class sun.tools.debug. RemoteThreadGroup

List a threadgroup's threads

LOAD. Static variable in class java.awt. FileDialog

The file load variable.

load(InputStream). Method in class java.util. Properties

Loads properties from an InputStream.

load(String). Method in class java.lang. Runtime

Loads a dynamic library, given a complete path name.

load(String). Static method in class java.lang. System

Loads a dynamic library, given a complete path name.

LOAD_FILE. Static variable in class java.awt. Event

A file loading event.

loadClass(String, boolean). Method in class java.lang. ClassLoader

Resolves the specified name to a Class.

LOADING. Static variable in class java.awt. MediaTracker

Flag indicating some media is currently being loaded.

loadLibrary(String). Method in class java.lang. Runtime

Loads a dynamic library with the specified library name.

loadLibrary(String). Static method in class java.lang. System

Loads a dynamic library with the specified library name.

localport. Variable in class java.net. SocketImpl

locate(int, int). Method in class java.awt. Component

Returns the component or subcomponent that contains the x,y location.

locate(int, int). Method in class java.awt. Container

Locates the component that contains the x,y position.

location(()). Method in class java.awt. Component

Returns the current location of this component.

location(int, int). Method in class java.awt. GridBagLayout

log(double). Static method in class java.lang. Math

Returns the natural logarithm (base e) of a.

Long(long). Constructor for class java.lang. Long

Constructs a Long object initialized to the specified value.

Long(String). Constructor for class java.lang. Long

Constructs a Long object initialized to the value specified by the String parameter.

longBitsToDouble(long). Static method in class java.lang. Double

Returns the double-float corresponding to a given bit representation.

longValue(()). Method in class java.lang. Double

Returns the long value of this Double (by casting to a long).

longValue(()). Method in class java.lang. Float

Returns the long value of this Float (by casting to a long).

longValue(()). Method in class java.lang. Integer

Returns the value of this Integer as a long.

longValue(()). Method in class java.lang. Long

Returns the value of this Long as a long.

longValue(()). Method in class java.lang. Number

Returns the value of the number as a long.

lookupConstraints(Component). Method in class java.awt. GridBagLayout

Retrieves the constraints for the specified component.

loop(()). Method in interface java.applet. AudioClip

Starts playing the clip in a loop.

LOST_FOCUS. Static variable in class java.awt. Event

A component lost the focus.

lostFocus(Event, Object). Method in class java.awt. Component

Indicates that this component has lost the input focus.

lowerCaseMode(boolean). Method in class java.io.StreamTokenizer

Examines a boolean to decide whether TT_WORD tokens are forced to be lower case.

M

magenta. Static variable in class java.awt.Color

The color magenta.

makeVisible(int). Method in class java.awt.List

Forces the item at the specified index to be visible.

makeVisible(int). Method in interface java.awt.peer.ListPeer

MalformedURLException(()). Constructor for class java.net.MalformedURLException

Constructs a MalformedURLException with no detail message.

MalformedURLException(String). Constructor for class java.net. MalformedURLException

Constructs a MalformedURLException with the specified detail message.

mark(int). Method in class java.io.BufferedInputStream

Marks the current position in the input stream.

mark(int). Method in class java.io.FilterInputStream

Marks the current position in the input stream.

mark(int). Method in class java.io.InputStream

Marks the current position in the input stream.

mark(int). Method in class java.io.LineNumberInputStream

Marks the current position in the input stream.

marklimit. Variable in class java.io.BufferedInputStream

The maximum readahead allowed after a mark() before subsequent calls to reset() fail.

markpos. Variable in class java.io.BufferedInputStream

The position in the buffer of the current mark.

markSupported(()). Method in class java.io.BufferedInputStream

Returns a boolean indicating if this stream type supports mark/reset.

markSupported(()). Method in class java.io.FilterInputStream

Returns true if this stream type supports mark/reset.

markSupported(()). Method in class java.io.InputStream

Returns a boolean indicating whether or not this stream type supports mark/reset.

markSupported(()). Method in class java.io.PushbackInputStream

Returns true if this stream type supports mark/reset.

max(double, double). Static method in class java.lang.Math

Takes two double values, a and b, and returns the greater number of the two.

max(float, float). Static method in class java.lang.Math

Takes two float values, a and b, and returns the greater number of the two.

max(int, int). Static method in class java.lang.Math

Takes two int values, a and b, and returns the greater number of the two.

max(long, long). Static method in class java.lang.Math

Takes two long values, a and b, and returns the greater number of the two.

MAX_PRIORITY. Static variable in class java.lang.Thread

The maximum priority that a Thread can have.

MAX_RADIX. Static variable in class java.lang.Character

The maximum radix available for conversion to and from Strings.

MAX_VALUE. Static variable in class java.lang.Boolean

The maximum value a Character can have.

MAX_VALUE. Static variable in class java.lang.Double

The maximum value a double can have.

MAX_VALUE. Static variable in class java.lang.Float

The maximum value a float can have.

MAX_VALUE. Static variable in class java.lang.Integer

The maximum value an Integer can have.

MAX_VALUE. Static variable in class java.lang.Long

The maximum value a Long can have.

MAXGRIDSZIE. Static variable in class java.awt.GridBagLayout

MediaTracker(Component). Constructor for class java.awt.MediaTracker

Creates a Media tracker to track images for a given Component.

MemoryImageSource(int, int, ColorModel, byte[], int, int). Constructor for class

java.awt.image.MemoryImageSource

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

MemoryImageSource(int, int, ColorModel, byte[], int, int, Hashtable). Constructor for class

java.awt.image.MemoryImageSource

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

MemoryImageSource(int, int, ColorModel, int[], int, int). Constructor for class

java.awt.image.MemoryImageSource

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

MemoryImageSource(int, int, ColorModel, int[], int, int, Hashtable). Constructor for class

java.awt.image.MemoryImageSource

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

MemoryImageSource(int, int, int[], int, int). Constructor for class java.awt.image.MemoryImageSource

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

MemoryImageSource(int, int, int[], int, int, Hashtable). Constructor for class

java.awt.image.MemoryImageSource

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

Menu(String). Constructor for class java.awt.Menu

Constructs a new Menu with the specified label.

Menu(String, boolean). Constructor for class java.awt.Menu

Constructs a new Menu with the specified label.

MenuBar(()). Constructor for class java.awt.MenuBar

Creates a new menu bar.

MenuComponent(()). Constructor for class java.awt.MenuComponent

MenuItem(String). Constructor for class java.awt.MenuItem

Constructs a new MenuItem with the specified label.

META_MASK. Static variable in class java.awt.Event

The meta modifier constant.

metaDown(()). Method in class java.awt.Event

Checks if the meta key is down.

min(double, double). Static method in class java.lang.Math

Takes two double values, a and b, and returns the smallest number of the two.

min(float, float). Static method in class java.lang.Math

Takes two float values, a and b, and returns the smallest number of the two.

min(int, int). Static method in class java.lang.Math

Takes two integer values, a and b, and returns the smallest number of the two.

min(long, long). Static method in class java.lang.Math

Takes two long values, a and b, and returns the smallest number of the two.

MIN_PRIORITY. Static variable in class java.lang.Thread

The minimum priority that a Thread can have.

MIN_RADIX. Static variable in class java.lang.Character

The minimum radix available for conversion to and from Strings.

MIN_VALUE. Static variable in class java.lang.Boolean

The minimum value a Charater can have.

MIN_VALUE. Static variable in class java.lang.Double

The minimum value a double can have.

MIN_VALUE. Static variable in class java.lang.Float

The minimum value a float can have.

MIN_VALUE. Static variable in class java.lang.Integer

The minimum value an Integer can have.

MIN_VALUE. Static variable in class java.lang.Long

The minimum value a Long can have.

minimumLayoutSize(Container). Method in class java.awt.BorderLayout

Returns the minimum dimensions needed to layout the components contained in the specified target container.

minimumLayoutSize(Container). Method in class java.awt.CardLayout

Calculates the minimum size for the specified panel.

minimumLayoutSize(Container). Method in class java.awt.FlowLayout

Returns the minimum dimensions needed to layout the components contained in the specified target container.

minimumLayoutSize(Container). Method in class java.awt.GridBagLayout

Returns the minimum dimensions needed to layout the components contained in the specified panel.

minimumLayoutSize(Container). Method in class java.awt.GridLayout

Returns the minimum dimensions needed to layout the components contained in the specified panel.

minimumLayoutSize(Container). Method in interface java.awt.LayoutManager

Calculates the minimum size dimensions for the specified panel given the components in the specified parent container.

minimumSize(Component). Method in class java.awt.Component

Returns the minimum size of this component.

minimumSize(ComponentPeer). Method in interface java.awt.peer.ComponentPeer

minimumSize(Container). Method in class java.awt.Container

Returns the minimum size of this container.

minimumSize(List). Method in class java.awt.List

Returns the minimum dimensions needed for the list.

minimumSize(TextArea). Method in class java.awt.TextArea

Returns the minimum size Dimensions of the TextArea.

minimumSize(TextField). Method in class java.awt.TextField

Returns the minimum size Dimensions needed for this TextField.

minimumSize(int). Method in class java.awt.List

Returns the minimum dimensions needed for the amount of rows in the list.

minimumSize(int). Method in interface java.awt.peer.ListPeer

minimumSize(int). Method in class java.awt.TextField

Returns the minimum size Dimensions needed for this TextField with the specified amount of columns.

minimumSize(int). Method in interface java.awt.peer.TextFieldPeer

minimumSize(int, int). Method in class java.awt.TextArea

Returns the specified minimum size Dimensions of the TextArea.

minimumSize(int, int). Method in interface java.awt.peer.TextAreaPeer

MINSIZE. Static variable in class java.awt.GridBagLayout

mkdir(File). Method in class java.io.File

Creates a directory and returns a boolean indicating the success of the creation.

makedirs(File). Method in class java.io.File

Creates all directories in this path.

modifiers. Variable in class java.awt.Event

The state of the modifier keys.

MOUSE_DOWN. Static variable in class java.awt.Event

The mouse down event.

MOUSE_DRAG. Static variable in class java.awt.Event

The mouse drag event.

MOUSE_ENTER. Static variable in class java.awt.Event

The mouse enter event.

MOUSE_EXIT. Static variable in class java.awt.Event

The mouse exit event.

MOUSE_MOVE. Static variable in class java.awt.Event

The mouse move event.

MOUSE_UP. Static variable in class java.awt.Event

The mouse up event.

mouseDown(Event, int, int). Method in class java.awt.Component

Called if the mouse is down.

mouseDrag(Event, int, int). Method in class java.awt.Component

Called if the mouse is dragged (the mouse button is down).

mouseEnter(Event, int, int). Method in class java.awt.Component

Called when the mouse enters the component.

mouseExit(Event, int, int). Method in class java.awt.Component

Called when the mouse exits the component.

mouseMove(Event, int, int). Method in class java.awt.Component

Called if the mouse moves (the mouse button is up).

mouseUp(Event, int, int). Method in class java.awt.Component

Called if the mouse is up.

move(int, int). Method in class java.awt.Component

Moves the Component to a new location.

move(int, int). Method in class java.awt.Point

Moves the point.

move(int, int). Method in class java.awt.Rectangle

Moves the rectangle.

MOVE_CURSOR. Static variable in class java.awt.Frame

N

N_RESIZE_CURSOR. Static variable in class java.awt.Frame

name. Variable in class java.awt.Font

The logical name of this font.

NaN. Static variable in class java.lang.Double

Not-a-Number.

NaN. Static variable in class java.lang.Float

Not-a-Number.

NE_RESIZE_CURSOR. Static variable in class java.awt.Frame

NEGATIVE_INFINITY. Static variable in class java.lang.Double

Negative infinity.

NEGATIVE_INFINITY. Static variable in class java.lang.Float

Negative infinity.

NegativeArraySizeException(String). Constructor for class java.lang.NegativeArraySizeException

Constructs a NegativeArraySizeException with no detail message.

NegativeArraySizeException(String). Constructor for class java.lang.NegativeArraySizeException

Constructs a NegativeArraySizeException with the specified detail message.

newInstance(Class). Method in class java.lang.Class

Creates a new instance of this Class.

newmodel. Variable in class java.awt.image.RGBImageFilter

next(Thread). Method in class sun.tools.debug.RemoteThread

Continue execution of this thread to the next line, but don't step into a method call.

next(Container). Method in class java.awt.CardLayout

Flips to the next card of the specified container.

nextDouble(). Method in class java.util.Random

Generates a pseudorandom uniformly distributed double value between 0.0 and 1.0.

nextElement(). Method in interface java.util.Enumeration

Returns the next element of the enumeration.

nextElement(). Method in class java.util.StringTokenizer

Returns the next element in the Enumeration.

nextFloat(). Method in class java.util.Random

Generates a pseudorandom uniformly distributed float value between 0.0 and 1.0.

nextFocus(). Method in class java.awt.Component

Moves the focus to the next component.

nextFocus(). Method in interface java.awt.peer.ComponentPeer

nextGaussian(). Method in class java.util.Random

Generates a pseudorandom Gaussian distributed double value with mean 0.0 and standard deviation 1.0.

nextInt(). Method in class java.util.Random

Generates a pseudorandom uniformly distributed int value.

nextLong(). Method in class java.util.Random

Generate a pseudorandom uniformly distributed long value.

nextToken(). Method in class java.io.StreamTokenizer

Parses a token from the input stream.

nextToken(). Method in class java.util.StringTokenizer

Returns the next token of the String.

nextToken(String). Method in class java.util.StringTokenizer

Returns the next token, after switching to the new delimiter set.

NoClassDefFoundError(). Constructor for class java.lang.NoClassDefFoundError

Constructs a NoClassDefFoundError with no detail message.

NoClassDefFoundError(String). Constructor for class java.lang.NoClassDefFoundError

Constructs a NoClassDefFoundError with the specified detail message.

NONE. Static variable in class java.awt.GridBagConstraints

NORM_PRIORITY. Static variable in class java.lang.Thread

The default priority that is assigned to a Thread.

NORTH. Static variable in class java.awt.GridBagConstraints

NORTHEAST. Static variable in class java.awt.GridBagConstraints

NORTHWEST. Static variable in class java.awt.GridBagConstraints

NoSuchElementException(). Constructor for class java.util.NoSuchElementException

Constructs a NoSuchElementException with no detail message.

NoSuchElementException(String). Constructor for class java.util.NoSuchElementException

Constructs a NoSuchElementException with the specified detail message.

NoSuchFieldError(). Constructor for class java.lang.NoSuchFieldError

Constructs a NoSuchFieldException without a detail message.

NoSuchFieldError(String). Constructor for class java.lang.NoSuchFieldError

Constructs a NoSuchFieldException with a detail message.

NoSuchMethodError(). Constructor for class java.lang.NoSuchMethodError

NoSuchMethodError(String). Constructor for class java.lang.NoSuchMethodError

Constructs a NoSuchMethodException with a detail message.

NoSuchMethodException(). Constructor for class java.lang.NoSuchMethodException

Constructs a NoSuchMethodException without a detail message.

NoSuchMethodException(String). Constructor for class java.lang.NoSuchMethodException

Constructs a NoSuchMethodException with a detail message.

notify(). Method in class java.lang.Object

Notifies a single waiting thread on a change in condition of another thread.

notifyAll(). Method in class java.lang.Object

Notifies all of the threads waiting for a condition to change.

notifyObservers(). Method in class java.util.Observable

Notifies all observers if an observable change occurs.

notifyObservers(Object). Method in class java.util.Observable

Notifies all observers of the specified observable change which occurred.

npoints. Variable in class java.awt.Polygon

The total number of points.

NullPointerException(String). Constructor for class java.lang.NullPointerException

Constructs a NullPointerException with no detail message.

NullPointerException(String). Constructor for class java.lang. NullPointerException

Constructs a NullPointerException with the specified detail message.

Number(String). Constructor for class java.lang.Number

NumberFormatException(String). Constructor for class java.lang.NumberFormatException

Constructs a NumberFormatException with no detail message.

NumberFormatException(String). Constructor for class java.lang. NumberFormatException

Constructs a NumberFormatException with the specified detail message.

nval. Variable in class java.io.StreamTokenizer

The number value.

NW_RESIZE_CURSOR. Static variable in class java.awt.Frame

O

Object(String). Constructor for class java.lang.Object

Observable(String). Constructor for class java.util. Observable

openConnection(String). Method in class java.net.URL

Creates (if not already in existence) a URLConnection object that contains a connection to the remote object referred to by the URL.

openConnection(String). Method in class java.net.URLConnection

Opens an input stream to the object referenced by the URL.

openStream(String). Method in class java.net.URL

Opens an input stream.

or(BitSet). Method in class java.util.BitSet

Logically ORs this bit set with the specified set of bits.

orange. Static variable in class java.awt.Color

The color orange.

ordinaryChar(int). Method in class java.io.StreamTokenizer

Specifies that this character is 'ordinary': it removes any significance as a word, comment, string, whitespace or number character.

ordinaryChars(int, int). Method in class java.io.StreamTokenizer

Specifies that characters in this range are 'ordinary'.

origmodel. Variable in class java.awt.image.RGBImageFilter

out. Static variable in class java.io.FileDescriptor

Handle to standard output.

out. Variable in class java.io.FilterOutputStream

The actual output stream.

out. Static variable in class java.lang.System

Standard output stream.

OutOfMemoryError(String). Constructor for class java.lang.OutOfMemoryError

Constructs an OutOfMemoryError with no detail message.

OutOfMemoryError(String). Constructor for class java.lang.OutOfMemoryError

Constructs an OutOfMemoryError with the specified detail message.

OutputStream(String). Constructor for class java.io.OutputStream

P

pack(). Method in class java.awt.Window

Packs the components of the Window.

paint(Graphics). Method in class java.awt.Canvas

Paints the canvas in the default background color.

paint(Graphics). Method in class java.awt.Component

Paints the component.

paint(Graphics). Method in interface java.awt.peer.ComponentPeer

paintAll(Graphics). Method in class java.awt.Component

Paints the component and its subcomponents.

paintComponents(Graphics). Method in class java.awt.Container

Paints the components in this container.

Panel(Panel). Constructor for class java.awt.Panel

Creates a new panel.

paramString(Button). Method in class java.awt.Button

Returns the parameter String of this button.

paramString(Checkbox). Method in class java.awt.Checkbox

Returns the parameter String of this Checkbox.

paramString(CheckboxMenuItem). Method in class java.awt.CheckboxMenuItem

Returns the parameter String of this button.

paramString(Choice). Method in class java.awt.Choice

Returns the parameter String of this Choice.

paramString(Component). Method in class java.awt.Component

Returns the parameter String of this Component.

paramString(Container). Method in class java.awt.Container

Returns the parameter String of this Container.

paramString(Dialog). Method in class java.awt.Dialog

Returns the parameter String of this Dialog.

paramString(Event). Method in class java.awt.Event

Returns the parameter String of this Event.

paramString(FileDialog). Method in class java.awt.FileDialog

Returns the parameter String of this file dialog.

paramString(Frame). Method in class java.awt.Frame

Returns the parameter String of this Frame.

paramString(Label). Method in class java.awt.Label

Returns the parameter String of this label.

paramString(List). Method in class java.awt.List

Returns the parameter String of this list.

paramString(MenuComponent). Method in class java.awt.MenuComponent

Returns the String parameter of this MenuComponent.

paramString(MenuItem). Method in class java.awt.MenuItem

Returns the String parameter of the menu item.

paramString(Scrollbar). Method in class java.awt.Scrollbar

Returns the String parameters for this Scrollbar.

paramString(TextArea). Method in class java.awt.TextArea

Returns the String of parameters for this TextArea.

paramString(TextComponent). Method in class java.awt.TextComponent

Returns the String of parameters for this TextComponent.

paramString(TextField). Method in class java.awt.TextField

Returns the String of parameters for this TExtField.

parentOf(ThreadGroup). Method in class java.lang.ThreadGroup

Checks to see if this Thread group is a parent of or is equal to another Thread group.

parse(String). Static method in class java.util.Date

Given a string representing a time, parse it and return the time value.

parseInt(String). Static method in class java.lang.Integer

Assuming the specified String represents an integer, returns that integer's value.

parseInt(String, int). Static method in class java.lang.Integer

Assuming the specified String represents an integer, returns that integer's value.

parseLong(String). Static method in class java.lang.Long

Assuming the specified String represents a long, return that long's value.

parseLong(String, int). Static method in class java.lang.Long

Assuming the specified String represents a long, returns that long's value.

parseNumbers(int). Method in class java.io.StreamTokenizer

Specifies that numbers should be parsed.

parseURL(URL, String, int, int). Method in class java.net.URLStreamHandler

This method is called to parse the string spec into URL u.

pathSeparator. Static variable in class java.io.File

The system dependent path separator string.

pathSeparatorChar. Static variable in class java.io.File

The system dependent path separator character.

peek(int). Method in class java.util.Stack

Peeks at the top of the stack.

PGDN. Static variable in class java.awt.Event

The page down key.

PGUP. Static variable in class java.awt.Event

The page up key.

Pi. Static variable in class java.lang.Math

The float representation of the value Pi.

pink. Static variable in class java.awt.Color

The color pink.

PipedInputStream(int). Constructor for class java.io.PipedInputStream

Creates an input file that isn't connected to anything (yet).

PipedInputStream(PipedOutputStream). Constructor for class java.io. PipedInputStream

Creates an input file from the specified PipedOutputStream.

PipedOutputStream(int). Constructor for class java.io.PipedOutputStream

Creates an output file that isn't connected to anything (yet).

PipedOutputStream(PipedInputStream). Constructor for class java.io. PipedOutputStream

Creates an output file connected to the specified PipedInputStream.

pixel_bits. Variable in class java.awt.image.ColorModel

PixelGrabber(Image, int, int, int, int, int[], int, int). Constructor for class java.awt.image.PixelGrabber

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the specified image into the given array.

PixelGrabber(ImageProducer, int, int, int, int, int[], int, int). Constructor for class java.awt.image.PixelGrabber

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the image produced by the specified ImageProducer into the given array.

PLAIN. Static variable in class java.awt.Font

The plain style constant.

play(int). Method in interface java.applet.AudioClip

Starts playing the clip.

play(URL). Method in class java.applet.Applet

Plays an audio clip.

play(URL, String). Method in class java.applet.Applet

Plays an audio clip.

Point(int, int). Constructor for class java.awt.Point

Constructs and initializes a Point from the specified x and y coordinates.

Polygon(int). Constructor for class java.awt.Polygon

Creates an empty polygon.

Polygon(int[], int[], int). Constructor for class java.awt.Polygon

Constructs and initializes a Polygon from the specified parameters.

pop(). Method in class java.util.Stack
Pops an item off the stack.

port. Variable in class java.net.SocketImpl
The port where the socket will make a connection.

pos. Variable in class java.io.BufferedInputStream
The current position in the buffer.

pos. Variable in class java.io.ByteArrayInputStream
The current position in the buffer.

pos. Variable in class java.io.StringBufferInputStream
The position in the buffer.

POSITIVE_INFINITY. Static variable in class java.lang.Double
Positive infinity.

POSITIVE_INFINITY. Static variable in class java.lang.Float
Positive infinity.

postEvent(Event). Method in class java.awt.Component
Posts an event to this component.

postEvent(Event). Method in class java.awt.MenuComponent
Posts the specified event to the menu.

postEvent(Event). Method in interface java.awt.MenuContainer

pow(double, double). Static method in class java.lang.Math
Returns the number a raised to the power of b.

preferredLayoutSize(Container). Method in class java.awt.BorderLayout
Returns the preferred dimensions for this layout given the components in the specified target container.

preferredLayoutSize(Container). Method in class java.awt.CardLayout
Calculates the preferred size for the specified panel.

preferredLayoutSize(Container). Method in class java.awt.FlowLayout
Returns the preferred dimensions for this layout given the components in the specified target container.

preferredLayoutSize(Container). Method in class java.awt.GridBagLayout
Returns the preferred dimensions for this layout given the components in the specified panel.

preferredLayoutSize(Container). Method in class java.awt.GridLayout
Returns the preferred dimensions for this layout given the components in the specified panel.

preferredLayoutSize(Container). Method in interface java.awt.LayoutManager
Calculates the preferred size dimensions for the specified panel given the components in the specified parent container.

PREFERRED_SIZE. Static variable in class java.awt.GridBagLayout

preferredSize(). Method in class java.awt.Component
Returns the preferred size of this component.

preferredSize(). Method in interface java.awt.peer.ComponentPeer

preferredSize(). Method in class java.awt.Container
Returns the preferred size of this container.

preferredSize(). Method in class java.awt.List
Returns the preferred dimensions needed for the list.

preferredSize(). Method in class java.awt.TextArea
Returns the preferred size Dimensions of the TextArea.

preferredSize(). Method in class java.awt.TextField
Returns the preferred size Dimensions needed for this TextField.

preferredSize(int). Method in class java.awt.List
Returns the preferred dimensions needed for the list with the specified amount of rows.

preferredSize(int). Method in interface java.awt.peer.ListPeer

preferredSize(int). Method in class java.awt.TextField
Returns the preferred size Dimensions needed for this TextField with the specified amount of columns.

preferredSize(int). Method in interface java.awt.peer.TextFieldPeer

preferredSize(int, int). Method in class java.awt.TextArea

Returns the specified row and column Dimensions of the TextArea.

preferredSize(int, int). Method in interface java.awt.peer.TextAreaPeer

prepareImage(Image, ImageObserver). Method in class java.awt.Component

Prepares an image for rendering on this Component.

prepareImage(Image, int, int, ImageObserver). Method in class java.awt. Component

Prepares an image for rendering on this Component at the specified width and height.

prepareImage(Image, int, int, ImageObserver). Method in interface java.awt.peer.ComponentPeer

prepareImage(Image, int, int, ImageObserver). Method in class java.awt.Toolkit

Prepares an image for rendering on the default screen at the specified width and height.

previous(Container). Method in class java.awt.CardLayout

Flips to the previous card of the specified container.

print(boolean). Method in class java.io.PrintStream

Prints a boolean.

print(char). Method in class java.io.PrintStream

Prints an character.

print(char[]). Method in class java.io.PrintStream

Prints an array of characters.

print(double). Method in class java.io.PrintStream

Prints a double.

print(float). Method in class java.io.PrintStream

Prints a float.

print(Graphics). Method in class java.awt.Component

Prints this component.

print(Graphics). Method in interface java.awt.peer.ComponentPeer

print(int). Method in class java.io.PrintStream

Prints an integer.

print(long). Method in class java.io.PrintStream

Prints a long.

print(Object). Method in class java.io.PrintStream

Prints an object.

print(String). Method in class java.io.PrintStream

Prints a String.

printAll(Graphics). Method in class java.awt.Component

Prints the component and its subcomponents.

printComponents(Graphics). Method in class java.awt.Container

Prints the components in this container.

println(). Method in class java.io.PrintStream

Prints a newline.

println(boolean). Method in class java.io.PrintStream

Prints a boolean followed by a newline.

println(char). Method in class java.io.PrintStream

Prints a character followed by a newline.

println(char[]). Method in class java.io.PrintStream

Prints an array of characters followed by a newline.

println(double). Method in class java.io.PrintStream

Prints a double followed by a newline.

println(float). Method in class java.io.PrintStream

Prints a float followed by a newline.

println(int). Method in class java.io.PrintStream

Prints an integer followed by a newline.

println(long). Method in class java.io.PrintStream

Prints a long followed by a newline.

println(Object). Method in class java.io.PrintStream

Prints an object followed by a newline.

println(String). Method in class java.io.PrintStream

Prints a string followed by a newline.

printStackTrace(). Method in class java.lang.Throwable
Prints the Throwable and the Throwable's stack trace.

printStackTrace(PrintStream). Method in class java.lang.Throwable

PrintStream(OutputStream). Constructor for class java.io.PrintStream
Creates a new PrintStream.

PrintStream(OutputStream, boolean). Constructor for class java.io.PrintStream
Creates a new PrintStream, with auto flushing.

printToConsole(String). Method in interface sun.tools.debug.DebuggerCallback
Print text to the debugger's console window.

Process(Properties). Constructor for class java.lang.Process

PROPERTIES. Static variable in interface java.awt.image.ImageObserver
The properties of the image are now available.

Properties(Properties). Constructor for class java.util.Properties
Creates an empty property list.

Properties(Properties). Constructor for class java.util.Properties
Creates an empty property list with specified defaults.

propertyNames(Properties). Method in class java.util.Properties
Enumerates all the keys.

ProtocolException(String). Constructor for class java.net.ProtocolException
Constructs a new ProtocolException with no detail message.

ProtocolException(String). Constructor for class java.net.ProtocolException
Constructs a new ProtocolException with the specified detail message.

push(Object). Method in class java.util.Stack
Pushes an item onto the stack.

pushBack. Variable in class java.io.PushbackInputStream
Push back character.

pushBack(InputStream). Method in class java.io.StreamTokenizer
Pushes back a stream token.

PushbackInputStream(InputStream). Constructor for class java.io. PushbackInputStream
Creates a PushbackInputStream.

put(Object, Object). Method in class java.util.Dictionary
Puts the specified element into the Dictionary, using the specified key.

put(Object, Object). Method in class java.util.Hashtable
Puts the specified element into the hashtable, using the specified key.

Q

quitEvent(DebuggerCallback).Method in interface sun.tools.debug.DebuggerCallback
The client interpreter has exited, either by returning from its main thread, or by calling System.exit().

quoteChar(int). Method in class java.io.StreamTokenizer
Specifies that matching pairs of this character delimit String constants.

R

random(Math).Static method in class java.lang.Math
Generates a random number between 0.0 and 1.0.

Random(Random). Constructor for class java.util.Random
Creates a new random number generator.

Random(long). Constructor for class java.util.Random
Creates a new random number generator using a single long seed.

RandomAccessFile(File, String). Constructor for class java.io.RandomAccessFile

Creates a RandomAccessFile from a specified File object and mode ("r" or "rw").

RandomAccessFile(String, String). Constructor for class java.io. RandomAccessFile

Creates a RandomAccessFile with the specified system dependent file name and the specified mode.

RANDOMPIXELORDER. Static variable in interface java.awt.image.ImageConsumer

The pixels will be delivered in a random order.

read(). Method in class java.io.BufferedInputStream

Reads a byte of data.

read(). Method in class java.io.ByteArrayInputStream

Reads a byte of data.

read(). Method in class java.io.FileInputStream

Reads a byte of data.

read(). Method in class java.io.FilterInputStream

Reads a byte.

read(). Method in class java.io.InputStream

Reads a byte of data.

read(). Method in class java.io.LineNumberInputStream

Reads a byte of data.

read(). Method in class java.io.PipedInputStream

Reads a byte of data.

read(). Method in class java.io.PushbackInputStream

Reads a byte of data.

read(). Method in class java.io.RandomAccessFile

Reads a byte of data.

read(). Method in class java.io.SequenceInputStream

Reads a stream, and upon reaching an EOF, flips to the next stream.

read(). Method in class java.io.StringBufferInputStream

Reads a byte of data.

read(byte[]). Method in class java.io.DataInputStream

Reads data into an array of bytes.

read(byte[]). Method in class java.io.FileInputStream

Reads data into an array of bytes.

read(byte[]). Method in class java.io.FilterInputStream

Reads into an array of bytes.

read(byte[]). Method in class java.io.InputStream

Reads into an array of bytes.

read(byte[]). Method in class java.io.RandomAccessFile

Reads data into an array of bytes.

read(byte[], int, int). Method in class java.io.BufferedInputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.ByteArrayInputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.DataInputStream

Reads data into an array of bytes.

read(byte[], int, int). Method in class java.io.FileInputStream

Reads data into an array of bytes.

read(byte[], int, int). Method in class java.io.FilterInputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.InputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.LineNumberInputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.PipedInputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.PushbackInputStream

Reads into an array of bytes.

read(byte[], int, int). Method in class java.io.RandomAccessFile

Reads a sub array as a sequence of bytes.

read(byte[], int, int). Method in class java.io.SequenceInputStream

Reads data into an array of bytes, and upon reaching an EOF, flips to the next stream.

read(byte[], int, int). Method in class java.io.StringBufferInputStream

Reads into an array of bytes.

readBoolean(). Method in interface java.io.DataInput

Reads in a boolean.

readBoolean(). Method in class java.io.DataInputStream

Reads a boolean.

readBoolean(). Method in class java.io.RandomAccessFile

Reads a boolean.

readByte(). Method in interface java.io.DataInput

Reads an 8 bit byte.

readByte(). Method in class java.io.DataInputStream

Reads an 8 bit byte.

readByte(). Method in class java.io.RandomAccessFile

Reads a byte.

readChar(). Method in interface java.io.DataInput

Reads a 16 bit char.

readChar(). Method in class java.io.DataInputStream

Reads a 16 bit char.

readChar(). Method in class java.io.RandomAccessFile

Reads a 16 bit char.

readDouble(). Method in interface java.io.DataInput

Reads a 64 bit double.

readDouble(). Method in class java.io.DataInputStream

Reads a 64 bit double.

readDouble(). Method in class java.io.RandomAccessFile

Reads a 64 bit double.

readFloat(). Method in interface java.io.DataInput

Reads a 32 bit float.

readFloat(). Method in class java.io.DataInputStream

Reads a 32 bit float.

readFloat(). Method in class java.io.RandomAccessFile

Reads a 32 bit float.

readFully(byte[]). Method in interface java.io.DataInput

Reads bytes, blocking until all bytes are read.

readFully(byte[]). Method in class java.io.DataInputStream

Reads bytes, blocking until all bytes are read.

readFully(byte[]). Method in class java.io.RandomAccessFile

Reads bytes, blocking until all bytes are read.

readFully(byte[], int, int). Method in interface java.io.DataInput

Reads bytes, blocking until all bytes are read.

readFully(byte[], int, int). Method in class java.io.DataInputStream

Reads bytes, blocking until all bytes are read.

readFully(byte[], int, int). Method in class java.io.RandomAccessFile

Reads bytes, blocking until all bytes are read.

readInt(). Method in interface java.io.DataInput

Reads a 32 bit int.

readInt(). Method in class java.io.DataInputStream

Reads a 32 bit int.

readInt(). Method in class java.io.RandomAccessFile

Reads a 32 bit int.

readLine(). Method in interface java.io.DataInput

readLine(). Method in class java.io.DataInputStream
Reads in a line that has been terminated by a '\n', '\r', '\r\n' or EOF.

readLine(). Method in class java.io.RandomAccessFile
Reads a line terminated by a '\n' or EOF.

readLong(). Method in interface java.io.DataInput
Reads a 64 bit long.

readLong(). Method in class java.io.DataInputStream
Reads a 64 bit long.

readLong(). Method in class java.io.RandomAccessFile
Reads a 64 bit long.

readShort(). Method in interface java.io.DataInput
Reads a 16 bit short.

readShort(). Method in class java.io.DataInputStream
Reads a 16 bit short.

readShort(). Method in class java.io.RandomAccessFile
Reads 16 bit short.

readUnsignedByte(). Method in interface java.io.DataInput
Reads an unsigned 8 bit byte.

readUnsignedByte(). Method in class java.io.DataInputStream
Reads an unsigned 8 bit byte.

readUnsignedByte(). Method in class java.io.RandomAccessFile
Reads an unsigned 8 bit byte.

readUnsignedShort(). Method in interface java.io.DataInput
Reads an unsigned 16 bit short.

readUnsignedShort(). Method in class java.io.DataInputStream
Reads 16 bit short.

readUnsignedShort(). Method in class java.io.RandomAccessFile
Reads 16 bit short.

readUTF(). Method in interface java.io.DataInput

readUTF(). Method in class java.io.DataInputStream
Reads a UTF format String.

readUTF(). Method in class java.io.RandomAccessFile
Reads a UTF formatted String.

readUTF(DataInput). Static method in class java.io.DataInputStream
Reads a UTF format String from the given input stream.

receive(DatagramPacket). Method in class java.net.DatagramSocket
Receives datagram packet.

Rectangle(). Constructor for class java.awt.Rectangle
Constructs a new rectangle.

Rectangle(Dimension). Constructor for class java.awt.Rectangle
Constructs a rectangle and initializes it to the specified width and height.

Rectangle(int, int). Constructor for class java.awt.Rectangle
Constructs a rectangle and initializes it with the specified width and height parameters.

Rectangle(int, int, int, int). Constructor for class java.awt.Rectangle
Constructs and initializes a rectangle with the specified parameters.

Rectangle(Point). Constructor for class java.awt.Rectangle
Constructs a rectangle and initializes it to the specified point.

Rectangle(Point, Dimension). Constructor for class java.awt.Rectangle
Constructs a rectangle and initializes it to a specified point and dimension.

red. Static variable in class java.awt.Color
The color red.

regionMatches(boolean, int, String, int, int). Method in class java.lang. String
Determines whether a region of this String matches the specified region of the specified String.

regionMatches(int, String, int, int). Method in class java.lang.String
Determines whether a region of this String matches the specified region of the specified String.

rehash(). Method in class java.util.Hashtable

Rehashes the content of the table into a bigger table.

RELATIVE. Static variable in class java.awt.GridBagConstraints

REMAINDER. Static variable in class java.awt. GridBagConstraints

RemoteDebugger(String, DebuggerCallback, boolean). Constructor for class sun.tools.debug.RemoteDebugger

Create a remote debugger, and connect it to a new client interpreter.

RemoteDebugger (String, String, DebuggerCallback, boolean). Constructor for class sun.tools.debug.RemoteDebugger

Create a remote debugger, connecting it with a running Java interpreter.

To connect to a running interpreter, it must be started with the "-debug" option, whereupon it will print out the password for that debugging session.

RemoteInt(int). Constructor for class sun.tools.debug.RemoteInt

remove(Component). Method in class java.awt.Container

Removes the specified component from this container.

remove(int). Method in class java.awt.Menu

Deletes the item from this menu at the specified index.

remove(int). Method in class java.awt.MenuBar

Removes the menu located at the specified index from the menu bar.

remove(MenuComponent). Method in class java.awt.Frame

Removes the specified menu bar from this Frame.

remove(MenuComponent). Method in class java.awt.Menu

Deletes the specified item from this menu.

remove(MenuComponent). Method in class java.awt.MenuBar

Removes the specified menu from the menu bar.

remove(MenuComponent). Method in interface java.awt.MenuContainer

remove(Object). Method in class java.util.Dictionary

Removes the element corresponding to the key.

remove(Object). Method in class java.util.Hashtable

Removes the element corresponding to the key.

removeAll(). Method in class java.awt.Container

Removes all the components from this container.

removeAllElements(). Method in class java.util.Vector

Removes all elements of the vector.

removeConsumer(ImageConsumer). Method in class java.awt.image. FilteredImageSource

Removes an ImageConsumer from the list of consumers interested in data for this image.

removeConsumer(ImageConsumer). Method in interface java.awt.image.ImageProducer

This method removes the given ImageConsumer object from the list of consumers currently registered to receive image data.

removeConsumer(ImageConsumer). Method in class java.awt.image.MemoryImageSource

Remove an ImageConsumer from the list of consumers interested in data for this image.

removeElement(Object). Method in class java.util.Vector

Removes the element from the vector.

removeElementAt(int). Method in class java.util.Vector

Deletes the element at the specified index.

removeLayoutComponent(Component). Method in class java.awt.BorderLayout

Removes the specified component from the layout.

removeLayoutComponent(Component). Method in class java.awt.CardLayout

Removes the specified component from the layout.

removeLayoutComponent(Component). Method in class java.awt.FlowLayout

Removes the specified component from the layout.

removeLayoutComponent(Component). Method in class java.awt.GridBagLayout

Removes the specified component from the layout.

removeLayoutComponent(Component). Method in class java.awt.GridLayout

Removes the specified component from the layout.

removeLayoutComponent(Component). Method in interface java.awt.LayoutManager

Removes the specified component from the layout.

removeNotify(Component). Method in class java.awt.Component

Notifies the Component to destroy the peer.

removeNotify(Container). Method in class java.awt.Container

Notifies the container to remove its peer.

removeNotify(List). Method in class java.awt.List

Removes the peer for this list.

removeNotify(Menu). Method in class java.awt.Menu

Removes the menu's peer.

removeNotify(MenuBar). Method in class java.awt.MenuBar

Removes the menu bar's peer.

removeNotify(MenuComponent). Method in class java.awt.MenuComponent

Removes the menu component's peer.

removeNotify(TextComponent). Method in class java.awt.TextComponent

Removes the TextComponent's peer.

renameTo(File). Method in class java.io.File

Renames a file and returns a boolean indicating whether or not this method was successful.

repaint(Component). Method in class java.awt.Component

Repaints the component.

repaint(int, int, int, int). Method in class java.awt.Component

Repaints part of the component.

repaint(long). Method in class java.awt.Component

Repaints the component.

repaint(long, int, int, int, int). Method in class java.awt.Component

Repaints part of the component.

repaint(long, int, int, int, int). Method in interface java.awt.peer.ComponentPeer

replace(char, char). Method in class java.lang.String

Converts this String by replacing all occurrences of oldChar with newChar.

replaceItem(String, int). Method in class java.awt.List

Replaces the item at the given index.

replaceText(String, int, int). Method in class java.awt.TextArea

Replaces text from the indicated start to end position with the new text specified.

replaceText(String, int, int). Method in interface java.awt.peer.TextAreaPeer

requestFocus(Component). Method in class java.awt.Component

Requests the input focus.

requestFocus(ComponentPeer). Method in interface java.awt.peer.ComponentPeer

requestTopDownLeftRightResend(ImageConsumer). Method in class

java.awt.image.FilteredImageSource

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order.

requestTopDownLeftRightResend(ImageConsumer). Method in interface

java.awt.image.ImageProducer

This method is used by an ImageConsumer to request that the ImageProducer attempt to resend the image data one more time in TOPDOWNLEFTRIGHT order so that higher quality conversion algorithms which depend on receiving pixels in order can be used to produce a better output version of the image.

requestTopDownLeftRightResend(ImageConsumer). Method in class java.awt.image.

MemoryImageSource

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order.

resendTopDownLeftRight(ImageProducer). Method in class java.awt.image. ImageFilter

Responds to a request for a TopDownLeftRight (TDLR) ordered resend of the pixel data from an ImageConsumer.

reset(BufferedInputStream). Method in class java.io.BufferedInputStream

Repositions the stream to the last marked position.

reset(). Method in class java.io.ByteArrayInputStream

Resets the buffer to the beginning.

reset(). Method in class java.io.ByteArrayOutputStream

Resets the buffer so that you can use it again without throwing away the already allocated buffer.

reset(). Method in class java.io.FilterInputStream

Repositions the stream to the last marked position.

reset(). Method in class java.io.InputStream

Repositions the stream to the last marked position.

reset(). Method in class java.io.LineNumberInputStream

Repositions the stream to the last marked position.

reset(). Method in class java.io.StringBufferInputStream

Resets the buffer to the beginning.

resetCurrentFrameIndex(). Method in class sun.tools.debug.RemoteThread

Reset the current stackframe

resetSyntax(). Method in class java.io.StreamTokenizer

Resets the syntax table so that all characters are special.

reshape(int, int, int, int). Method in class java.awt.Component

Reshapes the Component to the specified bounding box.

reshape(int, int, int, int). Method in interface java.awt.peer.ComponentPeer

reshape(int, int, int, int). Method in class java.awt. Rectangle

Reshapes the rectangle.

resize(Dimension). Method in class java.applet.Applet

Requests that the applet be resized.

resize(Dimension). Method in class java.awt.Component

Resizes the Component to the specified dimension.

resize(int, int). Method in class java.applet.Applet

Requests that the applet be resized.

resize(int, int). Method in class java.awt.Component

Resizes the Component to the specified width and height.

resize(int, int). Method in class java.awt.Rectangle

Resizes the rectangle.

resolveClass(Class). Method in class java.lang.ClassLoader

Resolves classes referenced by this Class.

resume(). Method in class sun.tools.debug.RemoteThread

Resume execution of this thread.

resume(). Method in class java.lang.Thread

Resumes this Thread execution.

resume(). Method in class java.lang.ThreadGroup

Resumes all the Threads in this Thread group and all of its sub groups.

RGBImageFilter(). Constructor for class java.awt.image.RGBImageFilter

RGBtoHSB(int, int, int, float[]). Static method in class java.awt.Color

Returns the HSB values corresponding to the color defined by the red, green, and blue components.

RIGHT. Static variable in class java.awt.Event

The right arrow key.

RIGHT. Static variable in class java.awt.FlowLayout

The right alignment variable.

right. Variable in class java.awt.Insets

The inset from the right.

RIGHT. Static variable in class java.awt.Label

The right alignment.

rint(double). Static method in class java.lang.Math

Converts a double value into an integral value in double format.

round(double). Static method in class java.lang.Math

Rounds off a double value by first adding 0.5 to it and then returning the largest integer that is less than or equal to this new value.

round(float). Static method in class java.lang.Math

Rounds off a float value by first adding 0.5 to it and then returning the largest integer that is less than or equal to this new value.

rowHeights. Variable in class java.awt.GridBagLayout

rowWeights. Variable in class java.awt.GridBagLayout

run(). Method in interface java.lang.Runnable

The method that is executed when a Runnable object is activated.

run(). Method in class java.lang.Thread

The actual body of this Thread.

run(int, String[]). Method in class sun.tools.debug.RemoteDebugger

Load and run a runnable Java class, with any optional parameters.

runFinalization(). Method in class java.lang.Runtime

Runs the finalization methods of any objects pending finalization.

runFinalization(). Static method in class java.lang.System

Runs the finalization methods of any objects pending finalization.

RuntimeException(). Constructor for class java.lang.RuntimeException

Constructs a RuntimeException with no detail message.

RuntimeException(String). Constructor for class java.lang.RuntimeException

Constructs a RuntimeException with the specified detail message.

S

S_RESIZE_CURSOR. Static variable in class java.awt.Frame

sameFile(URL). Method in class java.net.URL

Compares two URLs, excluding the "ref" fields: sameFile is true if the true references the same remote object, but not necessarily the same subpiece of that object.

SAVE. Static variable in class java.awt.FileDialog

The file save variable.

save(OutputStream, String). Method in class java.util.Properties

Save properties to an OutputStream.

SAVE_FILE. Static variable in class java.awt.Event

A file saving event.

SCROLL_ABSOLUTE. Static variable in class java.awt.Event

The absolute scroll event.

SCROLL_LINE_DOWN. Static variable in class java.awt.Event

The line down scroll event.

SCROLL_LINE_UP. Static variable in class java.awt.Event

The line up scroll event.

SCROLL_PAGE_DOWN. Static variable in class java.awt.Event

The page down scroll event.

SCROLL_PAGE_UP. Static variable in class java.awt.Event

The page up scroll event.

Scrollbar(). Constructor for class java.awt.Scrollbar

Constructs a new vertical Scrollbar.

Scrollbar(int). Constructor for class java.awt.Scrollbar

Constructs a new Scrollbar with the specified orientation.

Scrollbar(int, int, int, int, int). Constructor for class java.awt.Scrollbar

Constructs a new Scrollbar with the specified orientation, value, page size, and minimum and maximum values.

SE_RESIZE_CURSOR. Static variable in class java.awt.Frame

search(Object). Method in class java.util.Stack

Sees if an object is on the stack.

SecurityException(). Constructor for class java.lang.SecurityException

Constructs a SecurityException with no detail message.

SecurityException(String). Constructor for class java.lang.SecurityException

Constructs a SecurityException with the specified detail message.

SecurityManager(). Constructor for class java.lang.SecurityManager

Constructs a new SecurityManager.

seek(long). Method in class java.io.RandomAccessFile

Sets the file pointer to the specified absolute position.

select(int). Method in class java.awt.Choice

Selects the item with the specified position.

select(int). Method in interface java.awt.peer.ChoicePeer

select(int). Method in class java.awt.List

Selects the item at the specified index.

select(int). Method in interface java.awt.peer.ListPeer

select(int, int). Method in class java.awt.TextComponent

Selects the text found between the specified start and end locations.

select(int, int). Method in interface java.awt.peer.TextComponentPeer

select(String). Method in class java.awt.Choice

Selects the item with the specified String.

selectAll(). Method in class java.awt.TextComponent

Selects all the text in the TextComponent.

send(DatagramPacket). Method in class java.net.DatagramSocket

Sends Datagram Packet to the destination address

separator. Static variable in class java.io.File

The system dependent file separator String.

separatorChar. Static variable in class java.io.File

The system dependent file separator character.

SequenceInputStream(Enumeration). Constructor for class java.io.SequenceInputStream

Constructs a new SequenceInputStream initialized to the specified list.

SequenceInputStream(InputStream, InputStream). Constructor for class java.io.SequenceInputStream

Constructs a new SequenceInputStream initialized to the two specified input streams.

ServerSocket(int). Constructor for class java.net.ServerSocket

Creates a server socket on a specified port.

ServerSocket(int, int). Constructor for class java.net.ServerSocket

Creates a server socket, binds it to the specified local port and listens to it.

set(int). Method in class java.util.BitSet

Sets a bit.

set(String, String, int, String, String). Method in class java.net.URL

Sets the fields of the URL.

setAlignment(int). Method in class java.awt.Label

Sets the alignment for this label to the specified alignment.

setAlignment(int). Method in interface java.awt.peer.LabelPeer

setAllowUserInteraction(boolean). Method in class java.net.URLConnection

Some URL connections occasionally need to to interactions with the user.

setBackground(Color). Method in class java.awt.Component

Sets the background color.

setBackground(Color). Method in interface java.awt.peer.ComponentPeer

setBreakpointLine(int). Method in class sun.tools.debug.RemoteClass

Set a breakpoint at a specified source line number in a class.

setBreakpointMethod(RemoteField). Method in class sun.tools.debug.RemoteClass

Set a breakpoint at the first line of a class method.

setChanged(). Method in class java.util.Observable

Sets a flag to note an observable change.

setCharAt(int, char). Method in class java.lang.StringBuffer

Changes the character at the specified index to be ch.

setCheckboxGroup(CheckboxGroup). Method in class java.awt.Checkbox

Sets the CheckboxGroup to the specified group.

setCheckboxGroup(CheckboxGroup). Method in interface java.awt.peer.CheckboxPeer

setColor(Color). Method in class java.awt.Graphics

Sets the current color to the specified color.

setColorModel(ColorModel). Method in interface java.awt.image.ImageConsumer

The ColorModel object used for the majority of the pixels reported using the setPixels method calls.

setColorModel(ColorModel). Method in class java.awt.image.ImageFilter

Filter the information provided in the setColorModel method of the ImageConsumer interface.

setColorModel(ColorModel). Method in class java.awt.image.PixelGrabber

The setColorModel method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

setColorModel(ColorModel). Method in class java.awt.image.RGBImageFilter

If the ColorModel is an IndexColorModel, and the subclass has set the canFilterIndexColorModel flag to true, we substitute a filtered version of the color model here and wherever that original ColorModel object appears in the setPixels methods.

setConstraints(Component, GridBagConstraints). Method in class java.awt.GridBagLayout

Sets the constraints for the specified component.

setContentHandlerFactory(ContentHandlerFactory). Static method in class java.net.URLConnection

Sets the ContentHandler factory.

setCurrent(Checkbox). Method in class java.awt.CheckboxGroup

Sets the current choice to the specified Checkbox.

setCurrentFrameIndex(int). Method in class sun.tools.debug.RemoteThread

Set the current stackframe index

setCursor(int). Method in class java.awt.Frame

Set the cursor image to a predefined cursor.

setCursor(int). Method in interface java.awt.peer.FramePeer

setDaemon(boolean). Method in class java.lang.Thread

Marks this Thread as a daemon Thread or a user Thread.

setDaemon(boolean). Method in class java.lang.ThreadGroup

Changes the daemon status of this group.

setDate(int). Method in class java.util.Date

Sets the date.

setDefaultAllowUserInteraction(boolean). Static method in class java.net.URLConnection

Sets/gets the default value of the allowUserInteraction flag.

setDefaultRequestProperty(String, String). Static method in class java.net.URLConnection

Sets/gets the default value of a general request property.

setDefaultUseCaches(boolean). Method in class java.net.URLConnection

setDimensions(int, int). Method in class java.awt.image.CropImageFilter

Override the source image's dimensions and pass the dimensions of the rectangular cropped region to the ImageConsumer.

setDimensions(int, int). Method in interface java.awt.image.ImageConsumer

The dimensions of the source image are reported using the setDimensions method call.

setDimensions(int, int). Method in class java.awt.image.ImageFilter

Filters the information provided in the setDimensions method of the ImageConsumer interface.

setDimensions(int, int). Method in class java.awt.image.PixelGrabber

The setDimensions method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

setDirectory(String). Method in class java.awt.FileDialog

Set the directory of the Dialog to the specified directory.

setDirectory(String). Method in interface java.awt.peer.FileDialogPeer

setDoInput(boolean). Method in class java.net.URLConnection

A URL connection can be used for input and/or output.

setDoOutput(boolean). Method in class java.net.URLConnection

A URL connection can be used for input and/or output.

setEchoCharacter(char). Method in class java.awt.TextField

Sets the echo character for this TextField.

setEchoCharacter(char). Method in interface java.awt.peer.TextFieldPeer

setEditable(boolean). Method in class java.awt.TextComponent

Sets the specified boolean to indicate whether or not this TextComponent should be editable.
setEditable(boolean). Method in interface java.awt.peer.TextComponentPeer

setElementAt(Object, int). Method in class java.util. Vector
Sets the element at the specified index to be the specified object.

setFile(String). Method in class java.awt.FileDialog
Sets the file for this dialog to the specified file.

setFile(String). Method in interface java.awt.peer.FileDialogPeer

setFilenameFilter(FilenameFilter). Method in class java.awt.FileDialog
Sets the filter for this dialog to the specified filter.

setFilenameFilter(FilenameFilter). Method in interface java.awt.peer. FileDialogPeer

setFont(Font). Method in class java.awt.Component
Sets the font of the component.

setFont(Font). Method in interface java.awt.peer.ComponentPeer

setFont(Font). Method in class java.awt.Graphics
Sets the font for all subsequent text-drawing operations.

setFont(Font). Method in class java.awt.MenuItem
Sets the font to be used for this MenuItem to the specified font.

setForeground(Color). Method in class java.awt.Component
Sets the foreground color.

setForeground(Color). Method in interface java.awt.peer.ComponentPeer

setHelpMenu(Menu). Method in class java.awt.MenuBar
Sets the help menu to the specified menu on the menu bar.

setHints(int). Method in interface java.awt.image.ImageConsumer
The ImageProducer can deliver the pixels in any order, but the ImageConsumer may be able to scale or convert the pixels to the destination ColorModel more efficiently or with higher quality if it knows some information about how the pixels will be delivered up front.

setHints(int). Method in class java.awt.image.ImageFilter
Filters the information provided in the setHints method of the ImageConsumer interface.

setHints(int). Method in class java.awt.image.PixelGrabber
The setHints method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

setHours(int). Method in class java.util.Date
Sets the hours.

setIconImage(Image). Method in class java.awt.Frame
Sets the image to display when this Frame is iconized.

setIconImage(Image). Method in interface java.awt.peer.FramePeer

setIfModifiedSince(long). Method in class java.net. URLConnection
Some protocols support skipping fetching unless the object is newer than some amount of time.

setLabel(String). Method in class java.awt.Button
Sets the button with the specified label.

setLabel(String). Method in interface java.awt.peer.ButtonPeer

setLabel(String). Method in class java.awt.Checkbox
Sets the button with the specified label.

setLabel(String). Method in interface java.awt.peer.CheckboxPeer

setLabel(String). Method in class java.awt.MenuItem
Sets the label to be the specified label.

setLabel(String). Method in interface java.awt.peer.MenuItemPeer

setLayout(LayoutManager). Method in class java.awt. Container
Sets the layout manager for this container.

setLength(int). Method in class java.lang.StringBuffer
Sets the length of the String.

setLineIncrement(int). Method in class java.awt.Scrollbar
Sets the line increment for this scrollbar.

setLineIncrement(int). Method in interface java.awt.peer.ScrollbarPeer

setLineNumber(int). Method in class java.io. LineNumberInputStream
Sets the current line number.

setMaxPriority(int). Method in class java.lang.ThreadGroup

Sets the maximum priority of the group.

setMenuBar(MenuBar). Method in class java.awt.Frame

Sets the menubar for this Frame to the specified menubar.

setMenuBar(MenuBar). Method in interface java.awt.peer.FramePeer

setMinutes(int). Method in class java.util.Date

Sets the minutes.

setMonth(int). Method in class java.util.Date

Sets the month.

setMultipleSelections(boolean). Method in class java.awt.List

Sets whether this list should allow multiple selections or not.

setMultipleSelections(boolean). Method in interface java.awt.peer.ListPeer

setName(String). Method in class java.lang.Thread

Sets the Thread's name.

setPageIncrement(int). Method in class java.awt.Scrollbar

Sets the page increment for this scrollbar.

setPageIncrement(int). Method in interface java.awt.peer.ScrollbarPeer

setPaintMode(int). Method in class java.awt.Graphics

Sets the paint mode to overwrite the destination with the current color.

setPixels(int, int, int, int, ColorModel, byte[], int, int). Method in class java.awt.image.CropImageFilter

Determine whether the delivered byte pixels intersect the region to be extracted and passes through only that subset of pixels that appear in the output region.

setPixels(int, int, int, int, ColorModel, byte[], int, int). Method in interface java.awt.image.ImageConsumer

The pixels of the image are delivered using one or more calls to the setPixels method.

setPixels(int, int, int, int, ColorModel, byte[], int, int). Method in class java.awt.image.ImageFilter

Filters the information provided in the setPixels method of the ImageConsumer interface which takes an array of bytes.

setPixels(int, int, int, int, ColorModel, byte[], int, int). Method in class java.awt.image.PixelGrabber

The setPixels method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

setPixels(int, int, int, int, ColorModel, byte[], int, int). Method in class java.awt.image.RGBImageFilter

If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel.

setPixels(int, int, int, int, ColorModel, int[], int, int). Method in class java.awt.image.CropImageFilter

Determine if the delivered int pixels intersect the region to be extracted and pass through only that subset of pixels that appear in the output region.

setPixels(int, int, int, int, ColorModel, int[], int, int). Method in interface java.awt.image.ImageConsumer

The pixels of the image are delivered using one or more calls to the setPixels method.

setPixels(int, int, int, int, ColorModel, int[], int, int). Method in class java.awt.image.ImageFilter

Filters the information provided in the setPixels method of the ImageConsumer interface which takes an array of integers.

setPixels(int, int, int, int, ColorModel, int[], int, int). Method in class java.awt.image.PixelGrabber

The setPixels method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

setPixels(int, int, int, int, ColorModel, int[], int, int). Method in class java.awt.image.RGBImageFilter

If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel, otherwise converts the buffer of integer pixels to the default RGB ColorModel and passes the converted buffer to the filterRGBPixels method to be converted one by one.

setPriority(int). Method in class java.lang.Thread

Sets the Thread's priority.

setProperties(Hashtable). Method in class java.awt.image.CropImageFilter

Passes along the properties from the source object after adding a property indicating the cropped region.

setProperties(Hashtable). Method in interface java.awt.image.ImageConsumer

Sets the extensible list of properties associated with this image.

setProperties(Hashtable). Method in class java.awt.image.ImageFilter

Passes the properties from the source object along after adding a property indicating the stream of filters it has been run through.

setProperties(Hashtable). Method in class java.awt.image.PixelGrabber

The setProperties method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

setProperties(Properties). Static method in class java.lang.System

Sets the System properties to the specified properties.

setRequestProperty(String, String). Method in class java.net.URLConnection

Sets/gets a general request property.

setResizable(boolean). Method in class java.awt.Dialog

Sets the resizable flag.

setResizable(boolean). Method in interface java.awt.peer.DialogPeer

setResizable(boolean). Method in class java.awt.Frame

Sets the resizable flag.

setResizable(boolean). Method in interface java.awt.peer.FramePeer

setSeconds(int). Method in class java.util.Date

Sets the seconds.

setSecurityManager(SecurityManager). Static method in class java.lang.System

Sets the System security.

setSeed(long). Method in class java.util.Random

Sets the seed of the random number generator using a single long seed.

setSize(int). Method in class java.util.Vector

Sets the size of the vector.

setSocketFactory(SocketImplFactory). Static method in class java.net. ServerSocket

Sets the system's server SocketImplFactory.

setSocketImplFactory(SocketImplFactory). Static method in class java.net. Socket

Sets the system's client SocketImplFactory.

setSourcePath(String). Method in class sun.tools.debug. RemoteDebugger

Specify the list of paths to use when searching for a source file.

setState(boolean). Method in class java.awt.Checkbox

Sets the Checkbox to the specified boolean state.

setState(boolean). Method in class java.awt.CheckboxMenuItem

Sets the state of this MenuItem if it is a Checkbox.

setState(boolean). Method in interface java.awt.peer.CheckboxMenuItemPeer

setState(boolean). Method in interface java.awt.peer. CheckboxPeer

setStub(AppletStub). Method in class java.applet. Applet

Sets the applet stub.

setText(String). Method in class java.awt.Label

Sets the text for this label to the specified text.

setText(String). Method in interface java.awt.peer.LabelPeer

setText(String). Method in class java.awt. TextComponent

Sets the text of this TextComponent to the specified text.

setText(String). Method in interface java.awt.peer.TextComponentPeer

setTime(long). Method in class java.util.Date

Sets the time.

setTitle(String). Method in class java.awt.Dialog

Sets the title of the Dialog.

setTitle(String). Method in interface java.awt.peer.DialogPeer

setTitle(String). Method in class java.awt.Frame

Sets the title for this Frame to the specified title.

setTitle(String). Method in interface java.awt.peer.FramePeer

setURL(URL, String, String, int, String, String). Method in class java.net.URLStreamHandler

Calls the (protected) set method out of the URL given.

setURLStreamHandlerFactory(URLStreamHandlerFactory). Static method in class java.net. URL

Sets the URLStreamHandler factory.

setUseCaches(boolean). Method in class java.net.URLConnection

Some protocols do caching of documents.

setValue(int). Method in class java.awt.Scrollbar

Sets the value of this Scrollbar to the specified value.

setValue(int). Method in interface java.awt.peer.ScrollbarPeer

setValues(int, int, int, int). Method in class java.awt.Scrollbar

Sets the values for this Scrollbar.

setValues(int, int, int, int). Method in interface java.awt.peer.ScrollbarPeer

setXORMode(Color). Method in class java.awt.Graphics

Sets the paint mode to alternate between the current color and the new specified color.

setYear(int). Method in class java.util.Date

Sets the year.

SHIFT_MASK. Static variable in class java.awt.Event

The shift modifier constant.

shiftDown(()). Method in class java.awt.Event

Checks if the shift key is down.

show(()). Method in class java.awt.Component

Shows the component.

show(()). Method in interface java.awt.peer.ComponentPeer

show(()). Method in class java.awt.Window

Shows the Window.

show(boolean). Method in class java.awt.Component

Conditionally shows the component.

show(Container, String). Method in class java.awt.CardLayout

Flips to the specified component name in the specified container.

showDocument(URL). Method in interface java.applet.AppletContext

Shows a new document.

showDocument(URL, String). Method in interface java.applet.AppletContext

Show a new document in a target window or frame.

showStatus(String). Method in class java.applet.Applet

Shows a status message in the applet's context.

showStatus(String). Method in interface java.applet.AppletContext

Show a status string.

sin(double). Static method in class java.lang.Math

Returns the trigonometric sine of an angle.

SINGLEFRAME. Static variable in interface java.awt.image.ImageConsumer

The image contain a single static image.

SINGLEFRAMEDONE. Static variable in interface java.awt.image.ImageConsumer

One frame of the image is complete but there are more frames to be delivered.

SINGLEPASS. Static variable in interface java.awt.image.ImageConsumer

The pixels will be delivered in a single pass.

size. Variable in class java.awt.Font

The point size of this font.

size(()). Method in class java.util.BitSet

Calculates and returns the set's size

size(()). Method in class java.io.ByteArrayOutputStream

Returns the current size of the buffer.

size(()). Method in class java.awt.Component

Returns the current size of this component.

size(()). Method in class java.io.DataOutputStream

Returns the number of bytes written.

size(()). Method in class java.util.Dictionary

Returns the number of elements contained within the Dictionary.

size(()). Method in class java.util.Hashtable

Returns the number of elements contained in the hashtable.

size(()). Method in class java.util.Vector

Returns the number of elements in the vector.

skip(long). Method in class java.io.BufferedInputStream
Skips n bytes of input.

skip(long). Method in class java.io.ByteArrayInputStream
Skips n bytes of input.

skip(long). Method in class java.io.FileInputStream
Skips n bytes of input.

skip(long). Method in class java.io.FilterInputStream
Skips bytes of input.

skip(long). Method in class java.io.InputStream
Skips n bytes of input.

skip(long). Method in class java.io.LineNumberInputStream
Skips n bytes of input.

skip(long). Method in class java.io.StringBufferInputStream
Skips n bytes of input.

skipBytes(int). Method in interface java.io.DataInput
Skips bytes, block until all bytes are skipped.

skipBytes(int). Method in class java.io.DataInputStream
Skips bytes, blocks until all bytes are skipped.

skipBytes(int). Method in class java.io.RandomAccessFile

slashSlashComments(boolean). Method in class java.io.StreamTokenizer
If the flag is true, recognize C++ style(//) comments.

slashStarComments(boolean). Method in class java.io.StreamTokenizer
If the flag is true, recognize C style(/*) comments.

sleep(long). Static method in class java.lang.Thread
Causes the currently executing Thread to sleep for the specified number of milliseconds.

sleep(long, int). Static method in class java.lang.Thread
Sleep, in milliseconds and additional nanosecond.

Socket(InetAddress, int). Constructor for class java.net.Socket
Creates a stream socket and connects it to the specified address on the specified port.

Socket(InetAddress, int, boolean). Constructor for class java.net.Socket
Creates a socket and connects it to the specified address on the specified port.

Socket(String, int). Constructor for class java.net.Socket
Creates a stream socket and connects it to the specified port on the specified host.

Socket(String, int, boolean). Constructor for class java.net.Socket
Creates a socket and connects it to the specified port on the specified host.

SocketException(). Constructor for class java.net.SocketException
Constructs a new SocketException with no detail message.

SocketException(String). Constructor for class java.net.SocketException
Constructs a new SocketException with the specified detail message.

SocketImpl(). Constructor for class java.net.SocketImpl

SOMEBITS. Static variable in interface java.awt.image.ImageObserver
More pixels needed for drawing a scaled variation of the image are available.

SOUTH. Static variable in class java.awt.GridBagConstraints

SOUTHEAST. Static variable in class java.awt.GridBagConstraints

SOUTHWEST. Static variable in class java.awt.GridBagConstraints

sqrt(double). Static method in class java.lang.Math
Returns the square root of a.

Stack(). Constructor for class java.util.Stack

StackFrame(). Constructor for class sun.tools.debug.StackFrame

StackOverflowError(). Constructor for class java.lang.StackOverflowError
Constructs a StackOverflowError with no detail message.

StackOverflowError(String). Constructor for class java.lang.StackOverflowError
Constructs a StackOverflowError with the specified detail message.

start(). Method in class java.applet.Applet
Called to start the applet.

start(). Method in class java.lang.Thread

Starts this Thread.

startProduction(ImageConsumer). Method in class java.awt.image. FilteredImageSource

Adds an ImageConsumer to the list of consumers interested in data for this image, and immediately starts delivery of the image data through the ImageConsumer interface.

startProduction(ImageConsumer). Method in interface java.awt.image.ImageProducer

This method both registers the given ImageConsumer object as a consumer and starts an immediate reconstruction of the image data which will then be delivered to this consumer and any other consumer which may have already been registered with the producer.

startProduction(ImageConsumer). Method in class java.awt.image. MemoryImageSource

Adds an ImageConsumer to the list of consumers interested in data for this image, and immediately start delivery of the image data through the ImageConsumer interface.

startsWith(String). Method in class java.lang.String

Determines whether this String starts with some prefix.

startsWith(String, int). Method in class java.lang.String

Determines whether this String starts with some prefix.

STATICIMAGEDONE. Static variable in interface java.awt.image.ImageConsumer

The image is complete and there are no more pixels or frames to be delivered.

status(). Method in class java.awt.image.PixelGrabber

Return the status of the pixels.

statusAll(boolean). Method in class java.awt.MediaTracker

Returns the boolean OR of the status of all of the media being tracked.

statusID(int, boolean). Method in class java.awt.MediaTracker

Returns the boolean OR of the status of all of the media with a given ID.

step(boolean). Method in class sun.tools.debug.RemoteThread

Continue execution of this thread to the next instruction or line.

stop(). Method in class java.applet.Applet

Called to stop the applet.

stop(). Method in interface java.applet.AudioClip

Stops playing the clip.

stop(). Method in class sun.tools.debug.RemoteThread

Stop the remote thread.

stop(). Method in class sun.tools.debug.RemoteThreadGroup

Stop the remote threadgroup.

stop(). Method in class java.lang.Thread

Stops a Thread by tossing an object.

stop(). Method in class java.lang.ThreadGroup

Stops all the Threads in this Thread group and all of its sub groups.

stop(Throwable). Method in class java.lang.Thread

Stops a Thread by tossing an object.

StreamTokenizer(InputStream). Constructor for class java.io.StreamTokenizer

Creates a stream tokenizer that parses the specified input stream.

String(). Constructor for class java.lang.String

Constructs a new empty String.

String(byte[], int). Constructor for class java.lang.String

Constructs a new String whose value is the specified array of bytes.

String(byte[], int, int, int). Constructor for class java.lang.String

Constructs a new String whose initial value is the specified sub array of bytes.

String(char[]). Constructor for class java.lang.String

Constructs a new String whose initial value is the specified array of characters.

String(char[], int, int). Constructor for class java.lang.String

Constructs a new String whose initial value is the specified sub array of characters.

String(String). Constructor for class java.lang.String

Constructs a new String that is a copy of the specified String.

String(StringBuffer). Constructor for class java.lang.String

Construct a new string whose value is the current contents of the given string buffer

StringBuffer(StringBuffer). Constructor for class java.lang.StringBuffer
Constructs an empty String buffer.

StringBuffer(int). Constructor for class java.lang.StringBuffer
Constructs an empty String buffer with the specified initial length.

StringBuffer(String). Constructor for class java.lang.StringBuffer
Constructs a String buffer with the specified initial value.

StringBufferInputStream(String). Constructor for class java.io.StringBufferInputStream
Creates an StringBufferInputStream from the specified array of bytes.

StringIndexOutOfBoundsException(StringIndexOutOfBoundsException). Constructor for class java.lang.StringIndexOutOfBoundsException
Constructs a StringIndexOutOfBoundsException with no detail message.

StringIndexOutOfBoundsException(int). Constructor for class java.lang.StringIndexOutOfBoundsException
Constructs a StringIndexOutOfBoundsException initialized with the specified index.

StringIndexOutOfBoundsException(String). Constructor for class java.lang.StringIndexOutOfBoundsException
Constructs a StringIndexOutOfBoundsException with the specified detail message.

StringTokenizer(String). Constructor for class java.util.StringTokenizer
Constructs a StringTokenizer on the specified String, using the default delimiter set (which is " \t\n\r").

StringTokenizer(String, String). Constructor for class java.util.StringTokenizer
Constructs a StringTokenizer on the specified String, using the specified delimiter set.

StringTokenizer(String, String, boolean). Constructor for class java.util.StringTokenizer
Constructs a StringTokenizer on the specified String, using the specified delimiter set.

stringWidth(String). Method in class java.awt.FontMetrics
Returns the width of the specified String in this Font.

style. Variable in class java.awt.Font
The style of the font.

substituteColorModel(ColorModel, ColorModel). Method in class java.awt.image.RGBImageFilter
Registers two ColorModel objects for substitution.

substring(int). Method in class java.lang.String
Returns the substring of this String.

substring(int, int). Method in class java.lang.String
Returns the substring of a String.

suspend(suspend). Method in class sun.tools.debug.RemoteThread
Suspend execution of this thread.

suspend(suspend). Method in class java.lang.Thread
Suspends this Thread's execution.

suspend(suspend). Method in class java.lang.ThreadGroup
Suspends all the Threads in this Thread group and all of its sub groups.

sval. Variable in class java.io.StreamTokenizer
The Stream value.

SW_RESIZE_CURSOR. Static variable in class java.awt.Frame

sync(sync). Method in class java.awt.Toolkit
Syncs the graphics state; useful when doing animation.

T

tan(double). Static method in class java.lang.Math
Returns the trigonometric tangent of an angle.

target. Variable in class java.awt.Event
The target component.

TEXT_CURSOR. Static variable in class java.awt.Frame

TextArea(TextArea). Constructor for class java.awt.TextArea

Constructs a new TextArea.

TextArea(int, int). Constructor for class java.awt.TextArea

Constructs a new TextArea with the specified number of rows and columns.

TextArea(String). Constructor for class java.awt.TextArea

Constructs a new TextArea with the specified text displayed.

TextArea(String, int, int). Constructor for class java.awt.TextArea

Constructs a new TextArea with the specified text and number of rows and columns.

TextField()(). Constructor for class java.awt.TextField

Constructs a new TextField.

TextField(int). Constructor for class java.awt.TextField

Constructs a new TextField initialized with the specified columns.

TextField(String). Constructor for class java.awt.TextField

Constructs a new TextField initialized with the specified text.

TextField(String, int). Constructor for class java.awt.TextField

Constructs a new TextField initialized with the specified text and columns.

Thread()(). Constructor for class java.lang.Thread

Constructs a new Thread.

Thread(Runnable). Constructor for class java.lang.Thread

Constructs a new Thread which applies the run() method of the specified target.

Thread(Runnable, String). Constructor for class java.lang.Thread

Constructs a new Thread with the specified name and applies the run() method of the specified target.

Thread(String). Constructor for class java.lang.Thread

Constructs a new Thread with the specified name.

Thread(ThreadGroup, Runnable). Constructor for class java.lang.Thread

Constructs a new Thread in the specified Thread group that applies the run() method of the specified target.

Thread(ThreadGroup, Runnable, String). Constructor for class java.lang.Thread

Constructs a new Thread in the specified Thread group with the specified name and applies the run() method of the specified target.

Thread(ThreadGroup, String). Constructor for class java.lang.Thread

Constructs a new Thread in the specified Thread group with the specified name.

ThreadDeath()(). Constructor for class java.lang.ThreadDeath

threadDeathEvent(RemoteThread). Method in interface sun.tools.debug.DebuggerCallback

A thread has died.

ThreadGroup(String). Constructor for class java.lang.ThreadGroup

Creates a new ThreadGroup.

ThreadGroup(ThreadGroup, String). Constructor for class java.lang.ThreadGroup

Creates a new ThreadGroup with a specified name in the specified Thread group.

Throwable()(). Constructor for class java.lang.Throwable

Constructs a new Throwable with no detail message.

Throwable(String). Constructor for class java.lang.Throwable

Constructs a new Throwable with the specified detail message.

toBack()(). Method in class java.awt.Window

Sends the frame to the back of the Window.

toBack()(). Method in interface java.awt.peer.WindowPeer

toByteArray()(). Method in class java.io.ByteArrayOutputStream

Returns a copy of the input data.

toCharArray()(). Method in class java.lang.String

Converts this String to a character array.

toExternalForm()(). Method in class java.net.URL

Reverses the parsing of the URL.

toExternalForm(URL). Method in class java.net.URLStreamHandler

Reverses the parsing of the URL.

toFront()(). Method in class java.awt.Window

Brings the frame to the front of the Window.

toFront(). Method in interface java.awt.peer.WindowPeer

toGMTString(). Method in class java.util.Date

Converts a date to a String, using the Internet GMT conventions.

toHex(int). Static method in class sun.tools.debug.RemoteValue

Convert an int to a hexadecimal string.

toLocaleString(). Method in class java.util.Date

Converts a date to a String, using the locale conventions.

toLowerCase(). Method in class java.lang.String

Converts all of the characters in this String to lower case.

toLowerCase(char). Static method in class java.lang.Character

Returns the lower case character value of the specified ISO-LATIN-1 character.

Toolkit(). Constructor for class java.awt.Toolkit

top. Variable in class java.awt.Insets

The inset from the top.

TOPDOWNLEFTRIGHT. Static variable in interface java.awt.image.ImageConsumer

The pixels will be delivered in top-down, left-to-right order.

toString(). Method in class java.util.BitSet

Converts the BitSet to a String.

toString(). Method in class java.lang.Boolean

Returns a new String object representing this Boolean's value.

toString(). Method in class java.awt.BorderLayout

Returns the String representation of this BorderLayout's values.

toString(). Method in class java.io.ByteArrayOutputStream

Converts input data to a string.

toString(). Method in class java.awt.CardLayout

Returns the String representation of this CardLayout's values.

toString(). Method in class java.lang.Character

Returns a String object representing this character's value.

toString(). Method in class java.awt.CheckboxGroup

Returns the String representation of this CheckboxGroup's values.

toString(). Method in class java.lang.Class

Returns the name of this class or interface.

toString(). Method in class java.awt.Color

Returns the String representation of this Color's values.

toString(). Method in class java.awt.Component

Returns the String representation of this Component's values.

toString(). Method in class java.util.Date

Converts a date to a String, using the UNIX ctime conventions.

toString(). Method in class java.awt.Dimension

Returns the String representation of this Dimension's values.

toString(). Method in class java.lang.Double

Returns a String representation of this Double object.

toString(). Method in class java.awt.Event

Returns the String representation of this Event's values.

toString(). Method in class java.io.File

Returns a String object representing this file's path.

toString(). Method in class java.lang.Float

Returns a String representation of this Float object.

toString(). Method in class java.awt.FlowLayout

Returns the String representation of this FlowLayout's values.

toString(). Method in class java.awt.Font

Converts this object to a String representation.

toString(). Method in class java.awt.FontMetrics

Returns the String representation of this FontMetric's values.

toString(). Method in class java.awt.Graphics

Returns a String object representing this Graphic's value.

toString(). Method in class java.awt.GridBagLayout
Returns the String representation of this GridLayout's values.

toString(). Method in class java.awt.GridLayout
Returns the String representation of this GridLayout's values.

toString(). Method in class java.util.Hashtable
Converts to a rather lengthy String.

toString(). Method in class java.net.InetAddress
Converts the InetAddress to a String.

toString(). Method in class java.awt.Insets
Returns a String object representing this Inset's values.

toString(). Method in class java.lang.Integer
Returns a String object representing this Integer's value.

toString(). Method in class java.lang.Long
Returns a String object representing this Long's value.

toString(). Method in class java.awt.MenuComponent
Returns the String representation of this MenuComponent's values.

toString(). Method in class java.lang.Object
Returns a String that represents the value of this Object.

toString(). Method in class java.awt.Point
Returns the String representation of this Point's coordinates.

toString(). Method in class java.awt.Rectangle
Returns the String representation of this Rectangle's values.

toString(). Method in class sun.tools.debug.RemoteArray
Return a string version of the array.

toString(). Method in class sun.tools.debug.RemoteBoolean
Return the boolean's value as a string.

toString(). Method in class sun.tools.debug.RemoteByte
Return the byte's value as a string.

toString(). Method in class sun.tools.debug.RemoteChar
Return the char's value as a string.

toString(). Method in class sun.tools.debug.RemoteClass
Return a (somewhat verbose) description.

toString(). Method in class sun.tools.debug.RemoteDouble
Return the double's value as a string.

toString(). Method in class sun.tools.debug.RemoteField
Returns a String that represents the value of this Object.

toString(). Method in class sun.tools.debug.RemoteFloat
Return the float's value as a string.

toString(). Method in class sun.tools.debug.RemoteInt
Return the int's value as a string.

toString(). Method in class sun.tools.debug.RemoteLong
Return the long's value as a string.

toString(). Method in class sun.tools.debug.RemoteObject
Return object as a string.

toString(). Method in class sun.tools.debug.RemoteShort
Return the short's value as a string.

toString(). Method in class sun.tools.debug.RemoteString
Return the string value, or "null"

toString(). Method in class java.net.ServerSocket
Returns the implementation address and implementation port of this ServerSocket as a String.

toString(). Method in class java.net.Socket
Converts the Socket to a String.

toString(). Method in class java.net.SocketImpl
Returns the address and port of this Socket as a String.

toString(). Method in class sun.tools.debug.StackFrame
Returns a String that represents the value of this Object.

toString(). Method in class java.io.StreamTokenizer
Returns the String representation of the stream token.

toString(). Method in class java.lang.String
Converts this String to a String.

toString(). Method in class java.lang.StringBuffer
Converts to a String representing the data in the buffer.

toString(). Method in class java.lang.Thread
Returns a String representation of the Thread, including the thread's name, priority and thread group.

toString(). Method in class java.lang.ThreadGroup
Returns a String representation of the Thread group.

toString(). Method in class java.lang.Throwable
Returns a short description of the Throwable.

toString(). Method in class java.net.URL
Converts to a human-readable form.

toString(). Method in class java.net.URLConnection
Returns the String representation of the URL connection.

toString(). Method in class java.util.Vector
Converts the vector to a string.

toString(double). Static method in class java.lang.Double
Returns a String representation for the specified double value.

toString(float). Static method in class java.lang.Float
Returns a String representation for the specified float value.

toString(int). Method in class java.io.ByteArrayOutputStream
Converts input data to a string.

toString(int). Static method in class java.lang.Integer
Returns a new String object representing the specified integer.

toString(int, int). Static method in class java.lang.Integer
Returns a new String object representing the specified integer in the specified radix.

toString(long). Static method in class java.lang.Long
Returns a new String object representing the specified integer.

toString(long, int). Static method in class java.lang.Long
Returns a new String object representing the specified long in the specified radix.

totalMemory(). Method in class sun.tools.debug.RemoteDebugger
Report the total memory usage of the Java interpreter being debugged.

totalMemory(). Method in class java.lang.Runtime
Returns the total number of bytes in system memory.

toUpperCase(). Method in class java.lang.String
Converts all of the characters in this String to upper case.

toUpperCase(char). Static method in class java.lang.Character
Returns the upper case character value of the specified ISO-LATIN-1 character.

trace(boolean). Method in class sun.tools.debug.RemoteDebugger
Turn on/off method call tracing.

traceInstructions(boolean). Method in class java.lang.Runtime
Enables/Disables tracing of instructions.

traceMethodCalls(boolean). Method in class java.lang.Runtime
Enables/Disables tracing of method calls.

translate(int, int). Method in class java.awt.Event
Translates an event relative to the given component.

translate(int, int). Method in class java.awt.Graphics
Translates the specified parameters into the origin of the graphics context.

translate(int, int). Method in class java.awt.Point
Translates the point.

translate(int, int). Method in class java.awt.Rectangle
Translates the rectangle.

trim(). Method in class java.lang.String
Trims leading and trailing whitespace from this String.

trimToSize(). Method in class java.util.Vector
Trims the vector's capacity down to size.

TRUE. Static variable in class java.lang.Boolean
Assigns this Boolean to be true.

TT_EOF. Static variable in class java.io.StreamTokenizer
The End-of-file token.

TT_EOL. Static variable in class java.io.StreamTokenizer
The End-of-line token.

TT_NUMBER. Static variable in class java.io.StreamTokenizer
The number token.

TT_WORD. Static variable in class java.io.StreamTokenizer
The word token.

ttype. Variable in class java.io.StreamTokenizer
The type of the last token returned.

typeName(). Method in class sun.tools.debug.RemoteArray
Return this RemoteValue's type ("array").

typeName(). Method in class sun.tools.debug.RemoteBoolean
Print this RemoteValue's type ("boolean").

typeName(). Method in class sun.tools.debug.RemoteByte
Print this RemoteValue's type ("byte").

typeName(). Method in class sun.tools.debug.RemoteChar
Print this RemoteValue's type ("char").

typeName(). Method in class sun.tools.debug.RemoteClass
Returns the name of the class as its type.

typeName(). Method in class sun.tools.debug.RemoteDouble
Print this RemoteValue's type ("double").

typeName(). Method in class sun.tools.debug.RemoteFloat
Print this RemoteValue's type ("float").

typeName(). Method in class sun.tools.debug.RemoteInt
Print this RemoteValue's type ("int").

typeName(). Method in class sun.tools.debug.RemoteLong
Print this RemoteValue's type ("long").

typeName(). Method in class sun.tools.debug.RemoteObject
Returns the RemoteValue's type name ("Object").

typeName(). Method in class sun.tools.debug.RemoteShort
Print this RemoteValue's type ("short").

typeName(). Method in class sun.tools.debug.RemoteString
Print this RemoteValue's type ("String").

typeName(). Method in class sun.tools.debug.RemoteValue
Returns the RemoteValue's type as a string.

U

uncaughtException(Thread, Throwable). Method in class java.lang.ThreadGroup
Called when a thread in this group exists because of an uncaught exception.

UndefinedProperty. Static variable in class java.awt.Image
The UndefinedProperty object should be returned whenever a property which was not defined for a particular image is fetched.

union(Rectangle). Method in class java.awt.Rectangle
Computes the union of two rectangles.

UnknownError(). Constructor for class java.lang.UnknownError
Constructs an UnknownError with no detail message.

UnknownError(String). Constructor for class java.lang.UnknownError

Constructs an `UnknownError` with the specified detail message.

`UnknownHostException`(). Constructor for class `java.net.UnknownHostException`
Constructs a new `UnknownHostException` with no detail message.

`UnknownHostException`(`String`). Constructor for class `java.net.UnknownHostException`
Constructs a new `UnknownHostException` with the specified detail message.

`UnknownServiceException`(). Constructor for class `java.net.UnknownServiceException`
Constructs a new `UnknownServiceException` with no detail message.

`UnknownServiceException`(`String`). Constructor for class `java.net.UnknownServiceException`
Constructs a new `UnknownServiceException` with the specified detail message.

`unread`(`int`). Method in class `java.io.PushbackInputStream`
Pushes back a character.

`UnsatisfiedLinkError`(). Constructor for class `java.lang.UnsatisfiedLinkError`
Constructs an `UnsatisfiedLinkError` with no detail message.

`UnsatisfiedLinkError`(`String`). Constructor for class `java.lang.UnsatisfiedLinkError`
Constructs an `UnsatisfiedLinkError` with the specified detail message.

`UP`. Static variable in class `java.awt.Event`
The up arrow key.

`up`(`int`). Method in class `sun.tools.debug.RemoteThread`
Change the current stackframe to be one or more frames higher (as in, away from the current program counter).

`update`(`Graphics`). Method in class `java.awt.Component`
Updates the component.

`update`(`Observable`, `Object`). Method in interface `java.util.Observer`
Called when observers in the observable list need to be updated.

`url`. Variable in class `java.net.URLConnection`

`URL`(`String`). Constructor for class `java.net.URL`
Creates a `URL` from the unparsed absolute `URL`.

`URL`(`String`, `String`, `int`, `String`). Constructor for class `java.net.URL`
Creates an absolute `URL` from the specified protocol, host, port and file.

`URL`(`String`, `String`, `String`). Constructor for class `java.net.URL`
Creates an absolute `URL` from the specified protocol, host, and file.

`URL`(`URL`, `String`). Constructor for class `java.net.URL`
Creates a `URL` from the unparsed `URL` in the specified context. If spec is an absolute `URL` it is used as is.

`URLConnection`(`URL`). Constructor for class `java.net.URLConnection`
Constructs a `URLConnection` to the specified `URL`.

`URLStreamHandler`(). Constructor for class `java.net.URLStreamHandler`

`useCaches`. Variable in class `java.net.URLConnection`

`UTC`(`int`, `int`, `int`, `int`, `int`, `int`). Static method in class `java.util.Date`
Calculates a `UTC` value from `YMDHMS`.

`UTFDataFormatException`(). Constructor for class `java.io.UTFDataFormatException`
Constructs an `UTFDataFormatException` with no detail message.

`UTFDataFormatException`(`String`). Constructor for class `java.io.UTFDataFormatException`
Constructs an `UTFDataFormatException` with the specified detail message.

V

`valid`(). Method in class `java.io.FileDescriptor`
Determines whether the file descriptor object is valid.

`validate`(). Method in class `java.awt.Component`
Validates a component.

`validate`(). Method in class `java.awt.Container`
Validates this `Container` and all of the components contained within it.

valueOf(boolean). Static method in class java.lang.String
Returns a String object that represents the state of the specified boolean.

valueOf(char). Static method in class java.lang.String
Returns a String object that contains a single character

valueOf(char[]). Static method in class java.lang.String
Returns a String that is equivalent to the specified character array.

valueOf(char[], int, int). Static method in class java.lang.String
Returns a String that is equivalent to the specified character array.

valueOf(double). Static method in class java.lang.String
Returns a String object that represents the value of the specified double.

valueOf(float). Static method in class java.lang.String
Returns a String object that represents the value of the specified float.

valueOf(int). Static method in class java.lang.String
Returns a String object that represents the value of the specified integer.

valueOf(long). Static method in class java.lang.String
Returns a String object that represents the value of the specified long.

valueOf(Object). Static method in class java.lang.String
Returns a String that represents the String value of the object.

valueOf(String). Static method in class java.lang.Boolean
Returns the boolean value represented by the specified String.

valueOf(String). Static method in class java.lang.Double
Returns a new Double value initialized to the value represented by the specified String.

valueOf(String). Static method in class java.lang.Float
Returns the floating point value represented by the specified String.

valueOf(String). Static method in class java.lang.Integer
Assuming the specified String represents an integer, returns a new Integer object initialized to that value.

valueOf(String). Static method in class java.lang.Long
Assuming the specified String represents a long, returns a new Long object initialized to that value.

valueOf(String, int). Static method in class java.lang.Integer
Assuming the specified String represents an integer, returns a new Integer object initialized to that value.

valueOf(String, int). Static method in class java.lang.Long
Assuming the specified String represents a long, returns a new Long object initialized to that value.

Vector(). Constructor for class java.util.Vector
Constructs an empty vector.

Vector(int). Constructor for class java.util.Vector
Constructs an empty vector with the specified storage capacity.

Vector(int, int). Constructor for class java.util.Vector
Constructs an empty vector with the specified storage capacity and the specified capacityIncrement.

VerifyError(). Constructor for class java.lang.VerifyError
Constructor.

VerifyError(String). Constructor for class java.lang.VerifyError
Constructor with a detail message.

VERTICAL. Static variable in class java.awt.GridBagConstraints

VERTICAL. Static variable in class java.awt.Scrollbar
The vertical Scrollbar variable.

VirtualMachineError(). Constructor for class java.lang.VirtualMachineError
Constructs a VirtualMachineError with no detail message.

VirtualMachineError(String). Constructor for class java.lang.VirtualMachineError
Constructs a VirtualMachineError with the specified detail message.

W_RESIZE_CURSOR. Static variable in class java.awt.Frame

wait(). Method in class java.lang.Object

Causes a thread to wait forever until it is notified.

wait(long). Method in class java.lang.Object

Causes a thread to wait until it is notified or the specified timeout expires.

wait(long, int). Method in class java.lang.Object

More accurate wait.

WAIT_CURSOR. Static variable in class java.awt.Frame

waitFor(). Method in class java.lang.Process

Waits for the subprocess to complete.

waitForAll(). Method in class java.awt.MediaTracker

Starts loading all images.

waitForAll(long). Method in class java.awt.MediaTracker

Starts loading all images.

waitForID(int). Method in class java.awt.MediaTracker

Starts loading all images with the specified ID and waits until they have finished loading or receive an error.

waitForID(int, long). Method in class java.awt.MediaTracker

Starts loading all images with the specified ID.

weightx. Variable in class java.awt.GridBagConstraints

weighty. Variable in class java.awt.GridBagConstraints

WEST. Static variable in class java.awt.GridBagConstraints

when. Variable in class java.awt.Event

The time stamp.

white. Static variable in class java.awt.Color

The color white.

whitespaceChars(int, int). Method in class java.io.StreamTokenizer

Specifies that characters in this range are whitespace characters.

width. Variable in class java.awt.Dimension

The width dimension.

WIDTH. Static variable in interface java.awt.image.ImageObserver

The width of the base image is now available and can be taken from the width argument to the `imageUpdate` callback method.

width. Variable in class java.awt.Rectangle

The width of the rectangle.

Window(Frame). Constructor for class java.awt.Window

Constructs a new Window initialized to an invisible state.

WINDOW_DEICONIFY. Static variable in class java.awt.Event

The de-iconify window event.

WINDOW_DESTROY. Static variable in class java.awt.Event

The destroy window event.

WINDOW_EXPOSE. Static variable in class java.awt.Event

The expose window event.

WINDOW_ICONIFY. Static variable in class java.awt.Event

The iconify window event.

WINDOW_MOVED. Static variable in class java.awt.Event

The move window event.

wordChars(int, int). Method in class java.io.StreamTokenizer

Specifies that characters in this range are word characters.

write(byte[]). Method in interface java.io.DataOutput

Writes an array of bytes.

write(byte[]). Method in class java.io.FileOutputStream

Writes an array of bytes.

write(byte[]). Method in class java.io.FilterOutputStream

Writes an array of bytes.

write(byte[]). Method in class java.io.OutputStream
Writes an array of bytes.

write(byte[]). Method in class java.io.RandomAccessFile
Writes an array of bytes.

write(byte[], int, int). Method in class java.io.BufferedOutputStream
Writes a subarray of bytes.

write(byte[], int, int). Method in class java.io.ByteArrayOutputStream
Writes bytes to the buffer.

write(byte[], int, int). Method in interface java.io.DataOutput
Writes a subarray of bytes.

write(byte[], int, int). Method in class java.io.DataOutputStream
Writes a sub array of bytes.

write(byte[], int, int). Method in class java.io.FileOutputStream
Writes a sub array of bytes.

write(byte[], int, int). Method in class java.io.FilterOutputStream
Writes a subarray of bytes.

write(byte[], int, int). Method in class java.io.OutputStream
Writes a sub array of bytes.

write(byte[], int, int). Method in class java.io.PipedOutputStream
Writes a sub array of bytes.

write(byte[], int, int). Method in class java.io.PrintStream
Writes a sub array of bytes.

write(byte[], int, int). Method in class java.io.RandomAccessFile
Writes a sub array of bytes.

write(int). Method in class java.io.BufferedOutputStream
Writes a byte.

write(int). Method in class java.io.ByteArrayOutputStream
Writes a byte to the buffer.

write(int). Method in interface java.io.DataOutput
Writes a byte.

write(int). Method in class java.io.DataOutputStream
Writes a byte.

write(int). Method in class java.io.FileOutputStream
Writes a byte of data.

write(int). Method in class java.io.FilterOutputStream
Writes a byte.

write(int). Method in class java.io.OutputStream
Writes a byte.

write(int). Method in class java.io.PipedOutputStream
Write a byte.

write(int). Method in class java.io.PrintStream
Writes a byte.

write(int). Method in class java.io.RandomAccessFile
Writes a byte of data.

writeBoolean(boolean). Method in interface java.io.DataOutput
Writes a boolean.

writeBoolean(boolean). Method in class java.io.DataOutputStream
Writes a boolean.

writeBoolean(boolean). Method in class java.io.RandomAccessFile
Writes a boolean.

writeByte(int). Method in interface java.io.DataOutput
Writes an 8 bit byte.

writeByte(int). Method in class java.io.DataOutputStream
Writes an 8 bit byte.

writeByte(int). Method in class java.io.RandomAccessFile
Writes a byte.

writeBytes(String). Method in interface java.io.DataOutput
Writes a String as a sequence of bytes.

writeBytes(String). Method in class java.io.DataOutputStream
Writes a String as a sequence of bytes.

writeBytes(String). Method in class java.io.RandomAccessFile
Writes a String as a sequence of bytes.

writeChar(int). Method in interface java.io.DataOutput
Writes a 16 bit char.

writeChar(int). Method in class java.io.DataOutputStream
Writes a 16 bit char.

writeChar(int). Method in class java.io.RandomAccessFile
Writes a character.

writeChars(String). Method in interface java.io.DataOutput
Writes a String as a sequence of chars.

writeChars(String). Method in class java.io.DataOutputStream
Writes a String as a sequence of chars.

writeChars(String). Method in class java.io.RandomAccessFile
Writes a String as a sequence of chars.

writeDouble(double). Method in interface java.io.DataOutput
Writes a 64 bit double.

writeDouble(double). Method in class java.io.DataOutputStream
Writes a 64 bit double.

writeDouble(double). Method in class java.io.RandomAccessFile

writeFloat(float). Method in interface java.io.DataOutput
Writes a 32 bit float.

writeFloat(float). Method in class java.io.DataOutputStream
Writes a 32 bit float.

writeFloat(float). Method in class java.io.RandomAccessFile

writeInt(int). Method in interface java.io.DataOutput
Writes a 32 bit int.

writeInt(int). Method in class java.io.DataOutputStream
Writes a 32 bit int.

writeInt(int). Method in class java.io.RandomAccessFile
Writes an integer.

writeLong(long). Method in interface java.io.DataOutput
Writes a 64 bit long.

writeLong(long). Method in class java.io.DataOutputStream
Writes a 64 bit long.

writeLong(long). Method in class java.io.RandomAccessFile
Writes a long.

writeShort(int). Method in interface java.io.DataOutput
Writes a 16 bit short.

writeShort(int). Method in class java.io.DataOutputStream
Writes a 16 bit short.

writeShort(int). Method in class java.io.RandomAccessFile
Writes a short.

writeTo(OutputStream). Method in class java.io.ByteArrayOutputStream
Writes the contents of the buffer to another stream.

writeUTF(String). Method in interface java.io.DataOutput
Writes a String in UTF format.

writeUTF(String). Method in class java.io.DataOutputStream
Writes a String in UTF format.

writeUTF(String). Method in class java.io.RandomAccessFile
Writes a String in UTF format.

written. Variable in class java.io.DataOutputStream
The number of bytes written so far.

X

- x**. Variable in class java.awt.Event
The x coordinate of the event.
 - x**. Variable in class java.awt.Point
The x coordinate.
 - x**. Variable in class java.awt.Rectangle
The x coordinate of the rectangle.
 - xor**(BitSet). Method in class java.util.BitSet
Logically XORs this bit set with the specified set of bits.
 - xpoints**. Variable in class java.awt.Polygon
The array of x coordinates.
-

Y

- y**. Variable in class java.awt.Event
The y coordinate of the event.
- y**. Variable in class java.awt.Point
The y coordinate.
- y**. Variable in class java.awt.Rectangle
The y coordinate of the rectangle.
- yellow**. Static variable in class java.awt.Color
The color yellow.
- yield**(). Static method in class java.lang.Thread
Causes the currently executing Thread object to yield.
- ypoints**. Variable in class java.awt.Polygon
The array of y coordinates.

Class java.applet.Applet

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.applet.Applet

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Container
            |
            +----java.awt.Panel
                  |
                  +----java.applet.Applet
```

```
public class Applet
extends Panel
```

Base applet class.

Constructor Index

- [Applet\(\)](#)

Method Index

- [destroy\(\)](#)
Cleans up whatever resources are being held.
- [getAppletContext\(\)](#)
Gets a handle to the applet context.
- [getAppletInfo\(\)](#)
Returns a string containing information about the author, version and copyright of the applet.
- [getAudioClip\(URL\)](#)
Gets an audio clip.

Error Index

Gets an audio clip.

Error Index

[getAudioClip\(URL, String\)](#)

[getCodeBase\(\)](#)

Gets the base URL.

Error Index

getDocumentBase()

Gets the document URL.

Error Index

getImage(URL)

Gets an image given a URL.

Error Index

getImage(URL, String)

Gets an image relative to a URL.

Error Index

getParameter(String)

Gets a parameter of the applet.

Error Index

getParameterInfo()

Returns an array of strings describing the parameters that are understood by this applet.

Error Index

init()

Initializes the applet.

Error Index

isActive()

Returns true if the applet is active.

Error Index

play(URL)

Plays an audio clip.

Error Index

play(URL, String)

Plays an audio clip.

Error Index

resize(int, int)

Requests that the applet be resized.

Error Index

resize(Dimension)

Requests that the applet be resized.

Error Index

setStub(AppletStub)

Sets the applet stub.

Error Index

showStatus(String)

Shows a status message in the applet's context.

Error Index

start()

Called to start the applet.

Error Index

stop()

Called to stop the applet.

Constructors

Error Index Applet

```
public Applet()
```

Methods

Error Index setStub

```
public final void setStub(AppletStub stub)
```

Sets the applet stub. This is done by automatically by the system.

Error Index isActive

```
public boolean isActive()
```

Returns true if the applet is active. An applet is marked active just before the start method is called.

See Also:

start

Error Index getDocumentBase

```
public URL getDocumentBase()
```

Gets the document URL. This is the URL of the document in which the applet is embedded.

See Also:

getCodeBase

Error Index getCodeBase

```
public URL getCodeBase()
```


Gets the base URL. This is the URL of the applet itself.

See Also:

[getDocumentBase](#)

Error Index **getParameter**

```
public String getParameter(String name)
```

Gets a parameter of the applet.

Error Index **getAppletContext**

```
public AppletContext getAppletContext()
```

Gets a handle to the applet context. The applet context lets an applet control the applet's environment which is usually the browser or the applet viewer.

Error Index **resize**

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

Requests that the applet be resized.

Overrides:

resize in class Component

Error Index **resize**

```
public void resize(Dimension d)
```

Requests that the applet be resized.

Overrides:

resize in class Component

Error Index **showStatus**

```
public void showStatus(String msg)
```

Shows a status message in the applet's context.

Error Index `getImage`

```
public Image getImage(URL url)
```

Gets an image given a URL. Note that this method always returns an image object immediately, even if the image does not exist. The actual image data is loaded when it is first needed.

Error Index `getImage`

```
public Image getImage(URL url,  
                     String name)
```

Gets an image relative to a URL. This method returns immediately, even if the image does not exist. The actual image data is loaded when it is first needed.

See Also:

getImage

Error Index `getAudioClip`

```
public AudioClip getAudioClip(URL url)
```

Gets an audio clip.

Error Index `getAudioClip`

```
public AudioClip getAudioClip(URL url,  
                             String name)
```

Gets an audio clip.

See Also:

getAudioClip

Error Index `getAppletInfo`

```
public String getAppletInfo()
```

Returns a string containing information about the author, version and copyright of the applet.

Error IndexgetParameterInfo

```
public String[][] getParameterInfo()
```

Returns an array of strings describing the parameters that are understood by this applet. The array consists of sets of three strings: name/type/description. For example:

```
String pinfo[][] = {  
    {"fps", "1-10", "frames per second"},  
    {"repeat", "boolean", "repeat image loop"},  
    {"imgs", "url", "directory in which the images live"}  
};
```

Error Indexplay

```
public void play(URL url)
```

Plays an audio clip. Nothing happens if the audio clip could not be found.

Error Indexplay

```
public void play(URL url,  
                String name)
```

Plays an audio clip. Nothing happens if the audio clip could not be found.

Error Indexinit

```
public void init()
```

Initializes the applet. You never need to call this directly, it is called automatically by the system once the applet is created.

See Also:

start, stop, destroy

Error Index start

public void start()

Called to start the applet. You never need to call this method directly, it is called when the applet's document is visited.

See Also:

[init](#), [stop](#), [destroy](#)

Error Index stop

public void stop()

Called to stop the applet. It is called when the applet's document is no longer on the screen. It is guaranteed to be called before `destroy()` is called. You never need to call this method directly.

See Also:

[init](#), [start](#), [destroy](#)

Error Index destroy

public void destroy()

Cleans up whatever resources are being held. If the applet is active it is stopped.

See Also:

[init](#), [start](#), [stop](#)

Interface java.applet.AppletContext

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.applet.AppletContext

public interface **AppletContext**
extends [Object](#)

This interface corresponds to an applet's environment. It can be used by an applet to obtain information from the applet's environment, which is usually the browser or the applet viewer.

Error Index

Error Index

Gets an applet by name.

[getApplet](#)(String)

Error Index

Enumerates the applets in this context.

[getApplets](#)()

Error Index

Gets an audio clip.

[getAudioClip](#)(URL)

Error Index

Gets an image.

[getImage](#)(URL)

Error Index

Shows a new document.

[showDocument](#)(URL)

Error Index

Show a new document in a target window or frame.

[showDocument](#)(URL, String)

Error Index

Show a status string.

[showStatus](#)(String)

Methods

Error Index

[getAudioClip](#)

public abstract AudioClip getAudioClip(URL url)

Gets an audio clip.

Error IndexgetImage

public abstract Image getImage(URL url)

Gets an image. This usually involves downloading it over the net. However, the environment may decide to cache images. This method takes an array of URLs, each of which will be tried until the image is found.

Error IndexgetApplet

public abstract Applet getApplet(String name)

Gets an applet by name.

Returns:

null if the applet does not exist.

Error IndexgetApplets

public abstract Enumeration getApplets()

Enumerates the applets in this context. Only applets that are accessible will be returned. This list always includes the applet itself.

Error IndexshowDocument

public abstract void showDocument(URL url)

Shows a new document. This may be ignored by the applet context.

Error IndexshowDocument

public abstract void showDocument(URL url,
String target)

Show a new document in a target window or frame. This may be ignored by the applet context. This

method accepts the target strings: `_self` show in current frame `_parent` show in parent frame
`_top` show in top-most frame `_blank` show in new unnamed top-level window show in
new top-level window named

Error Index

showStatus

public abstract void showStatus(String status)

Show a status string.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.applet.AppletStub

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.applet.AppletStub

public interface **AppletStub**
extends [Object](#)

This interface is used to implement an applet viewer. It is not normally used by applet programmers.

Error Index

Error Index

appletResize(int, int)

Called when the applet wants to be resized.

Error Index

getAppletContext()

Gets a handler to the applet's context.

Error Index

getCodeBase()

Gets the base URL.

Error Index

getDocumentBase()

Gets the document URL.

Error Index

getParameter(String)

Gets a parameter of the applet.

Error Index

isActive()

Returns true if the applet is active.

Methods

Error Index

isActive

public abstract boolean **isActive**()

Returns true if the applet is active.

Error Index

getDocumentBase

```
public abstract URL getDocumentBase()
```

Gets the document URL.

Error Index

getCodeBase

```
public abstract URL getCodeBase()
```

Gets the base URL.

Error Index

getParameter

```
public abstract String getParameter(String name)
```

Gets a parameter of the applet.

Error Index

getAppletContext

```
public abstract AppletContext getAppletContext()
```

Gets a handler to the applet's context.

Error Index

appletResize

```
public abstract void appletResize(int width,  
                                   int height)
```

Called when the applet wants to be resized.

Interface java.applet.AudioClip

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.applet.AudioClip

public interface **AudioClip**
extends [Object](#)

A very high level abstraction of audio.

Error Index

Error Index

[loop\(\)](#)

Starts playing the clip in a loop.

Error Index

[play\(\)](#)

Starts playing the clip.

Error Index

[stop\(\)](#)

Stops playing the clip.

Methods

Error Index

[play](#)

public abstract void play()

Starts playing the clip. Each time this method is called, the clip is restarted from the beginning.

Error Index

[loop](#)

public abstract void loop()

Starts playing the clip in a loop.

Error Index stop

public abstract void stop()

Stops playing the clip.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.AWTError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.AWTError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.awt.AWTError
```

```
public class AWTError
    extends Error
```

An AWT Error.

Error Index

Error Index

[AWTError](#)(String)

Constructors

Error Index

AWTError

```
public AWTError(String msg)
```

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.AWTException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.AWTException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.awt.AWTException
```

```
public class AWTException
extends Exception
```

Signals that an Abstract Window Toolkit exception has occurred.

Error Index

Error Index

[AWTException](#)(String)

Constructs an AWTException with the specified detail message.

Constructors

Error Index

[AWTException](#)

```
public AWTException(String msg)
```

Constructs an AWTException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

msg - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.BorderLayout

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.BorderLayout

[java.lang.Object](#)

|
+----java.awt.BorderLayout

```
public class BorderLayout
  extends Object
  implements LayoutManager
```

A TNT style border bag layout. It will layout a container using members named "North", "South", "East", "West" and "Center". The "North", "South", "East" and "West" components get layed out according to their preferred sizes and the constraints of the container's size. The "Center" component will get any space left over.

Error Index

Error Index

[**BorderLayout\(\)**](#)

Constructs a new BorderLayout.

Error Index

[**BorderLayout\(int, int\)**](#)

Constructs a BorderLayout with the specified gaps.

Error Index

Error Index

[**addLayoutComponent\(String, Component\)**](#)

Adds the specified named component to the layout.

Error Index

[**layoutContainer\(Container\)**](#)

Lays out the specified container.

Error Index

[**minimumLayoutSize\(Container\)**](#)

Returns the minimum dimensions needed to layout the components contained in the specified target

container.

Error Index

preferredLayoutSize(Container)

Returns the preferred dimensions for this layout given the components in the specified target container.

Error Index

removeLayoutComponent(Component)

Removes the specified component from the layout.

Error Index

toString()

Returns the String representation of this BorderLayout's values.

Error Index

Error Index

BorderLayout

public BorderLayout()

Constructs a new BorderLayout.

Error Index

BorderLayout

public BorderLayout(int hgap,
int vgap)

Constructs a BorderLayout with the specified gaps.

Parameters:

hgap - the horizontal gap

vgap - the vertical gap

Error Index

Error Index

addLayoutComponent

public void addLayoutComponent(String name,
Component comp)

Adds the specified named component to the layout.

Parameters:

name - the String name
comp - the component to be added

Error Index

removeLayoutComponent

```
public void removeLayoutComponent(Component comp)
```

Removes the specified component from the layout.

Parameters:

comp - the component to be removed

Error Index

minimumLayoutSize

```
public Dimension minimumLayoutSize(Container target)
```

Returns the minimum dimensions needed to layout the components contained in the specified target container.

Parameters:

target - the Container on which to do the layout

See Also:

Container, preferredLayoutSize

Error Index

preferredLayoutSize

```
public Dimension preferredLayoutSize(Container target)
```

Returns the preferred dimensions for this layout given the components in the specified target container.

Parameters:

target - the component which needs to be laid out

See Also:

Container, minimumLayoutSize

Error Index

layoutContainer

```
public void layoutContainer(Container target)
```

Lays out the specified container. This method will actually reshape the components in the specified target container in order to satisfy the constraints of the BorderLayout object.

Parameters:

target - the component being laid out

See Also:

[Container](#)

Error Index toString

public [String](#) toString()

Returns the String representation of this BorderLayout's values.

Overrides:

[toString](#) in class [Object](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Button

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Button

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Button
```

```
public class Button
    extends Component
```

A class that produces a labeled button component.

Error Index

Error Index

Button()

Constructs a Button with no label.

Error Index

Button(String)

Constructs a Button with a string label.

Error Index

Error Index

addNotify()

Creates the peer of the button.

Error Index

getLabel()

Gets the label of the button.

Error Index

paramString()

Returns the parameter String of this button.

Error Index

setLabel(String)

Sets the button with the specified label.

Error Index

Error Index

Button

public Button()

Constructs a Button with no label.

Error Index

Button

public Button(String label)

Constructs a Button with a string label.

Parameters:

label - the button label

Error Index

Error Index

addNotify

public synchronized void addNotify()

Creates the peer of the button. This peer allows us to change the look of the button without changing its functionality.

Overrides:

addNotify in class Component

Error Index

getLabel

public String getLabel()

Gets the label of the button.

See Also:

[setLabel](#)

Error Index

setLabel

public void setLabel([String](#) label)

Sets the button with the specified label.

Parameters:

label - the label to set the button with

See Also:

[getLabel](#)

Error Index

paramString

protected [String](#) paramString()

Returns the parameter String of this button.

Overrides:

[paramString](#) in class [Component](#)

Class java.awt.Canvas

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Canvas

[java.lang.Object](#)

```
|
+----java.awt.Component
      |
      +----java.awt.Canvas
```

public class **Canvas**
extends [Component](#)

A Canvas component. This is a generic component which needs to be subclassed in order to add some interesting functionality.

Error Index

Error Index

[Canvas\(\)](#)

Error Index

Error Index

[addNotify\(\)](#)

Creates the peer of the canvas.

Error Index

[paint](#)(Graphics)

Paints the canvas in the default background color.

Error Index

Error Index

Canvas

public Canvas()

Error Index

Error Index addNotify

public synchronized void addNotify()

Creates the peer of the canvas. This peer allows you to change the user interface of the canvas without changing its functionality.

Overrides:

addNotify in class Component

Error Index paint

public void paint(Graphics g)

Paints the canvas in the default background color.

Parameters:

g - the specified Graphics window

Overrides:

paint in class Component

Class java.awt.CardLayout

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.CardLayout

[java.lang.Object](#)

|

+----java.awt.CardLayout

```
public class CardLayout
    extends Object
    implements LayoutManager
```

A layout manager for a container that contains several 'cards'. Only one card is visible at a time, allowing you to flip through the cards.

Error Index

Error Index

[CardLayout\(\)](#)

Creates a new card layout.

Error Index

[CardLayout\(int, int\)](#)

Creates a card layout with the specified gaps.

Error Index

Error Index

[addLayoutComponent\(String, Component\)](#)

Adds the specified component with the specified name to the layout.

Error Index

[first\(Container\)](#)

Flip to the first card.

Error Index

[last\(Container\)](#)

Flips to the last card of the specified container.

Error Index

layoutContainer(Container)

Performs a layout in the specified panel.

Error Index

minimumLayoutSize(Container)

Calculates the minimum size for the specified panel.

Error Index

next(Container)

Flips to the next card of the specified container.

Error Index

preferredLayoutSize(Container)

Calculates the preferred size for the specified panel.

Error Index

previous(Container)

Flips to the previous card of the specified container.

Error Index

removeLayoutComponent(Component)

Removes the specified component from the layout.

Error Index

show(Container, String)

Flips to the specified component name in the specified container.

Error Index

toString()

Returns the String representation of this CardLayout's values.

Error Index

Error Index

CardLayout

public CardLayout()

Creates a new card layout.

Error Index

CardLayout

public CardLayout(int hgap,
int vgap)

Creates a card layout with the specified gaps.

Parameters:

hgap - the horizontal gap

vgap - the vertical gap

Error Index

Error Index

addLayoutComponent

```
public void addLayoutComponent(String name,  
                               Component comp)
```

Adds the specified component with the specified name to the layout.

Parameters:

name - the name of the component
comp - the component to be added

Error Index

removeLayoutComponent

```
public void removeLayoutComponent(Component comp)
```

Removes the specified component from the layout.

Parameters:

comp - the component to be removed

Error Index

preferredLayoutSize

```
public Dimension preferredLayoutSize(Container parent)
```

Calculates the preferred size for the specified panel.

Parameters:

parent - the name of the parent container

Returns:

the dimensions of this panel.

See Also:

minimumLayoutSize

Error Index

minimumLayoutSize

```
public Dimension minimumLayoutSize(Container parent)
```

Calculates the minimum size for the specified panel.

Parameters:

parent - the name of the parent container

Returns:

the dimensions of this panel.

See Also:

preferredLayoutSize

Error Index layoutContainer

public void layoutContainer(Container parent)

Performs a layout in the specified panel.

Parameters:

parent - the name of the parent container

Error Index first

public void first(Container parent)

Flip to the first card.

Parameters:

parent - the name of the parent container

Error Index next

public void next(Container parent)

Flips to the next card of the specified container.

Parameters:

parent - the name of the container

Error Index previous

public void previous(Container parent)

Flips to the previous card of the specified container.

Parameters:

parent - the name of the parent container

Error Indexlast

```
public void last(Container parent)
```

Flips to the last card of the specified container.

Parameters:

parent - the name of the parent container

Error Indexshow

```
public void show(Container parent,  
                String name)
```

Flips to the specified component name in the specified container.

Parameters:

parent - the name of the parent container

name - the component name

Error IndextoString

```
public String toString()
```

Returns the String representation of this CardLayout's values.

Overrides:

toString in class Object

Class java.awt.Checkbox

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Checkbox

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Checkbox
```

```
public class Checkbox
    extends Component
```

A Checkbox object is a graphical user interface element that has a boolean state.

Error Index

Error Index

Checkbox()

Constructs a Checkbox with no label, no Checkbox group, and initialized to a false state.

Error Index

Checkbox(String)

Constructs a Checkbox with the specified label, no Checkbox group, and initialized to a false state.

Error Index

Checkbox(String, CheckboxGroup, boolean)

Constructs a Checkbox with the specified label, specified Checkbox group, and specified boolean state.

Error Index

Error Index

addNotify()

Creates the peer of the Checkbox.

Error Index

getCheckboxGroup()

Returns the checkbox group.

Error Index

getLabel()

Gets the label of the button.

Error Index

getState()

Returns the boolean state of the Checkbox.

Error Index

paramString()

Returns the parameter String of this Checkbox.

Error Index

setCheckboxGroup(CheckboxGroup)

Sets the CheckboxGroup to the specified group.

Error Index

setLabel(String)

Sets the button with the specified label.

Error Index

setState(boolean)

Sets the Checkbox to the specified boolean state.

Error Index

Error Index

Checkbox

public Checkbox()

Constructs a Checkbox with no label, no Checkbox group, and initialized to a false state.

Error Index

Checkbox

public Checkbox(String label)

Constructs a Checkbox with the specified label, no Checkbox group, and initialized to a false state.

Parameters:

label - the label on the Checkbox

Error Index

Checkbox

public Checkbox(String label,
CheckboxGroup group,
boolean state)

Constructs a Checkbox with the specified label, specified Checkbox group, and specified boolean state. If the specified CheckboxGroup is not equal to null, then this Checkbox becomes a Checkbox button. If the Checkbox becomes a button, this simply means that only one Checkbox in a CheckboxGroup may be set at a time.

Parameters:

label - the label on the Checkbox
group - the CheckboxGroup this Checkbox is in
state - is the initial state of this Checkbox

Error Index

Error Index addNotify

```
public synchronized void addNotify()
```

Creates the peer of the Checkbox. The peer allows you to change the look of the Checkbox without changing its functionality.

Overrides:

addNotify in class Component

Error Index getLabel

```
public String getLabel()
```

Gets the label of the button.

See Also:

setLabel

Error Index setLabel

```
public void setLabel(String label)
```

Sets the button with the specified label.

Parameters:

label - the label of the button

See Also:

getLabel

Error Index `getState`

```
public boolean getState()
```

Returns the boolean state of the Checkbox.

See Also:

[setState](#)

Error Index `setState`

```
public void setState(boolean state)
```

Sets the Checkbox to the specified boolean state.

Parameters:

state - the boolean state

See Also:

[getState](#)

Error Index `getCheckboxGroup`

```
public CheckboxGroup getCheckboxGroup()
```

Returns the checkbox group.

See Also:

[setCheckboxGroup](#)

Error Index `setCheckboxGroup`

```
public void setCheckboxGroup(CheckboxGroup g)
```

Sets the CheckboxGroup to the specified group.

Parameters:

g - the new CheckboxGroup

See Also:

[getCheckboxGroup](#)

Error Index paramString

protected String paramString()

Returns the parameter String of this Checkbox.

Overrides:

paramString in class Component

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.CheckboxGroup

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.CheckboxGroup

[java.lang.Object](#)

|
+----java.awt.CheckboxGroup

```
public class CheckboxGroup  
extends Object
```

This class is used to create a multiple-exclusion scope for a set of Checkbox buttons. For example, creating a set of Checkbox buttons with the same CheckboxGroup object means that only one of those Checkbox buttons will be allowed to be "on" at a time.

Error Index

Error Index

CheckboxGroup()

Creates a new CheckboxGroup.

Error Index

Error Index

getCurrent()

Gets the current choice.

Error Index

setCurrent(Checkbox)

Sets the current choice to the specified Checkbox.

Error Index

toString()

Returns the String representation of this CheckboxGroup's values.

Error Index

Error Index CheckboxGroup

public CheckboxGroup()

Creates a new CheckboxGroup.

Error Index

Error Index getCurrent

public Checkbox getCurrent()

Gets the current choice.

Error Index setCurrent

public synchronized void setCurrent(Checkbox box)

Sets the current choice to the specified Checkbox. If the Checkbox belongs to a different group, just return.

Parameters:

box - the current Checkbox choice

Error Index toString

public String toString()

Returns the String representation of this CheckboxGroup's values. Convert to String.

Overrides:

toString in class Object

Class java.awt.CheckboxMenuItem

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.CheckboxMenuItem

```
java.lang.Object
|
+----java.awt.MenuComponent
      |
      +----java.awt.MenuItem
            |
            +----java.awt.CheckboxMenuItem
```

```
public class CheckboxMenuItem
    extends MenuItem
```

This class produces a checkbox that represents a choice in a menu.

Error Index

Error Index

[CheckboxMenuItem](#)(String)

Creates the checkbox item with the specified label.

Error Index

Error Index

[addNotify](#)()

Creates the peer of the checkbox item.

Error Index

[getState](#)()

Returns the state of this MenuItem.

Error Index

[paramString](#)()

Returns the parameter String of this button.

Error Index

[setState](#)(boolean)

Sets the state of this MenuItem if it is a Checkbox.

Error Index

Error Index

CheckboxMenuItem

```
public CheckboxMenuItem(String label)
```

Creates the checkbox item with the specified label.

Parameters:

label - the button label

Error Index

Error Index

addNotify

```
public synchronized void addNotify()
```

Creates the peer of the checkbox item. This peer allows us to change the look of the checkbox item without changing its functionality.

Overrides:

addNotify in class MenuItem

Error Index

getState

```
public boolean getState()
```

Returns the state of this MenuItem. This method is only valid for a Checkbox.

Error Index

setState

```
public void setState(boolean t)
```

Sets the state of this MenuItem if it is a Checkbox.

Parameters:

t - the specified state of the checkbox

Error Index paramString

public String paramString()

Returns the parameter String of this button.

Overrides:

paramString in class MenuItem

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Choice

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Choice

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Choice
```

```
public class Choice
    extends Component
```

The Choice class is a pop-up menu of choices. The current choice is displayed as the title of the menu.

Error Index

Error Index

Choice()

Constructs a new Choice.

Error Index

Error Index

addItem(String)

Adds an item to this Choice.

Error Index

addNotify()

Creates the Choice's peer.

Error Index

countItems()

Returns the number of items in this Choice.

Error Index

getItem(int)

Returns the String at the specified index in the Choice.

Error Index

getSelectedIndex()

Returns the index of the currently selected item.

Error Index

getSelectedItem()

Returns a String representation of the current choice.

Error Index

paramString()

Returns the parameter String of this Choice.

Error Index

select(int)

Selects the item with the specified position.

Error Index

select(String)

Selects the item with the specified String.

Error Index

Error Index

Choice

public Choice()

Constructs a new Choice.

Error Index

Error Index

addNotify

public synchronized void addNotify()

Creates the Choice's peer. This peer allows us to change the look of the Choice without changing its functionality.

Overrides:

addNotify in class Component

Error Index

countItems

public int countItems()

Returns the number of items in this Choice.

See Also:

getItem

Error Index `getItem`

```
public String getItem(int index)
```

Returns the String at the specified index in the Choice.

Parameters:

index - the index at which to begin

See Also:

countItems

Error Index `addItem`

```
public synchronized void addItem(String item)
```

Adds an item to this Choice.

Parameters:

item - the item to be added

Throws: NullPointerException

If the item's value is equal to null.

Error Index `getSelectedItem`

```
public String getSelectedItem()
```

Returns a String representation of the current choice.

See Also:

getSelectedIndex

Error Index `getSelectedIndex`

```
public int getSelectedIndex()
```

Returns the index of the currently selected item.

See Also:

[getSelectedItem](#)

Error Index **select**

public synchronized void select(int pos)

Selects the item with the specified position.

Parameters:

pos - the choice item position

Throws: [IllegalArgumentException](#)

If the choice item position is invalid.

See Also:

[getSelectedItem](#), [getSelectedIndex](#)

Error Index **select**

public void select([String](#) str)

Selects the item with the specified String.

Parameters:

str - the specified String

See Also:

[getSelectedItem](#), [getSelectedIndex](#)

Error Index **paramString**

protected [String](#) paramString()

Returns the parameter String of this Choice.

Overrides:

[paramString](#) in class [Component](#)

Class java.awt.Color

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Color

[java.lang.Object](#)

|

+----java.awt.Color

public final class **Color**
extends [Object](#)

A class to encapsulate RGB Colors.

Variable Index

- [black](#)
The color black.
- [blue](#)
The color blue.
- [cyan](#)
The color cyan.
- [darkGray](#)
The color dark gray.
- [gray](#)
The color gray.
- [green](#)
The color green.

Error Index

The color light gray.

[lightGray](#)

Error Index

The color magenta.

[magenta](#)

Error Index

The color orange.

[orange](#)

Error Index

The color pink.

[pink](#)

Error Index

The color red.

red

Error Index

The color white.

white

Error Index

The color yellow.

yellow

Error Index

Error Index

Color(int, int, int)

Creates a color with the specified red, green, and blue values in the range (0 - 255).

Error Index

Color(int)

Creates a color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7.

Error Index

Color(float, float, float)

Creates a color with the specified red, green, and blue values in the range (0.0 - 1.0).

Error Index

- HSBtoRGB(float, float, float)
Returns the RGB value defined by the default RGB ColorModel, of the color corresponding to the given HSB color components.
- RGBtoHSB(int, int, int, float[])
Returns the HSB values corresponding to the color defined by the red, green, and blue components.

Error Index

brighter()

Returns a brighter version of this color.

Error Index

darker()

Returns a darker version of this color.

Error Index

equals(Object)

Compares this object against the specified object.

Error Index

getBlue()

Gets the blue component.

- getColor(String)
Gets the specified Color property.

- getColor(String, Color)
Gets the specified Color property of the specified Color.
- getColor(String, int)
Gets the specified Color property of the color value.

Error Index

- getGreen()
Gets the green component.
- getHSBColor(float, float, float)
A static Color factory for generating a Color object from HSB values.

Error Index

- getRGB()
Gets the RGB value representing the color in the default RGB ColorModel.

Error Index

- getRed()
Gets the red component.

Error Index

- hashCode()
Computes the hash code.

Error Index

- toString()
Returns the String representation of this Color's values.

Variables

Error Index

white

```
public final static Color white
```

The color white.

Error Index

lightGray

```
public final static Color lightGray
```

The color light gray.

Error Index

gray

```
public final static Color gray
```

The color gray.

Error Index darkGray

public final static Color darkGray

The color dark gray.

Error Index black

public final static Color black

The color black.

Error Index red

public final static Color red

The color red.

Error Index pink

public final static Color pink

The color pink.

Error Index orange

public final static Color orange

The color orange.

Error Index yellow

public final static Color yellow

The color yellow.

Error Index green

```
public final static Color green
```

The color green.

Error Index **magenta**

```
public final static Color magenta
```

The color magneta.

Error Index **cyan**

```
public final static Color cyan
```

The color cyan.

Error Index **blue**

```
public final static Color blue
```

The color blue.

Error Index

Error Index **Color**

```
public Color(int r,  
             int g,  
             int b)
```

Creates a color with the specified red, green, and blue values in the range (0 - 255). The actual color used in rendering will depend on finding the best match given the color space available for a given output device.

Parameters:

r - the red component
g - the green component
b - the blue component

See Also:

[getRed](#), [getGreen](#), [getBlue](#), [getRGB](#)

Error Index Color

public Color(int rgb)

Creates a color with the specified combined RGB value consisting of the red component in bits 16-23, the green component in bits 8-15, and the blue component in bits 0-7. The actual color used in rendering will depend on finding the best match given the color space available for a given output device.

Parameters:

rgb - the combined RGB components

See Also:

[getRGBdefault](#), [getRed](#), [getGreen](#), [getBlue](#), [getRGB](#)

Error Index Color

public Color(float r,
float g,
float b)

Creates a color with the specified red, green, and blue values in the range (0.0 - 1.0). The actual color used in rendering will depend on finding the best match given the color space available for a given output device.

Parameters:

r - the red component

g - the red component

b - the red component

See Also:

[getRed](#), [getGreen](#), [getBlue](#), [getRGB](#)

Error Index

Error Index getRed

public int getRed()

Gets the red component.

See Also:

[getRGB](#)

Error Index **getGreen**

```
public int getGreen()
```

Gets the green component.

See Also:

getRGB

Error Index **getBlue**

```
public int getBlue()
```

Gets the blue component.

See Also:

getRGB

Error Index **getRGB**

```
public int getRGB()
```

Gets the RGB value representing the color in the default RGB ColorModel. (Bits 24-31 are 0xff, 16-23 are red, 8-15 are green, 0-7 are blue).

See Also:

getRGBdefault, getRed, getGreen, getBlue

Error Index **brighter**

```
public Color brighter()
```

Returns a brighter version of this color.

Error Index **darker**

```
public Color darker()
```

Returns a darker version of this color.

Error Index

hashCode

```
public int hashCode()
```

Computes the hash code.

Overrides:

hashCode in class Object

Error Index

equals

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

Compares this object against the specified object.

Parameters:

obj - the object to compare with.

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index

toString

```
public String toString()
```

Returns the String representation of this Color's values.

Overrides:

toString in class Object

Error Index

getColor

```
public static Color getColor(String nm)
```

Gets the specified Color property.

Parameters:

nm - the name of the color property

Error Index

getColor

```
public static Color getColor(String nm,  
                             Color v)
```

Gets the specified Color property of the specified Color.

Parameters:

nm - the name of the color property

v - the specified color

Returns:

the new color.

Error Index

getColor

```
public static Color getColor(String nm,  
                             int v)
```

Gets the specified Color property of the color value.

Parameters:

nm - the name of the color property

v - the color value

Returns:

the new color.

Error Index

HSBtoRGB

```
public static int HSBtoRGB(float hue,  
                           float saturation,  
                           float brightness)
```

Returns the RGB value defined by the default RGB ColorModel, of the color corresponding to the given HSB color components.

Parameters:

hue - the hue component of the color

saturation - the saturation of the color

brightness - the brightness of the color

See Also:

getRGBdefault, getRGB

Error Index

RGBtoHSB

```
public static float[] RGBtoHSB(int r,  
                                int g,  
                                int b,  
                                float hsbvals[])
```

Returns the HSB values corresponding to the color defined by the red, green, and blue components.

Parameters:

r - the red component of the color
g - the green component of the color
b - the blue component of the color
hsbvals - the array to be used to return the 3 HSB values, or null

Returns:

the array used to store the results [hue, saturation, brightness]

See Also:

[getRGBdefault](#), [getRGB](#)

Error Index [getHSBColor](#)

```
public static Color getHSBColor(float h,  
                                float s,  
                                float b)
```

A static Color factory for generating a Color object from HSB values.

Parameters:

h - the hue component
s - the saturation of the color
b - the brightness of the color

Returns:

the Color object for the corresponding RGB color

Class java.awt.Component

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Component

[java.lang.Object](#)

|

+----java.awt.Component

```
public class Component
  extends Object
  implements ImageObserver
```

A generic Abstract Window Toolkit component.

Error Index

Error Index

action(Event, Object)

Called if an action occurs in the Component.

Error Index

addNotify()

Notifies the Component to create a peer.

Error Index

bounds()

Returns the current bounds of this component.

Error Index

checkImage(Image, ImageObserver)

Returns the status of the construction of a screen representation of the specified image.

Error Index

checkImage(Image, int, int, ImageObserver)

Returns the status of the construction of a scaled screen representation of the specified image.

Error Index

createImage(ImageProducer)

Creates an image from the specified image producer.

Error Index

createImage(int, int)

Creates an off-screen drawable Image to be used for double buffering.

Error Index

deliverEvent(Event)

Delivers an event to this component or one of its sub components.

Error Index

disable()

Disables a component.

Error Index

enable()

Enables a component.

Error Index

enable(boolean)

Conditionally enables a component.

Error Index

getBackground()

Gets the background color.

Error Index

getColorModel()

Gets the ColorModel used to display the component on the output device.

Error Index

getFont()

Gets the font of the component.

Error Index

getFontMetrics(Font)

Gets the font metrics for this component.

Error Index

getForeground()

Gets the foreground color.

Error Index

getGraphics()

Gets a Graphics context for this component.

Error Index

getParent()

Gets the parent of the component.

Error Index

getPeer()

Gets the peer of the component.

Error Index

getToolkit()

Gets the toolkit of the component.

Error Index

gotFocus(Event, Object)

Indicates that this component has received the input focus.

Error Index

handleEvent(Event)

Handles the event.

Error Index

Hides the component.

hide()

Error Index

Repaints the component when the image has changed.

imageUpdate(Image, int, int, int, int, int)

Error Index

Checks whether a specified x,y location is "inside" this Component.

inside(int, int)

Error Index

Invalidates a component.

invalidate()

Error Index

Checks if this Component is enabled.

isEnabled()

Error Index

Checks if this Component is showing on screen.

isShowing()

Error Index

Checks if this Component is valid.

isValid()

Error Index

Checks if this Component is visible.

isVisible()

Error Index

Called if a character is pressed.

keyDown(Event, int)

Error Index

Called if a character is released.

keyUp(Event, int)

Error Index

Lays out the component.

layout()

Error Index

Prints a listing to a print stream.

list()

Error Index

Prints a listing to the specified print out stream.

list(PrintStream)

Error Index

Prints out a list, starting at the specified indentation, to the specified print stream.

list(PrintStream, int)

Error Index

Returns the component or subcomponent that contains the x,y location.

locate(int, int)

Error Index

location()

Returns the current location of this component.

Error Index

lostFocus(Event, Object)

Indicates that this component has lost the input focus.

Error Index

minimumSize()

Returns the minimum size of this component.

Error Index

mouseDown(Event, int, int)

Called if the mouse is down.

Error Index

mouseDrag(Event, int, int)

Called if the mouse is dragged (the mouse button is down).

Error Index

mouseEnter(Event, int, int)

Called when the mouse enters the component.

Error Index

mouseExit(Event, int, int)

Called when the mouse exits the component.

Error Index

mouseMove(Event, int, int)

Called if the mouse moves (the mouse button is up).

Error Index

mouseUp(Event, int, int)

Called if the mouse is up.

Error Index

move(int, int)

Moves the Component to a new location.

Error Index

nextFocus()

Moves the focus to the next component.

Error Index

paint(Graphics)

Paints the component.

Error Index

paintAll(Graphics)

Paints the component and its subcomponents.

Error Index

paramString()

Returns the parameter String of this Component.

Error Index

postEvent(Event)

Posts an event to this component.

Error Index

preferredSize()

Returns the preferred size of this component.

Error Index

prepareImage(Image, ImageObserver)

Prepares an image for rendering on this Component.

Error Index

prepareImage(Image, int, int, ImageObserver)

Prepares an image for rendering on this Component at the specified width and height.

Error Index

print(Graphics)

Prints this component.

Error Index

printAll(Graphics)

Prints the component and its subcomponents.

Error Index

removeNotify()

Notifies the Component to destroy the peer.

Error Index

repaint()

Repaints the component.

Error Index

repaint(long)

Repaints the component.

Error Index

repaint(int, int, int, int)

Repaints part of the component.

Error Index

repaint(long, int, int, int, int)

Repaints part of the component.

Error Index

requestFocus()

Requests the input focus.

Error Index

reshape(int, int, int, int)

Reshapes the Component to the specified bounding box.

Error Index

resize(int, int)

Resizes the Component to the specified width and height.

Error Index

resize(Dimension)

Resizes the Component to the specified dimension.

Error Index

setBackground(Color)

Sets the background color.

Error Index

setFont(Font)

Sets the font of the component.

Error Index

setForeground(Color)

Sets the foreground color.

Error Index

show()

Shows the component.

Error Index

show(boolean)

Conditionally shows the component.

Error Index

size()

Returns the current size of this component.

Error Index

toString()

Returns the String representation of this Component's values.

Error Index

update(Graphics)

Updates the component.

Error Index

validate()

Validates a component.

Error Index

Error Index

getParent

public Container getParent()

Gets the parent of the component.

Error Index

getPeer

public ComponentPeer getPeer()

Gets the peer of the component.

Error Index

getToolkit

```
public Toolkit getToolkit()
```

Gets the toolkit of the component. This toolkit is used to create the peer for this component. Note that the Frame which contains a Component controls which toolkit is used so if the Component has not yet been added to a Frame or if it is later moved to a different Frame, the toolkit it uses may change.

Error Index isValid

```
public boolean isValid()
```

Checks if this Component is valid. Components are invalidated when they are first shown on the screen.

See Also:

validate, invalidate

Error Index isVisible

```
public boolean isVisible()
```

Checks if this Component is visible. Components are initially visible (with the exception of top level components such as Frame).

See Also:

show, hide

Error Index isShowing

```
public boolean isShowing()
```

Checks if this Component is showing on screen. This means that the component must be visible, and it must be in a container that is visible and showing.

See Also:

show, hide

Error Index isEnabled

```
public boolean isEnabled()
```

Checks if this Component is enabled. Components are initially enabled.

See Also:

enable, disable

Error Index location

public Point location()

Returns the current location of this component. The location will be in the parent's coordinate space.

See Also:

move

Error Index size

public Dimension size()

Returns the current size of this component.

See Also:

resize

Error Index bounds

public Rectangle bounds()

Returns the current bounds of this component.

See Also:

reshape

Error Index enable

public synchronized void enable()

Enables a component.

See Also:

isEnabled, disable

Error Index enable

public void enable(boolean cond)

Conditionally enables a component.

Parameters:

cond - if true, enables component; disables otherwise.

See Also:

enable, disable

Error Index disable

public synchronized void disable()

Disables a component.

See Also:

isEnabled, enable

Error Index show

public synchronized void show()

Shows the component.

See Also:

isVisible, hide

Error Index show

public void show(boolean cond)

Conditionally shows the component.

Parameters:

cond - if true, it shows the component; hides otherwise.

See Also:

show, hide

Error Index hide

public synchronized void hide()

Hides the component.

See Also:

[isVisible](#), [hide](#)

Error Index getForeground

public [Color](#) getForeground()

Gets the foreground color. If the component does not have a foreground color, the foreground color of its parent is returned.

See Also:

[setForeground](#)

Error Index setForeground

public synchronized void setForeground([Color](#) c)

Sets the foreground color.

Parameters:

c - the Color

See Also:

[getForeground](#)

Error Index getBackground

public [Color](#) getBackground()

Gets the background color. If the component does not have a background color, the background color of its parent is returned.

See Also:

[setBackground](#)

Error Index setBackground

public synchronized void setBackground(Color c)

Sets the background color.

Parameters:

c - the Color

See Also:

getBackground

Error Index getFont

public Font getFont()

Gets the font of the component. If the component does not have a font, the font of its parent is returned.

See Also:

setFont

Error Index setFont

public synchronized void setFont(Font f)

Sets the font of the component.

Parameters:

f - the font

See Also:

getFont

Error Index getColorModel

public synchronized ColorModel getColorModel()

Gets the ColorModel used to display the component on the output device.

See Also:

ColorModel

Error Index **move**

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

Moves the Component to a new location. The x and y coordinates are in the parent's coordinate space.

Parameters:

x - the x coordinate

y - the y coordinate

See Also:

[location](#), [reshape](#)

Error Index **resize**

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

Resizes the Component to the specified width and height.

Parameters:

width - the width of the component

height - the height of the component

See Also:

[size](#), [reshape](#)

Error Index **resize**

```
public void resize(Dimension d)
```

Resizes the Component to the specified dimension.

Parameters:

d - the component dimension

See Also:

[size](#), [reshape](#)

Error Index **reshape**

```
public synchronized void reshape(int x,  
                                int y,  
                                int width,
```


int height)

Reshapes the Component to the specified bounding box.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the component
height - the height of the component

See Also:

[bounds](#), [move](#), [resize](#)

Error Index preferredSize

public [Dimension](#) preferredSize()

Returns the preferred size of this component.

See Also:

[minimumSize](#), [LayoutManager](#)

Error Index minimumSize

public [Dimension](#) minimumSize()

Returns the minimum size of this component.

See Also:

[preferredSize](#), [LayoutManager](#)

Error Index layout

public void layout()

Lays out the component. This is usually called when the component is validated.

See Also:

[validate](#), [LayoutManager](#)

Error Index validate

public void validate()

Validates a component.

See Also:

[invalidate](#), [layout](#), [LayoutManager](#)

Error Index

invalidate

public void invalidate()

Invalidates a component.

See Also:

[validate](#), [layout](#), [LayoutManager](#)

Error Index

getGraphics

public [Graphics](#) getGraphics()

Gets a Graphics context for this component. This method will return null if the component is currently not on the screen.

See Also:

[paint](#)

Error Index

getFontMetrics

public [FontMetrics](#) getFontMetrics([Font](#) font)

Gets the font metrics for this component. This will return null if the component is currently not on the screen.

Parameters:

font - the font

See Also:

[getFont](#)

Error Index

paint

public void paint([Graphics](#) g)

Paints the component.

Parameters:

g - the specified Graphics window

See Also:

update

Error Index update

```
public void update(Graphics g)
```

Updates the component. This method is called in response to a call to repaint. You can assume that the background is not cleared.

Parameters:

g - the specified Graphics window

See Also:

paint, repaint

Error Index paintAll

```
public void paintAll(Graphics g)
```

Paints the component and its subcomponents.

Parameters:

g - the specified Graphics window

See Also:

paint

Error Index repaint

```
public void repaint()
```

Repaints the component. This will result in a call to update as soon as possible.

See Also:

paint

Error Index repaint

```
public void repaint(long tm)
```

Repaints the component. This will result in a call to update within *tm* milliseconds.

Parameters:

tm - maximum time in milliseconds before update

See Also:

[paint](#)

Error Index [repaint](#)

```
public void repaint(int x,  
    int y,  
    int width,  
    int height)
```

Repaints part of the component. This will result in a call to update as soon as possible.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width

height - the height

See Also:

[repaint](#)

Error Index [repaint](#)

```
public void repaint(long tm,  
    int x,  
    int y,  
    int width,  
    int height)
```

Repaints part of the component. This will result in a call to update within *tm* milliseconds.

Parameters:

tm - maximum time in milliseconds before update

x - the x coordinate

y - the y coordinate

width - the width

height - the height

See Also:

[repaint](#)

Error Index print

```
public void print(Graphics g)
```

Prints this component. The default implementation of this method calls paint.

Parameters:

g - the specified Graphics window

See Also:

paint

Error Index printAll

```
public void printAll(Graphics g)
```

Prints the component and its subcomponents.

Parameters:

g - the specified Graphics window

See Also:

print

Error Index imageUpdate

```
public boolean imageUpdate(Image img,  
    int flags,  
    int x,  
    int y,  
    int w,  
    int h)
```

Repaints the component when the image has changed.

Returns:

true if image has changed; false otherwise.

Error Index createImage

```
public Image createImage(ImageProducer producer)
```

Creates an image from the specified image producer.

Parameters:

producer - the image producer

Error Index **createImage**

```
public Image createImage(int width,  
                           int height)
```

Creates an off-screen drawable Image to be used for double buffering.

Parameters:

width - the specified width
height - the specified height

Error Index **prepareImage**

```
public boolean prepareImage(Image image,  
                             ImageObserver observer)
```

Prepares an image for rendering on this Component. The image data is downloaded asynchronously in another thread and the appropriate screen representation of the image is generated.

Parameters:

image - the Image to prepare a screen representation for
observer - the ImageObserver object to be notified as the image is being prepared

Returns:

true if the image has already been fully prepared

See Also:

ImageObserver

Error Index **prepareImage**

```
public boolean prepareImage(Image image,  
                             int width,  
                             int height,  
                             ImageObserver observer)
```

Prepares an image for rendering on this Component at the specified width and height. The image data is downloaded asynchronously in another thread and an appropriately scaled screen representation of the image is generated.

Parameters:

image - the Image to prepare a screen representation for
width - the width of the desired screen representation
height - the height of the desired screen representation
observer - the ImageObserver object to be notified as the image is being prepared

Returns:

true if the image has already been fully prepared

See Also:

[ImageObserver](#)

Error Index **checkImage**

```
public int checkImage(Image image,
                     ImageObserver observer)
```

Returns the status of the construction of a screen representation of the specified image. This method does not cause the image to begin loading. Use the `prepareImage` method to force the loading of an image.

Parameters:

image - the Image to check the status of

observer - the ImageObserver object to be notified as the image is being prepared

Returns:

the boolean OR of the ImageObserver flags for the data that is currently available

See Also:

[ImageObserver](#), [prepareImage](#)

Error Index **checkImage**

```
public int checkImage(Image image,
                     int width,
                     int height,
                     ImageObserver observer)
```

Returns the status of the construction of a scaled screen representation of the specified image. This method does not cause the image to begin loading, use the `prepareImage` method to force the loading of an image.

Parameters:

image - the Image to check the status of

width - the width of the scaled version to check the status of

height - the height of the scaled version to check the status of

observer - the ImageObserver object to be notified as the image is being prepared

Returns:

the boolean OR of the ImageObserver flags for the data that is currently available

See Also:

[ImageObserver](#), [prepareImage](#)

Error Index **inside**

```
public synchronized boolean inside(int x,  
                                   int y)
```

Checks whether a specified x,y location is "inside" this Component. By default, x and y are inside an Component if they fall within the bounding box of that Component.

Parameters:

x - the x coordinate

y - the y coordinate

See Also:

locate

Error Index locate

```
public Component locate(int x,  
                        int y)
```

Returns the component or subcomponent that contains the x,y location.

Parameters:

x - the x coordinate

y - the y coordinate

See Also:

inside

Error Index deliverEvent

```
public void deliverEvent(Event e)
```

Delivers an event to this component or one of its sub components.

Parameters:

e - the event

See Also:

handleEvent, postEvent

Error Index postEvent

```
public boolean postEvent(Event e)
```

Posts an event to this component. This will result in a call to handleEvent. If handleEvent returns false the event is passed on to the parent of this component.

Parameters:

e - the event

See Also:

[handleEvent](#), [deliverEvent](#)

Error Index [handleEvent](#)

```
public boolean handleEvent(Event evt)
```

Handles the event. Returns true if the event is handled and should not be passed to the parent of this component. The default event handler calls some helper methods to make life easier on the programmer.

Parameters:

evt - the event

See Also:

[mouseEnter](#), [mouseExit](#), [mouseMove](#), [mouseDown](#), [mouseDrag](#), [mouseUp](#), [keyDown](#), [action](#)

Error Index [mouseDown](#)

```
public boolean mouseDown(Event evt,  
    int x,  
    int y)
```

Called if the mouse is down.

Parameters:

evt - the event

x - the x coordinate

y - the y coordinate

See Also:

[handleEvent](#)

Error Index [mouseDrag](#)

```
public boolean mouseDrag(Event evt,  
    int x,  
    int y)
```

Called if the mouse is dragged (the mouse button is down).

Parameters:

evt - the event

x - the x coordinate

y - the y coordinate

See Also:

[handleEvent](#)

Error Index

mouseUp

```
public boolean mouseUp(Event evt,  
    int x,  
    int y)
```

Called if the mouse is up.

Parameters:

evt - the event
x - the x coordinate
y - the y coordinate

See Also:

[handleEvent](#)

Error Index

mouseMove

```
public boolean mouseMove(Event evt,  
    int x,  
    int y)
```

Called if the mouse moves (the mouse button is up).

Parameters:

evt - the event
x - the x coordinate
y - the y coordinate

See Also:

[handleEvent](#)

Error Index

mouseEnter

```
public boolean mouseEnter(Event evt,  
    int x,  
    int y)
```

Called when the mouse enters the component.

Parameters:

evt - the event
x - the x coordinate
y - the y coordinate

See Also:

[handleEvent](#)

Error Index

mouseExit

```
public boolean mouseExit(Event evt,  
    int x,  
    int y)
```

Called when the mouse exits the component.

Parameters:

evt - the event
x - the x coordinate
y - the y coordinate

See Also:

[handleEvent](#)

Error Index

keyDown

```
public boolean keyDown(Event evt,  
    int key)
```

Called if a character is pressed.

Parameters:

evt - the event
key - the key that's pressed

See Also:

[handleEvent](#)

Error Index

keyUp

```
public boolean keyUp(Event evt,  
    int key)
```

Called if a character is released.

Parameters:

evt - the event
key - the key that's released

See Also:

[handleEvent](#)

Error Index **action**

```
public boolean action(Event evt,  
                     Object what)
```

Called if an action occurs in the Component.

Parameters:

evt - the event

what - the action that's occurring

See Also:

handleEvent

Error Index **addNotify**

```
public void addNotify()
```

Notifies the Component to create a peer.

See Also:

getPeer, removeNotify

Error Index **removeNotify**

```
public synchronized void removeNotify()
```

Notifies the Component to destroy the peer.

See Also:

getPeer, addNotify

Error Index **gotFocus**

```
public boolean gotFocus(Event evt,  
                       Object what)
```

Indicates that this component has received the input focus.

See Also:

requestFocus, lostFocus

Error Index **lostFocus**

```
public boolean lostFocus(Event evt,  
                        Object what)
```

Indicates that this component has lost the input focus.

See Also:

requestFocus, gotFocus

Error Index **requestFocus**

```
public void requestFocus()
```

Requests the input focus. The gotFocus() method will be called if this method is successful.

See Also:

gotFocus

Error Index **nextFocus**

```
public void nextFocus()
```

Moves the focus to the next component.

See Also:

requestFocus, gotFocus

Error Index **paramString**

```
protected String paramString()
```

Returns the parameter String of this Component.

Error Index **toString**

```
public String toString()
```

Returns the String representation of this Component's values.

Overrides:

toString in class Object

Error Index list

```
public void list()
```

Prints a listing to a print stream.

Error Index list

```
public void list(PrintStream out)
```

Prints a listing to the specified print out stream.

Parameters:

out - the Stream name

Error Index list

```
public void list(PrintStream out,  
                int indent)
```

Prints out a list, starting at the specified indentation, to the specified print stream.

Parameters:

out - the Stream name

indent - the start of the list

Class java.awt.Container

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Container

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Container
```

```
public class Container
    extends Component
```

A generic Abstract Window Toolkit(AWT) container object is a component that can contain other AWT components.

Error Index

Error Index

add(Component)

Adds the specified component to this container.

Error Index

add(Component, int)

Adds the specified component to this container at the given position.

Error Index

add(String, Component)

Adds the specified component to this container.

Error Index

addNotify()

Notifies the container to create a peer.

Error Index

countComponents()

Returns the number of components in this panel.

Error Index

deliverEvent(Event)

Delivers an event.

Error Index

getComponent(int)

Gets the nth component in this container.

Error Index

getComponents()

Gets all the components in this container.

Error Index

getLayout()

Gets the layout manager for this container.

Error Index

insets()

Returns the insets of the container.

Error Index

layout()

Does a layout on this Container.

Error Index

list(PrintStream, int)

Prints out a list, starting at the specified indentation, to the specified out stream.

Error Index

locate(int, int)

Locates the component that contains the x,y position.

Error Index

minimumSize()

Returns the minimum size of this container.

Error Index

paintComponents(Graphics)

Paints the components in this container.

Error Index

paramString()

Returns the parameter String of this Container.

Error Index

preferredSize()

Returns the preferred size of this container.

Error Index

printComponents(Graphics)

Prints the components in this container.

Error Index

remove(Component)

Removes the specified component from this container.

Error Index

removeAll()

Removes all the components from this container.

Error Index

removeNotify()

Notifies the container to remove its peer.

Error Index

setLayout(LayoutManager)

Sets the layout manager for this container.

Error Index

validate()

Validates this Container and all of the components contained within it.

Error Index

Error Index

countComponents

```
public int countComponents()
```

Returns the number of components in this panel.

See Also:

getComponent

Error Index

getComponent

```
public synchronized Component getComponent(int n)
```

Gets the nth component in this container.

Parameters:

n - the number of the component to get

Throws: ArrayIndexOutOfBoundsException

If the nth value does not exist.

Error Index

getComponents

```
public synchronized Component[] getComponents()
```

Gets all the components in this container.

Error Index

insets

```
public Insets insets()
```

Returns the insets of the container. The insets indicate the size of the border of the container. A Frame, for example, will have a top inset that corresponds to the height of the Frame's title bar.

See Also:

LayoutManager

Error Index **add**

```
public Component add(Component comp)
```

Adds the specified component to this container.

Parameters:

comp - the component to be added

Error Index **add**

```
public synchronized Component add(Component comp,  
int pos)
```

Adds the specified component to this container at the given position.

Parameters:

comp - the component to be added

pos - the position at which to insert the component. -1 means insert at the end.

See Also:

remove

Error Index **add**

```
public synchronized Component add(String name,  
Component comp)
```

Adds the specified component to this container. The component is also added to the layout manager of this container using the name specified .

Parameters:

name - the component name

comp - the component to be added

See Also:

remove, LayoutManager

Error Index **remove**

```
public synchronized void remove(Component comp)
```

Removes the specified component from this container.

Parameters:

comp - the component to be removed

See Also:

add

Error Index removeAll

public synchronized void removeAll()

Removes all the components from this container.

See Also:

add, remove

Error Index getLayout

public LayoutManager getLayout()

Gets the layout manager for this container.

See Also:

layout, setLayout

Error Index setLayout

public void setLayout(LayoutManager mgr)

Sets the layout manager for this container.

Parameters:

mgr - the specified layout manager

See Also:

layout, getLayout

Error Index layout

public synchronized void layout()

Does a layout on this Container.

Overrides:

layout in class Component

See Also:

setLayout

Error Index validate

public synchronized void validate()

Validates this Container and all of the components contained within it.

Overrides:

validate in class Component

See Also:

validate, invalidate

Error Index preferredSize

public synchronized Dimension preferredSize()

Returns the preferred size of this container.

Overrides:

preferredSize in class Component

See Also:

minimumSize

Error Index minimumSize

public synchronized Dimension minimumSize()

Returns the minimum size of this container.

Overrides:

minimumSize in class Component

See Also:

preferredSize

Error Index paintComponents

public void paintComponents(Graphics g)

Paints the components in this container.

Parameters:

g - the specified Graphics window

See Also:

paint, paintAll

Error Index printComponents

public void printComponents(Graphics g)

Prints the components in this container.

Parameters:

g - the specified Graphics window

See Also:

print, printAll

Error Index deliverEvent

public void deliverEvent(Event e)

Delivers an event. The appropriate component is located and the event is delivered to it.

Parameters:

e - the event

Overrides:

deliverEvent in class Component

See Also:

handleEvent, postEvent

Error Index locate

public Component locate(int x,
int y)

Locates the component that contains the x,y position.

Parameters:

x - the x coordinate

y - the y coordinate

Returns:

null if the component is not within the x and y coordinates; returns the component otherwise.

Overrides:

locate in class Component

See Also:

inside

Error Index **addNotify**

public synchronized void addNotify()

Notifies the container to create a peer. It will also notify the components contained in this container.

Overrides:

addNotify in class Component

See Also:

removeNotify

Error Index **removeNotify**

public synchronized void removeNotify()

Notifies the container to remove its peer. It will also notify the components contained in this container.

Overrides:

removeNotify in class Component

See Also:

addNotify

Error Index **paramString**

protected String paramString()

Returns the parameter String of this Container.

Overrides:

paramString in class Component

Error Index **list**

public void list(PrintStream out,
int indent)

Prints out a list, starting at the specified indentation, to the specified out stream.

Parameters:

out - the Stream name

indent - the start of the list

Overrides:

list in class Component

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Dialog

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Dialog

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Container
            |
            +----java.awt.Window
                  |
                  +----java.awt.Dialog
```

```
public class Dialog
    extends Window
```

A class that produces a dialog - a window that takes input from the user. The default layout for a dialog is BorderLayout.

Error Index

Error Index

[Dialog](#)(Frame, boolean)

Constructs an initially invisible Dialog.

Error Index

[Dialog](#)(Frame, String, boolean)

Constructs an initially invisible Dialog with a title.

Error Index

Error Index

[addNotify](#)()

Creates the frame's peer.

Error Index

[getTitle](#)()

Gets the title of the Dialog.

Error Index

isModal()

Returns true if the Dialog is modal.

Error Index

isResizable()

Returns true if the user can resize the frame.

Error Index

paramString()

Returns the parameter String of this Dialog.

Error Index

setResizable(boolean)

Sets the resizable flag.

Error Index

setTitle(String)

Sets the title of the Dialog.

Error Index

Error Index

Dialog

```
public Dialog(Frame parent,  
             boolean modal)
```

Constructs an initially invisible Dialog. A modal Dialog grabs all the input from the user.

Parameters:

parent - the owner of the dialog

modal - if true, dialog blocks input to other windows when shown

See Also:

resize, show

Error Index

Dialog

```
public Dialog(Frame parent,  
             String title,  
             boolean modal)
```

Constructs an initially invisible Dialog with a title. A modal Dialog grabs all the input from the user.

Parameters:

parent - the owner of the dialog

title - the title of the dialog

modal - if true, dialog blocks input to other windows when shown

See Also:

resize, show

Error Index

Error Index addNotify

public synchronized void addNotify()

Creates the frame's peer. The peer allows us to change the appearance of the frame without changing its functionality.

Overrides:

addNotify in class Window

Error Index isModal

public boolean isModal()

Returns true if the Dialog is modal. A modal Dialog grabs all the input from the user.

Error Index getTitle

public String getTitle()

Gets the title of the Dialog.

See Also:

setTitle

Error Index setTitle

public void setTitle(String title)

Sets the title of the Dialog.

Parameters:

title - the new title being given to the Dialog

See Also:

getTitle

Error Index isResizable

public boolean isResizable()

Returns true if the user can resize the frame.

Error Index setResizable

public void setResizable(boolean resizable)

Sets the resizable flag.

Parameters:

resizable - true if resizable; false otherwise

Error Index paramString

protected String paramString()

Returns the parameter String of this Dialog.

Overrides:

paramString in class Container

Class java.awt.Dimension

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Dimension

[java.lang.Object](#)
|
+----java.awt.Dimension

public class **Dimension**
extends [Object](#)

A class to encapsulate a width and a height Dimension.

Error Index

- [height](#)
The height dimension.
- [width](#)
The width dimension.

Error Index

Error Index

[Dimension](#)()

Constructs a Dimension with a 0 width and 0 height.

Error Index

[Dimension](#)([Dimension](#))

Constructs a Dimension and initializes it to the specified value.

Error Index

[Dimension](#)(int, int)

Constructs a Dimension and initializes it to the specified width and specified height.

Error Index

Error Index

toString()

Returns the String representation of this Dimension's values.

Variables

Error Index

width

public int width

The width dimension.

Error Index

height

public int height

The height dimension.

Error Index

Error Index

Dimension

public Dimension()

Constructs a Dimension with a 0 width and 0 height.

Error Index

Dimension

public Dimension(Dimension d)

Constructs a Dimension and initializes it to the specified value.

Parameters:

d - the specified dimension for the width and height values

Error Index

Dimension

```
public Dimension(int width,  
                 int height)
```

Constructs a Dimension and initializes it to the specified width and specified height.

Parameters:

width - the specified width dimension
height - the specified height dimension

Error Index

Error Index toString

```
public String toString()
```

Returns the String representation of this Dimension's values.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Event

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Event

[java.lang.Object](#)

|
+----java.awt.Event

public class **Event**
extends [Object](#)

Event is a platform-independent class that encapsulates events from the local Graphical User Interface(GUI) platform.

Error Index

Error Index

An action event.

ACTION_EVENT

Error Index

The alt modifier constant.

ALT_MASK

Error Index

The control modifier constant.

CTRL_MASK

Error Index

The down arrow key.

DOWN

Error Index

The end key.

END

Error Index

The F1 function key.

F1

Error Index

The F10 function key.

F10

Error Index

The F11 function key.

F11

Error Index

The F12 function key.

F12

Error Index

The F2 function key.

F2

Error Index

The F3 function key.

F3

Error Index

The F4 function key.

F4

Error Index

The F5 function key.

F5

Error Index

The F6 function key.

F6

Error Index

The F7 function key.

F7

Error Index

The F8 function key.

F8

Error Index

The F9 function key.

F9

Error Index

A component gained the focus.

GOT_FOCUS

Error Index

The home key.

HOME

Error Index

The key action keyboard event.

KEY_ACTION

Error Index

The key action keyboard event.

KEY_ACTION_RELEASE

Error Index

The key press keyboard event.

KEY_PRESS

Error Index

KEY_RELEASE

The key release keyboard event.

Error Index

LEFT

The left arrow key.

Error Index

LIST_DESELECT

Error Index

LIST_SELECT

Error Index

LOAD_FILE

A file loading event.

Error Index

LOST_FOCUS

A component lost the focus.

Error Index

META_MASK

The meta modifier constant.

Error Index

MOUSE_DOWN

The mouse down event.

Error Index

MOUSE_DRAG

The mouse drag event.

Error Index

MOUSE_ENTER

The mouse enter event.

Error Index

MOUSE_EXIT

The mouse exit event.

Error Index

MOUSE_MOVE

The mouse move event.

Error Index

MOUSE_UP

The mouse up event.

Error Index

PGDN

The page down key.

Error Index

PGUP

The page up key.

Error Index

RIGHT

The right arrow key.

Error Index

SAVE_FILE

A file saving event.

Error Index

SCROLL_ABSOLUTE

The absolute scroll event.

Error Index

SCROLL_LINE_DOWN

The line down scroll event.

Error Index

SCROLL_LINE_UP

The line up scroll event.

Error Index

SCROLL_PAGE_DOWN

The page down scroll event.

Error Index

SCROLL_PAGE_UP

The page up scroll event.

Error Index

SHIFT_MASK

The shift modifier constant.

Error Index

UP

The up arrow key.

Error Index

WINDOW_DEICONIFY

The de-iconify window event.

Error Index

WINDOW_DESTROY

The destroy window event.

Error Index

WINDOW_EXPOSE

The expose window event.

Error Index

WINDOW_ICONIFY

The iconify window event.

Error Index

WINDOW_MOVED

The move window event.

- arg
An arbitrary argument.
- clickCount
The number of consecutive clicks.
- evt
The next event.
- id

The type of this event.

- key

The key that was pressed in a keyboard event.

Error Index

modifiers

The state of the modifier keys.

Error Index

target

The target component.

Error Index

when

The time stamp.

Error Index

x

The x coordinate of the event.

Error Index

y

The y coordinate of the event.

Error Index

Error Index

Event(Object, long, int, int, int, int, int, Object)

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and argument.

Error Index

Event(Object, long, int, int, int, int, int)

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and an argument set to null.

Error Index

Event(Object, int, Object)

Constructs an event with the specified target component, event type, and argument.

Error Index

Error Index

controlDown()

Checks if the control key is down.

Error Index

metaDown()

Checks if the meta key is down.

Error Index

paramString()

Returns the parameter String of this Event.

Error Index

shiftDown()

Checks if the shift key is down.

Error Index

toString()

Returns the String representation of this Event's values.

Error Index

translate(int, int)

Translates an event relative to the given component.

Error Index

Error Index

SHIFT_MASK

public final static int SHIFT_MASK

The shift modifier constant.

Error Index

CTRL_MASK

public final static int CTRL_MASK

The control modifier constant.

Error Index

META_MASK

public final static int META_MASK

The meta modifier constant.

Error Index

ALT_MASK

public final static int ALT_MASK

The alt modifier constant.

Error Index

HOME

```
public final static int HOME
```

The home key.

Error Index END

```
public final static int END
```

The end key.

Error Index PGUP

```
public final static int PGUP
```

The page up key.

Error Index PGDN

```
public final static int PGDN
```

The page down key.

Error Index UP

```
public final static int UP
```

The up arrow key.

Error Index DOWN

```
public final static int DOWN
```

The down arrow key.

Error Index LEFT

```
public final static int LEFT
```

The left arrow key.

Error Index RIGHT

public final static int RIGHT

The right arrow key.

Error Index F1

public final static int F1

The F1 function key.

Error Index F2

public final static int F2

The F2 function key.

Error Index F3

public final static int F3

The F3 function key.

Error Index F4

public final static int F4

The F4 function key.

Error Index F5

public final static int F5

The F5 function key.

Error Index F6

public final static int F6

The F6 function key.

Error Index F7

public final static int F7

The F7 function key.

Error Index F8

public final static int F8

The F8 function key.

Error Index F9

public final static int F9

The F9 function key.

Error Index F10

public final static int F10

The F10 function key.

Error Index F11

public final static int F11

The F11 function key.

Error Index F12

public final static int F12

The F12 function key.

Error Index WINDOW_DESTROY

```
public final static int WINDOW_DESTROY
```

The destroy window event.

Error Index WINDOW_EXPOSE

```
public final static int WINDOW_EXPOSE
```

The expose window event.

Error Index WINDOW_ICONIFY

```
public final static int WINDOW_ICONIFY
```

The iconify window event.

Error Index WINDOW_DEICONIFY

```
public final static int WINDOW_DEICONIFY
```

The de-iconify window event.

Error Index WINDOW_MOVED

```
public final static int WINDOW_MOVED
```

The move window event.

Error Index KEY_PRESS

```
public final static int KEY_PRESS
```

The key press keyboard event.

Error Index KEY_RELEASE

```
public final static int KEY_RELEASE
```

The key release keyboard event.

Error Index KEY_ACTION

```
public final static int KEY_ACTION
```

The key action keyboard event.

Error Index KEY_ACTION_RELEASE

```
public final static int KEY_ACTION_RELEASE
```

The key action keyboard event.

Error Index MOUSE_DOWN

```
public final static int MOUSE_DOWN
```

The mouse down event.

Error Index MOUSE_UP

```
public final static int MOUSE_UP
```

The mouse up event.

Error Index MOUSE_MOVE

```
public final static int MOUSE_MOVE
```

The mouse move event.

Error Index MOUSE_ENTER

public final static int MOUSE_ENTER

The mouse enter event.

Error Index MOUSE_EXIT

public final static int MOUSE_EXIT

The mouse exit event.

Error Index MOUSE_DRAG

public final static int MOUSE_DRAG

The mouse drag event.

Error Index SCROLL_LINE_UP

public final static int SCROLL_LINE_UP

The line up scroll event.

Error Index SCROLL_LINE_DOWN

public final static int SCROLL_LINE_DOWN

The line down scroll event.

Error Index SCROLL_PAGE_UP

public final static int SCROLL_PAGE_UP

The page up scroll event.

Error Index SCROLL_PAGE_DOWN

public final static int SCROLL_PAGE_DOWN

The page down scroll event.

Error Index

SCROLL_ABSOLUTE

```
public final static int SCROLL_ABSOLUTE
```

The absolute scroll event.

Error Index

LIST_SELECT

```
public final static int LIST_SELECT
```

Error Index

LIST_DESELECT

```
public final static int LIST_DESELECT
```

Error Index

ACTION_EVENT

```
public final static int ACTION_EVENT
```

An action event.

Error Index

LOAD_FILE

```
public final static int LOAD_FILE
```

A file loading event.

Error Index

SAVE_FILE

```
public final static int SAVE_FILE
```

A file saving event.

Error Index

GOT_FOCUS

```
public final static int GOT_FOCUS
```

A component gained the focus.

Error Index _{LOST_FOCUS}

public final static int LOST_FOCUS

A component lost the focus.

Error Index _{target}

public Object target

The target component.

Error Index _{when}

public long when

The time stamp.

Error Index _{id}

public int id

The type of this event.

Error Index _x

public int x

The x coordinate of the event.

Error Index _y

public int y

The y coordinate of the event.

Error Index **key**

public int key

The key that was pressed in a keyboard event.

Error Index **modifiers**

public int modifiers

The state of the modifier keys.

Error Index **clickCount**

public int clickCount

The number of consecutive clicks. This field is relevant only for MOUSE_DOWN events. If the field isn't set it will be 0. Otherwise, it will be 1 for single-clicks, 2 for double-clicks, and so on.

Error Index **arg**

public Object arg

An arbitrary argument.

Error Index **evt**

public Event evt

The next event. Used when putting events into a linked list.

Error Index

Error Index **Event**

```
public Event(Object target,  
            long when,  
            int id,
```

```
int x,  
int y,  
int key,  
int modifiers,  
Object arg)
```

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and argument.

Parameters:

target - the target component
when - the time stamp
id - the event type
x - the x coordinate
y - the y coordinate
key - the key pressed in a keyboard event
modifiers - the state of the modifier keys
arg - the specified argument

Error Index Event

```
public Event(Object target,  
            long when,  
            int id,  
            int x,  
            int y,  
            int key,  
            int modifiers)
```

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and an argument set to null.

Parameters:

target - the target component
when - the time stamp
id - the event type
x - the x coordinate
y - the y coordinate
key - the key pressed in a keyboard event
modifiers - the state of the modifier keys

Error Index Event

```
public Event(Object target,  
            int id,  
            Object arg)
```

Constructs an event with the specified target component, event type, and argument.

Parameters:

target - the target component

id - the event type

arg - the specified argument

Error Index

Error Index

translate

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

Translates an event relative to the given component. This involves at a minimum translating the coordinates so they make sense within the given component. It may also involve translating a region in the case of an expose event.

Parameters:

x - the x coordinate

y - the y coordinate

Error Index

shiftDown

```
public boolean shiftDown()
```

Checks if the shift key is down.

See Also:

[modifiers](#), [controlDown](#), [metaDown](#)

Error Index

controlDown

```
public boolean controlDown()
```

Checks if the control key is down.

See Also:

[modifiers](#), [shiftDown](#), [metaDown](#)

Error Index metaDown

public boolean metaDown()

Checks if the meta key is down.

See Also:

[modifiers](#), [shiftDown](#), [controlDown](#)

Error Index paramString

protected [String](#) paramString()

Returns the parameter String of this Event.

Error Index toString

public [String](#) toString()

Returns the String representation of this Event's values.

Overrides:

[toString](#) in class [Object](#)

Class java.awt.FileDialog

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.FileDialog

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Container
            |
            +----java.awt.Window
                  |
                  +----java.awt.Dialog
                        |
                        +----java.awt.FileDialog
```

```
public class FileDialog
    extends Dialog
```

The File Dialog class displays a file selection dialog. It is a modal dialog and will block the calling thread when the show method is called to display it, until the user has chosen a file.

See Also:

[show](#)

Error Index

Error Index

LOAD

The file load variable.

Error Index

SAVE

The file save variable.

Error Index

Error Index

FileDialog(Frame, String)

Creates a file dialog for loading a file.

Error Index

FileDialog(Frame, String, int)

Creates a file dialog with the specified title and mode.

Error Index

Error Index

addNotify()

Creates the frame's peer.

Error Index

getDirectory()

Gets the directory of the Dialog.

Error Index

getFile()

Gets the file of the Dialog.

Error Index

getFilenameFilter()

Gets the filter.

Error Index

getMode()

Gets the mode of the file dialog.

Error Index

paramString()

Returns the parameter String of this file dialog.

Error Index

setDirectory(String)

Set the directory of the Dialog to the specified directory.

Error Index

setFile(String)

Sets the file for this dialog to the specified file.

Error Index

setFilenameFilter(FilenameFilter)

Sets the filter for this dialog to the specified filter.

Error Index

Error Index

LOAD

public final static int LOAD

The file load variable.

Error Index SAVE

```
public final static int SAVE
```

The file save variable.

Error Index

Error Index FileDialog

```
public FileDialog(Frame parent,  
                 String title)
```

Creates a file dialog for loading a file.

Parameters:

parent - the owner of the dialog
title - the title of the Dialog

Error Index FileDialog

```
public FileDialog(Frame parent,  
                 String title,  
                 int mode)
```

Creates a file dialog with the specified title and mode.

Parameters:

parent - the owner of the dialog
title - the title of the Dialog
mode - the mode of the Dialog

Error Index

Error Index addNotify

```
public synchronized void addNotify()
```

Creates the frame's peer. The peer allows us to change the look of the file dialog without changing its functionality.

Overrides:

addNotify in class Dialog

Error Index

getMode

```
public int getMode()
```

Gets the mode of the file dialog.

Error Index

getDirectory

```
public String getDirectory()
```

Gets the directory of the Dialog.

Error Index

setDirectory

```
public void setDirectory(String dir)
```

Set the directory of the Dialog to the specified directory.

Parameters:

dir - the specific directory

Error Index

getFile

```
public String getFile()
```

Gets the file of the Dialog.

Error Index

setFile

```
public void setFile(String file)
```

Sets the file for this dialog to the specified file. This will become the default file if set before the dialog is shown.

Parameters:

file - the file being set

Error Index `getFilenameFilter`

```
public FilenameFilter getFilenameFilter()
```

Gets the filter.

Error Index `setFilenameFilter`

```
public void setFilenameFilter(FilenameFilter filter)
```

Sets the filter for this dialog to the specified filter.

Parameters:

filter - the specified filter

Error Index `paramString`

```
protected String paramString()
```

Returns the parameter String of this file dialog. Parameter String.

Overrides:

paramString in class Dialog

Class java.awt.FlowLayout

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.FlowLayout

[java.lang.Object](#)

|

+----java.awt.FlowLayout

```
public class FlowLayout
  extends Object
  implements LayoutManager
```

Flow layout is used to layout buttons in a panel. It will arrange buttons left to right until no more buttons fit on the same line. Each line is centered.

Error Index

Error Index

CENTER

The right alignment variable.

Error Index

LEFT

The left alignment variable.

Error Index

RIGHT

The right alignment variable.

Error Index

Error Index

FlowLayout()

Constructs a new Flow Layout with a centered alignment.

Error Index

FlowLayout(int)

Constructs a new Flow Layout with the specified alignment.

Error Index

FlowLayout(int, int, int)

Constructs a new Flow Layout with the specified alignment and gap values.

Error Index

Error Index

addLayoutComponent(String, Component)

Adds the specified component to the layout.

Error Index

layoutContainer(Container)

Lays out the container.

Error Index

minimumLayoutSize(Container)

Returns the minimum dimensions needed to layout the components contained in the specified target container.

Error Index

preferredLayoutSize(Container)

Returns the preferred dimensions for this layout given the components in the specified target container.

Error Index

removeLayoutComponent(Component)

Removes the specified component from the layout.

Error Index

toString()

Returns the String representation of this FlowLayout's values.

Error Index

Error Index

LEFT

public final static int LEFT

The left alignment variable.

Error Index

CENTER

public final static int CENTER

The right alignment variable.

Error Index RIGHT

```
public final static int RIGHT
```

The right alignment variable.

Error Index

Error Index FlowLayout

```
public FlowLayout()
```

Constructs a new Flow Layout with a centered alignment.

Error Index FlowLayout

```
public FlowLayout(int align)
```

Constructs a new Flow Layout with the specified alignment.

Parameters:

align - the alignment value

Error Index FlowLayout

```
public FlowLayout(int align,  
                  int hgap,  
                  int vgap)
```

Constructs a new Flow Layout with the specified alignment and gap values.

Parameters:

align - the alignment value

hgap - the horizontal gap variable

vgap - the vertical gap variable

Error Index

Error Index

addLayoutComponent

```
public void addLayoutComponent(String name,  
                               Component comp)
```

Adds the specified component to the layout. Not used by this class.

Parameters:

name - the name of the component
comp - the the component to be added

Error Index

removeLayoutComponent

```
public void removeLayoutComponent(Component comp)
```

Removes the specified component from the layout. Not used by this class.

Parameters:

comp - the component to remove

Error Index

preferredLayoutSize

```
public Dimension preferredLayoutSize(Container target)
```

Returns the preferred dimensions for this layout given the components in the specified target container.

Parameters:

target - the component which needs to be laid out

See Also:

Container, minimumLayoutSize

Error Index

minimumLayoutSize

```
public Dimension minimumLayoutSize(Container target)
```

Returns the minimum dimensions needed to layout the components contained in the specified target container.

Parameters:

target - the component which needs to be laid out

See Also:

preferredLayoutSize

Error Index layoutContainer

public void layoutContainer(Container target)

Lays out the container. This method will actually reshape the components in the target in order to satisfy the constraints of the BorderLayout object.

Parameters:

target - the specified component being laid out.

See Also:

Container

Error Index toString

public String toString()

Returns the String representation of this FlowLayout's values.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Font

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Font

[java.lang.Object](#)
|
+----java.awt.Font

public class **Font**
extends [Object](#)

A class that produces font objects.

Error Index

Error Index

BOLD

The bold style constant.

Error Index

ITALIC

The italicized style constant.

Error Index

PLAIN

The plain style constant.

Error Index

name

The logical name of this font.

Error Index

size

The point size of this font.

Error Index

style

The style of the font.

Error Index

Error Index

Font(String, int, int)

Creates a new font with the specified name, style and point size.

Error Index

Error Index

equals(Object)

Compares this object to the specified object.

Error Index

getFamily()

Gets the platform specific family name of the font.

Error Index

getFont(String)

Gets a font from the system properties list.

Error Index

getFont(String, Font)

Gets the specified font from the system properties list.

Error Index

getName()

Gets the logical name of the font.

Error Index

getSize()

Gets the point size of the font.

Error Index

getStyle()

Gets the style of the font.

Error Index

hashCode()

Returns a hashcode for this font.

Error Index

isBold()

Returns true if the font is bold.

Error Index

isItalic()

Returns true if the font is italic.

Error Index

isPlain()

Returns true if the font is plain.

Error Index

toString()

Converts this object to a String representation.

Error Index

Error Index PLAIN

public final static int PLAIN

The plain style constant. This can be combined with the other style constants for mixed styles.

Error Index BOLD

public final static int BOLD

The bold style constant. This can be combined with the other style constants for mixed styles.

Error Index ITALIC

public final static int ITALIC

The italicized style constant. This can be combined with the other style constants for mixed styles.

Error Index name

protected String name

The logical name of this font.

Error Index style

protected int style

The style of the font. This is the sum of the constants PLAIN, BOLD, or ITALIC.

Error Index size

protected int size

The point size of this font.

Error Index

Error Index Font

```
public Font(String name,  
           int style,  
           int size)
```

Creates a new font with the specified name, style and point size.

Parameters:

name - the font name

style - the constant style used

size - the point size of the font

See Also:

getFontList

Error Index

Error Index getFamily

```
public String getFamily()
```

Gets the platform specific family name of the font. Use getName to get the logical name of the font.

See Also:

getName

Error Index getName

```
public String getName()
```

Gets the logical name of the font.

See Also:

getFamily

Error Index getStyle

```
public int getStyle()
```

Gets the style of the font.

See Also:

isPlain, isBold, isItalic

Error Index getSize

```
public int getSize()
```

Gets the point size of the font.

Error Index isPlain

```
public boolean isPlain()
```

Returns true if the font is plain.

See Also:

getStyle

Error Index isBold

```
public boolean isBold()
```

Returns true if the font is bold.

See Also:

getStyle

Error Index isItalic

```
public boolean isItalic()
```

Returns true if the font is italic.

See Also:

getStyle

Error Index `getFont`

```
public static Font getFont(String nm)
```

Gets a font from the system properties list.

Parameters:

nm - the property name

Error Index `getFont`

```
public static Font getFont(String nm,  
                           Font font)
```

Gets the specified font from the system properties list.

Parameters:

nm - the property name

font - a default font to return if property 'nm' is not defined

Error Index `hashCode`

```
public int hashCode()
```

Returns a hashcode for this font.

Overrides:

hashCode in class Object

Error Index `equals`

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

Compares this object to the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index toString

public String toString()

Converts this object to a String representation.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.FontMetrics

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.FontMetrics

[java.lang.Object](#)

|

+----java.awt.FontMetrics

```
public class FontMetrics
    extends Object
```

A font metrics object. Note that the implementations of these methods are inefficient, they are usually overridden with more efficient toolkit specific implementations.

Error Index

Error Index

font

The actual font.

Error Index

Error Index

FontMetrics(Font)

Creates a new FontMetrics object with the specified font.

Error Index

Error Index

bytesWidth(byte[], int, int)

Returns the width of the specified array of bytes in this Font.

Error Index

charWidth(int)

Returns the width of the specified character in this Font.

Error Index

charWidth(char)

Returns the width of the specified character in this Font.

Error Index

charsWidth(char[], int, int)

Returns the width of the specified character array in this Font.

Error Index

getAscent()

Gets the font ascent.

Error Index

getDescent()

Gets the font descent.

Error Index

getFont()

Gets the font.

Error Index

getHeight()

Gets the total height of the font.

Error Index

getLeading()

Gets the standard leading, or line spacing, for the font.

Error Index

getMaxAdvance()

Gets the maximum advance width of any character in this Font.

Error Index

getMaxAscent()

Gets the maximum ascent of all characters in this Font.

Error Index

getMaxDecent()

For backward compatibility only.

Error Index

getMaxDescent()

Gets the maximum descent of all characters.

Error Index

getWidths()

Gets the widths of the first 256 characters in the Font.

Error Index

stringWidth(String)

Returns the width of the specified String in this Font.

Error Index

toString()

Returns the String representation of this FontMetric's values.

Error Index

Error Index font

protected Font font

The actual font.

See Also:

getFont

Error Index

Error Index FontMetrics

protected FontMetrics(Font font)

Creates a new FontMetrics object with the specified font.

Parameters:

font - the font

See Also:

Font

Error Index

Error Index getFont

public Font getFont()

Gets the font.

Error Index getLeading

public int getLeading()

Gets the standard leading, or line spacing, for the font. This is the logical amount of space to be reserved between the descent of one line of text and the ascent of the next line. The height metric is calculated to include this extra space.

Error Index `getAscent`

```
public int getAscent()
```

Gets the font ascent. The font ascent is the distance from the base line to the top of the characters.

See Also:

[getMaxAscent](#)

Error Index `getDescent`

```
public int getDescent()
```

Gets the font descent. The font descent is the distance from the base line to the bottom of the characters.

See Also:

[getMaxDescent](#)

Error Index `getHeight`

```
public int getHeight()
```

Gets the total height of the font. This is the distance between the baseline of adjacent lines of text. It is the sum of the leading + ascent + descent.

Error Index `getMaxAscent`

```
public int getMaxAscent()
```

Gets the maximum ascent of all characters in this Font. No character will extend further above the baseline than this metric.

See Also:

[getAscent](#)

Error Index `getMaxDescent`

```
public int getMaxDescent()
```

Gets the maximum descent of all characters. No character will descend further below the baseline than this metric.

See Also:

[getDescent](#)

Error Index `getMaxDecent`

```
public int getMaxDecent()
```

For backward compatibility only.

See Also:

[getMaxDescent](#)

Error Index `getMaxAdvance`

```
public int getMaxAdvance()
```

Gets the maximum advance width of any character in this Font.

Returns:

-1 if the max advance is not known.

Error Index `charWidth`

```
public int charWidth(int ch)
```

Returns the width of the specified character in this Font.

Parameters:

ch - the specified font

See Also:

[stringWidth](#)

Error Index `charWidth`

```
public int charWidth(char ch)
```

Returns the width of the specified character in this Font.

Parameters:

ch - the specified font

See Also:

stringWidth

Error Index stringWidth

```
public int stringWidth(String str)
```

Returns the width of the specified String in this Font.

Parameters:

str - the String to be checked

See Also:

charsWidth, bytesWidth

Error Index charsWidth

```
public int charsWidth(char data[],  
    int off,  
    int len)
```

Returns the width of the specified character array in this Font.

Parameters:

data - the data to be checked

off - the start offset of the data

len - the maximum number of bytes checked

See Also:

stringWidth, bytesWidth

Error Index bytesWidth

```
public int bytesWidth(byte data[],  
    int off,  
    int len)
```

Returns the width of the specified array of bytes in this Font.

Parameters:

data - the data to be checked

off - the start offset of the data

len - the maximum number of bytes checked

See Also:

stringWidth, charsWidth

Error Index getWidths

```
public int[] getWidths()
```

Gets the widths of the first 256 characters in the Font.

Error Index toString

```
public String toString()
```

Returns the String representation of this FontMetric's values.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Frame

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Frame

```
java.lang.Object
|
+----java.awt.Component
|
+----java.awt.Container
|
+----java.awt.Window
|
+----java.awt.Frame
```

```
public class Frame
  extends Window
  implements MenuContainer
```

A Frame is a top-level window with a title. The default layout for a frame is BorderLayout.

Error Index

Error Index

CROSSHAIR_CURSOR

Error Index

DEFAULT_CURSOR

Error Index

E_RESIZE_CURSOR

Error Index

HAND_CURSOR

Error Index

MOVE_CURSOR

Error Index

NE_RESIZE_CURSOR

Error Index

NW_RESIZE_CURSOR

Error Index

N_RESIZE_CURSOR

Error Index

SE_RESIZE_CURSOR

Error Index

SW_RESIZE_CURSOR

Error Index

S_RESIZE_CURSOR

Error Index

TEXT_CURSOR

Error Index

WAIT_CURSOR

Error Index

W_RESIZE_CURSOR

Error Index

Error Index

Frame()

Constructs a new Frame that is initially invisible.

Error Index

Frame(String)

Constructs a new, initially invisible Frame with the specified title.

Error Index

Error Index

addNotify()

Creates the Frame's peer.

Error Index

dispose()

Disposes of the Frame.

Error Index

getCursorType()

Return the cursor type

Error Index

getIconImage()

Returns the icon image for this Frame.

Error Index

getMenuBar()

Gets the menu bar for this Frame.

Error Index

getTitle()

Gets the title of the Frame.

Error Index

isResizable()

Returns true if the user can resize the Frame.

Error Index

paramString()

Returns the parameter String of this Frame.

Error Index

remove()(MenuComponent)

Removes the specified menu bar from this Frame.

Error Index

setCursor()(int)

Set the cursor image to a predefined cursor.

Error Index

setIconImage()(Image)

Sets the image to display when this Frame is iconized.

Error Index

setMenuBar()(MenuBar)

Sets the menubar for this Frame to the specified menubar.

Error Index

setResizable()(boolean)

Sets the resizable flag.

Error Index

setTitle()(String)

Sets the title for this Frame to the specified title.

Error Index

Error Index

DEFAULT_CURSOR

public final static int DEFAULT_CURSOR

Error Index

CROSSHAIR_CURSOR

public final static int CROSSHAIR_CURSOR

Error Index

TEXT_CURSOR

```
public final static int TEXT_CURSOR
```

Error Index WAIT_CURSOR

```
public final static int WAIT_CURSOR
```

Error Index SW_RESIZE_CURSOR

```
public final static int SW_RESIZE_CURSOR
```

Error Index SE_RESIZE_CURSOR

```
public final static int SE_RESIZE_CURSOR
```

Error Index NW_RESIZE_CURSOR

```
public final static int NW_RESIZE_CURSOR
```

Error Index NE_RESIZE_CURSOR

```
public final static int NE_RESIZE_CURSOR
```

Error Index N_RESIZE_CURSOR

```
public final static int N_RESIZE_CURSOR
```

Error Index S_RESIZE_CURSOR

```
public final static int S_RESIZE_CURSOR
```

Error Index W_RESIZE_CURSOR

```
public final static int W_RESIZE_CURSOR
```

Error Index E_RESIZE_CURSOR

```
public final static int E_RESIZE_CURSOR
```

Error Index HAND_CURSOR

```
public final static int HAND_CURSOR
```

Error Index MOVE_CURSOR

```
public final static int MOVE_CURSOR
```

Error Index

Error Index Frame

```
public Frame()
```

Constructs a new Frame that is initially invisible.

See Also:

[resize](#), [show](#)

Error Index Frame

```
public Frame(String title)
```

Constructs a new, initially invisible Frame with the specified title.

Parameters:

title - te specified title

See Also:

[resize](#), [show](#)

Error Index

Error Index addNotify

public synchronized void addNotify()

Creates the Frame's peer. The peer allows us to change the look of the Frame without changing its functionality.

Overrides:

addNotify in class Window

Error Index getTitle

public String getTitle()

Gets the title of the Frame.

See Also:

setTitle

Error Index setTitle

public void setTitle(String title)

Sets the title for this Frame to the specified title.

Parameters:

title - the specified title of this Frame

See Also:

getTitle

Error Index getIconImage

public Image getIconImage()

Returns the icon image for this Frame.

Error Index setIconImage

public void setIconImage(Image image)

Sets the image to display when this Frame is iconized. Note that not all platforms support the concept of iconizing a window.

Parameters:

image - the icon image to be displayed

Error Index

getMenuBar

public MenuBar getMenuBar()

Gets the menu bar for this Frame.

Error Index

setMenuBar

public synchronized void setMenuBar(MenuBar mb)

Sets the menubar for this Frame to the specified menubar.

Parameters:

mb - the menubar being set

Error Index

remove

public synchronized void remove(MenuComponent m)

Removes the specified menu bar from this Frame.

Error Index

dispose

public synchronized void dispose()

Disposes of the Frame. This method must be called to release the resources that are used for the frame.

Overrides:

dispose in class Window

Error Index isResizable

```
public boolean isResizable()
```

Returns true if the user can resize the Frame.

Error Index setResizable

```
public void setResizable(boolean resizable)
```

Sets the resizable flag.

Parameters:

resizable - true if resizable; false otherwise.

Error Index setCursor

```
public void setCursor(int cursorType)
```

Set the cursor image to a predefined cursor.

Parameters:

cursorType - one of the cursor constants defined above.

Error Index getCursorType

```
public int getCursorType()
```

Return the cursor type

Error Index paramString

```
protected String paramString()
```

Returns the parameter String of this Frame.

Overrides:

paramString in class Container

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Graphics

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Graphics

[java.lang.Object](#)

|

+----java.awt.Graphics

```
public class Graphics
  extends Object
```

Graphics is the abstract base class for all graphic contexts for various devices.

Error Index

Error Index

[Graphics\(\)](#)

Constructs a new Graphics Object.

Error Index

Error Index

[clearRect](#)(int, int, int, int)

Clears the specified rectangle by filling it with the current background color of the current drawing surface.

Error Index

[clipRect](#)(int, int, int, int)

Clips to a rectangle.

Error Index

[copyArea](#)(int, int, int, int, int, int)

Copies an area of the screen.

Error Index

[create](#)()

Creates a new Graphics Object that is a copy of the original Graphics Object.

Error Index

[create](#)(int, int, int, int)

Creates a new Graphics Object with the specified parameters, based on the original Graphics Object.

Error Index

dispose()

Disposes of this graphics context.

Error Index

draw3DRect(int, int, int, int, boolean)

Draws a highlighted 3-D rectangle.

Error Index

drawArc(int, int, int, int, int, int)

Draws an arc bounded by the specified rectangle from startAngle to endAngle.

Error Index

drawBytes(byte[], int, int, int, int)

Draws the specified bytes using the current font and color.

Error Index

drawChars(char[], int, int, int, int)

Draws the specified characters using the current font and color.

Error Index

drawImage(Image, int, int, ImageObserver)

Draws the specified image at the specified coordinate (x, y).

Error Index

drawImage(Image, int, int, int, int, ImageObserver)

Draws the specified image inside the specified rectangle.

Error Index

drawImage(Image, int, int, Color, ImageObserver)

Draws the specified image at the specified coordinate (x, y), with the given solid background Color.

Error Index

drawImage(Image, int, int, int, int, Color, ImageObserver)

Draws the specified image inside the specified rectangle, with the given solid background Color.

Error Index

drawLine(int, int, int, int)

Draws a line between the coordinates (x1,y1) and (x2,y2).

Error Index

drawOval(int, int, int, int)

Draws an oval inside the specified rectangle using the current color.

Error Index

drawPolygon(int[], int[], int)

Draws a polygon defined by an array of x points and y points.

Error Index

drawPolygon(Polygon)

Draws a polygon defined by the specified point.

Error Index

drawRect(int, int, int, int)

Draws the outline of the specified rectangle using the current color.

Error Index

drawRoundRect(int, int, int, int, int, int)

Draws an outlined rounded corner rectangle using the current color.

Error Index

drawString(String, int, int)

Draws the specified String using the current font and color.

Error Index

fill3DRect(int, int, int, int, boolean)

Paints a highlighted 3-D rectangle using the current color.

Error Index

fillArc(int, int, int, int, int, int)

Fills an arc using the current color.

Error Index

fillOval(int, int, int, int)

Fills an oval inside the specified rectangle using the current color.

Error Index

fillPolygon(int[], int[], int)

Fills a polygon with the current color using an even-odd fill rule (otherwise known as an alternating rule).

Error Index

fillPolygon(Polygon)

Fills the specified polygon with the current color using an even-odd fill rule (otherwise known as an alternating rule).

Error Index

fillRect(int, int, int, int)

Fills the specified rectangle with the current color.

Error Index

fillRoundRect(int, int, int, int, int, int)

Draws a rounded rectangle filled in with the current color.

Error Index

finalize()

Disposes of this graphics context once it is no longer referenced.

Error Index

getClipRect()

Returns the bounding rectangle of the current clipping area.

Error Index

getColor()

Gets the current color.

Error Index

getFont()

Gets the current font.

Error Index

getFontMetrics()

Gets the current font metrics.

Error Index

getFontMetrics(Font)

Gets the current font metrics for the specified font.

Error Index

setColor(Color)

Sets the current color to the specified color.

Error Index

setFont(Font)

Sets the font for all subsequent text-drawing operations.

Error Index

setPaintMode()

Sets the paint mode to overwrite the destination with the current color.

Error Index

setXORMode(Color)

Sets the paint mode to alternate between the current color and the new specified color.

Error Index

toString()

Returns a String object representing this Graphic's value.

Error Index

translate(int, int)

Translates the specified parameters into the origin of the graphics context.

Error Index

Error Index

Graphics

protected Graphics()

Constructs a new Graphics Object. Graphic contexts cannot be created directly. They must be obtained from another graphics context or created by a Component.

See Also:

getGraphics, create

Error Index

Error Index

create

public abstract Graphics create()

Creates a new Graphics Object that is a copy of the original Graphics Object.

Error Index

create

```
public Graphics create(int x,  
                        int y,  
                        int width,  
                        int height)
```

Creates a new Graphics Object with the specified parameters, based on the original Graphics Object. This method translates the specified parameters, x and y, to the proper origin coordinates and then clips the Graphics Object to the area.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the area
height - the height of the area

See Also:

translate

Error Index **translate**

```
public abstract void translate(int x,  
                               int y)
```

Translates the specified parameters into the origin of the graphics context. All subsequent operations on this graphics context will be relative to this origin.

Parameters:

x - the x coordinate
y - the y coordinate

Error Index **getColor**

```
public abstract Color getColor()
```

Gets the current color.

See Also:

setColor

Error Index **setColor**

```
public abstract void setColor(Color c)
```

Sets the current color to the specified color. All subsequent graphics operations will use this specified color.

Parameters:

c - the color to be set

See Also:

Color, getColor

Error Index

setPaintMode

public abstract void setPaintMode()

Sets the paint mode to overwrite the destination with the current color.

Error Index

setXORMode

public abstract void setXORMode(Color c1)

Sets the paint mode to alternate between the current color and the new specified color. When drawing operations are performed, pixels which are the current color will be changed to the specified color and vice versa. Pixels of colors other than those two colors will be changed in an unpredictable, but reversible manner - if you draw the same figure twice then all pixels will be restored to their original values.

Parameters:

c1 - the second color

Error Index

getFont

public abstract Font getFont()

Gets the current font.

See Also:

setFont

Error Index

setFont

public abstract void setFont(Font font)

Sets the font for all subsequent text-drawing operations.

Parameters:

font - the specified font

See Also:

[Font](#), [getFont](#), [drawString](#), [drawBytes](#), [drawChars](#)

Error Index

getFontMetrics

public [FontMetrics](#) getFontMetrics()

Gets the current font metrics.

See Also:

[getFont](#)

Error Index

getFontMetrics

public abstract [FontMetrics](#) getFontMetrics([Font](#) f)

Gets the current font metrics for the specified font.

Parameters:

f - the specified font

See Also:

[getFont](#), [getFontMetrics](#)

Error Index

getClipRect

public abstract [Rectangle](#) getClipRect()

Returns the bounding rectangle of the current clipping area.

See Also:

[clipRect](#)

Error Index

clipRect

public abstract void clipRect(int x,
int y,
int width,
int height)

Clips to a rectangle. The resulting clipping area is the intersection of the current clipping area and the specified rectangle. Graphic operations have no effect outside of the clipping area.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle

See Also:

getClipRect

Error Index copyArea

```
public abstract void copyArea(int x,  
                             int y,  
                             int width,  
                             int height,  
                             int dx,  
                             int dy)
```

Copies an area of the screen.

Parameters:

x - the x-coordinate of the source
y - the y-coordinate of the source
width - the width
height - the height
dx - the horizontal distance
dy - the vertical distance

Error Index drawLine

```
public abstract void drawLine(int x1,  
                              int y1,  
                              int x2,  
                              int y2)
```

Draws a line between the coordinates (x1,y1) and (x2,y2). The line is drawn below and to the left of the logical coordinates.

Parameters:

x1 - the first point's x coordinate
y1 - the first point's y coordinate
x2 - the second point's x coordinate
y2 - the second point's y coordinate

Error Index fillRect

```
public abstract void fillRect(int x,  
                             int y,  
                             int width,  
                             int height)
```

Fills the specified rectangle with the current color.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle

See Also:

[drawRect](#), [clearRect](#)

Error Index drawRect

```
public void drawRect(int x,  
                    int y,  
                    int width,  
                    int height)
```

Draws the outline of the specified rectangle using the current color. Use drawRect(x, y, width-1, height-1) to draw the outline inside the specified rectangle.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle

See Also:

[fillRect](#), [clearRect](#)

Error Index clearRect

```
public abstract void clearRect(int x,  
                              int y,  
                              int width,  
                              int height)
```

Clears the specified rectangle by filling it with the current background color of the current drawing surface. Which drawing surface it selects depends on how the graphics context was created.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle

See Also:

[fillRect](#), [drawRect](#)

Error Index **drawRoundRect**

```
public abstract void drawRoundRect(int x,  
                                   int y,  
                                   int width,  
                                   int height,  
                                   int arcWidth,  
                                   int arcHeight)
```

Draws an outlined rounded corner rectangle using the current color.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

arcWidth - the horizontal diameter of the arc at the four corners

arcHeight - the horizontal diameter of the arc at the four corners

See Also:

[fillRoundRect](#)

Error Index **fillRoundRect**

```
public abstract void fillRoundRect(int x,  
                                   int y,  
                                   int width,  
                                   int height,  
                                   int arcWidth,  
                                   int arcHeight)
```

Draws a rounded rectangle filled in with the current color.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

arcWidth - the horizontal diameter of the arc at the four corners

arcHeight - the horizontal diameter of the arc at the four corners

See Also:

[drawRoundRect](#)

Error Index

draw3DRect

```
public void draw3DRect(int x,  
    int y,  
    int width,  
    int height,  
    boolean raised)
```

Draws a highlighted 3-D rectangle.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

raised - a boolean that states whether the rectangle is raised or not

Error Index

fill3DRect

```
public void fill3DRect(int x,  
    int y,  
    int width,  
    int height,  
    boolean raised)
```

Paints a highlighted 3-D rectangle using the current color.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

raised - a boolean that states whether the rectangle is raised or not

Error Index

drawOval

```
public abstract void drawOval(int x,  
    int y,  
    int width,  
    int height)
```

Draws an oval inside the specified rectangle using the current color.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

See Also:

[fillOval](#)

Error Index [fillOval](#)

```
public abstract void fillOval(int x,  
                             int y,  
                             int width,  
                             int height)
```

Fills an oval inside the specified rectangle using the current color.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

See Also:

[drawOval](#)

Error Index [drawArc](#)

```
public abstract void drawArc(int x,  
                             int y,  
                             int width,  
                             int height,  
                             int startAngle,  
                             int arcAngle)
```

Draws an arc bounded by the specified rectangle from startAngle to endAngle. 0 degrees is at the 3-o'clock position. Positive arc angles indicate counter-clockwise rotations, negative arc angles are drawn clockwise.

Parameters:

x - the x coordinate

y - the y coordinate

width - the width of the rectangle

height - the height of the rectangle

startAngle - the beginning angle

arcAngle - the angle of the arc (relative to startAngle).

See Also:

[fillArc](#)

Error Index [fillArc](#)

```
public abstract void fillArc(int x,  
    int y,  
    int width,  
    int height,  
    int startAngle,  
    int arcAngle)
```

Fills an arc using the current color. This generates a pie shape.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the arc
height - the height of the arc
startAngle - the beginning angle
arcAngle - the angle of the arc (relative to startAngle).

See Also:

[drawArc](#)

Error Index

drawPolygon

```
public abstract void drawPolygon(int xPoints[],  
    int yPoints[],  
    int nPoints)
```

Draws a polygon defined by an array of x points and y points.

Parameters:

xPoints - an array of x points
yPoints - an array of y points
nPoints - the total number of points

See Also:

[fillPolygon](#)

Error Index

drawPolygon

```
public void drawPolygon(Polygon p)
```

Draws a polygon defined by the specified point.

Parameters:

p - the specified polygon

See Also:

[fillPolygon](#)

Error Index

fillPolygon

```
public abstract void fillPolygon(int xPoints[],
                                int yPoints[],
                                int nPoints)
```

Fills a polygon with the current color using an even-odd fill rule (otherwise known as an alternating rule).

Parameters:

xPoints - an array of x points
yPoints - an array of y points
nPoints - the total number of points

See Also:

[drawPolygon](#)

Error Index

fillPolygon

```
public void fillPolygon(Polygon p)
```

Fills the specified polygon with the current color using an even-odd fill rule (otherwise known as an alternating rule).

Parameters:

p - the polygon

See Also:

[drawPolygon](#)

Error Index

drawString

```
public abstract void drawString(String str,
                                int x,
                                int y)
```

Draws the specified String using the current font and color. The x,y position is the starting point of the baseline of the String.

Parameters:

str - the String to be drawn
x - the x coordinate
y - the y coordinate

See Also:

[drawChars](#), [drawBytes](#)

Error Index

drawChars

```
public void drawChars(char data[],  
    int offset,  
    int length,  
    int x,  
    int y)
```

Draws the specified characters using the current font and color.

Parameters:

data - the array of characters to be drawn
offset - the start offset in the data
length - the number of characters to be drawn
x - the x coordinate
y - the y coordinate

See Also:

[drawString](#), [drawBytes](#)

Error Index

drawBytes

```
public void drawBytes(byte data[],  
    int offset,  
    int length,  
    int x,  
    int y)
```

Draws the specified bytes using the current font and color.

Parameters:

data - the data to be drawn
offset - the start offset in the data
length - the number of bytes that are drawn
x - the x coordinate
y - the y coordinate

See Also:

[drawString](#), [drawChars](#)

Error Index

drawImage

```
public abstract boolean drawImage(Image img,  
    int x,  
    int y,  
    ImageObserver observer)
```

Draws the specified image at the specified coordinate (x, y). If the image is incomplete the image

observer will be notified later.

Parameters:

img - the specified image to be drawn
x - the x coordinate
y - the y coordinate
observer - notifies if the image is complete or not

See Also:

Image, ImageObserver

Error Index

drawImage

```
public abstract boolean drawImage(Image img,  
    int x,  
    int y,  
    int width,  
    int height,  
    ImageObserver observer)
```

Draws the specified image inside the specified rectangle. The image is scaled if necessary. If the image is incomplete the image observer will be notified later.

Parameters:

img - the specified image to be drawn
x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle
observer - notifies if the image is complete or not

See Also:

Image, ImageObserver

Error Index

drawImage

```
public abstract boolean drawImage(Image img,  
    int x,  
    int y,  
    Color bgcolor,  
    ImageObserver observer)
```

Draws the specified image at the specified coordinate (x, y), with the given solid background Color. If the image is incomplete the image observer will be notified later.

Parameters:

img - the specified image to be drawn
x - the x coordinate
y - the y coordinate
observer - notifies if the image is complete or not

See Also:

Image, ImageObserver

Error Index

drawImage

```
public abstract boolean drawImage(Image img,  
    int x,  
    int y,  
    int width,  
    int height,  
    Color bgcolor,  
    ImageObserver observer)
```

Draws the specified image inside the specified rectangle, with the given solid background Color. The image is scaled if necessary. If the image is incomplete the image observer will be notified later.

Parameters:

img - the specified image to be drawn
x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle
observer - notifies if the image is complete or not

See Also:

Image, ImageObserver

Error Index

dispose

```
public abstract void dispose()
```

Disposes of this graphics context. The Graphics context cannot be used after being disposed of.

See Also:

finalize

Error Index

finalize

```
public void finalize()
```

Disposes of this graphics context once it is no longer referenced.

Overrides:

finalize in class Object

See Also:

dispose

Error Index toString

public String toString()

Returns a String object representing this Graphic's value.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.GridBagConstraints

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.GridBagConstraints

[java.lang.Object](#)

|

+----java.awt.GridBagConstraints

public class **GridBagConstraints**
extends [Object](#)
implements [Cloneable](#)

GridBagConstraints is used to specify constraints for components laid out using the GridBagLayout class.

See Also:

[GridBagLayout](#)

Error Index

Error Index

[BOTH](#)

Error Index

[CENTER](#)

Error Index

[EAST](#)

Error Index

[HORIZONTAL](#)

Error Index

[NONE](#)

Error Index

[NORTH](#)

Error Index

[NORTHEAST](#)

Error Index

[NORTHWEST](#)

Error Index

[RELATIVE](#)

Error Index

REMAINDER

Error Index

SOUTH

Error Index

SOUTHEAST

Error Index

SOUTHWEST

Error Index

VERTICAL

Error Index

WEST

Error Index

anchor

Error Index

fill

Error Index

gridheight

Error Index

gridwidth

Error Index

gridx

Error Index

gridy

Error Index

insets

Error Index

ipadx

Error Index

ipady

Error Index

weightx

Error Index

weighty

Error Index

Error Index

GridBagConstraints()

Error Index

Error Index

clone()

Creates a clone of the object.

Error Index

Error Index

RELATIVE

public final static int RELATIVE

Error Index

REMAINDER

public final static int REMAINDER

Error Index

NONE

public final static int NONE

Error Index

BOTH

public final static int BOTH

Error Index

HORIZONTAL

public final static int HORIZONTAL

Error Index

VERTICAL

public final static int VERTICAL

Error Index CENTER

public final static int CENTER

Error Index NORTH

public final static int NORTH

Error Index NORTHEAST

public final static int NORTHEAST

Error Index EAST

public final static int EAST

Error Index SOUTHEAST

public final static int SOUTHEAST

Error Index SOUTH

public final static int SOUTH

Error Index SOUTHWEST

public final static int SOUTHWEST

Error Index WEST

public final static int WEST

Error Index NORTHWEST

```
public final static int NORTHWEST
```

Error Index_{gridx}

```
public int gridx
```

Error Index_{gridy}

```
public int gridy
```

Error Index_{gridwidth}

```
public int gridwidth
```

Error Index_{gridheight}

```
public int gridheight
```

Error Index_{weightx}

```
public double weightx
```

Error Index_{weighty}

```
public double weighty
```

Error Index_{anchor}

```
public int anchor
```

Error Index_{fill}

```
public int fill
```


Error Index insets

public Insets insets

Error Index ipadx

public int ipadx

Error Index ipady

public int ipady

Error Index

Error Index GridBagConstraints

public GridBagConstraints()

Error Index

Error Index clone

public Object clone()

Creates a clone of the object.

Overrides:

clone in class Object

Class java.awt.GridBagLayout

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.GridBagLayout

[java.lang.Object](#)

|
+----java.awt.GridBagLayout

```
public class GridBagLayout
    extends Object
    implements LayoutManager
```

GridBagLayout is a flexible layout manager that aligns components vertically and horizontally, without requiring that the components be the same size. Each GridBagLayout uses a rectangular grid of cells, with each component occupying one or more cells (called its *display area*). Each component managed by a GridBagLayout is associated with a [GridBagConstraints](#) instance that specifies how the component is laid out within its display area. How a GridBagLayout places a set of components depends on each component's GridBagConstraints and minimum size, as well as the preferred size of the components' container.

To use a GridBagLayout effectively, you must customize one or more of its components' GridBagConstraints. You customize a GridBagConstraints object by setting one or more of its instance variables:

[gridx](#), [gridy](#)

Specifies the cell at the upper left of the component's display area, where the upper-left-most cell has address gridx=0, gridy=0. Use GridBagConstraints.RELATIVE (the default value) to specify that the component be placed just to the right of (for gridx) or just below (for gridy) the component that was added to the container just before this component was added.

[gridwidth](#), [gridheight](#)

Specifies the number of cells in a row (for gridwidth) or column (for gridheight) in the component's display area. The default value is 1. Use GridBagConstraints.REMAINDER to specify that the component be the last one in its row (for gridwidth) or column (for gridheight). Use GridBagConstraints.RELATIVE to specify that the component be the next to last one in its row (for gridwidth) or column (for gridheight).

[fill](#)

Used when the component's display area is larger than the component's requested size to determine whether (and how) to resize the component. Valid values are GridBagConstraints.NONE (the default), GridBagConstraints.HORIZONTAL (make the component wide enough to fill its display area horizontally, but don't change its height), GridBagConstraints.VERTICAL (make the component tall enough to fill its display area vertically, but don't change its width), and GridBagConstraints.BOTH (make the component fill its display area entirely).

[ipadx](#), [ipady](#)

Specifies the internal padding: how much to add to the minimum size of the component. The width of the component will be at least its minimum width plus ipadx*2 pixels (since

the padding applies to both sides of the component). Similarly, the height of the component will be at least the minimum height plus $ipady*2$ pixels.

insets

Specifies the external padding of the component -- the minimum amount of space between the component and the edges of its display area.

anchor

Used when the component is smaller than its display area to determine where (within the area) to place the component. Valid values are

GridBagConstraints.CENTER (the default),
GridBagConstraints.NORTH, GridBagConstraints.NORTHEAST,
GridBagConstraints.EAST, GridBagConstraints.SOUTHEAST,
GridBagConstraints.SOUTH, GridBagConstraints.SOUTHWEST,
GridBagConstraints.WEST, and GridBagConstraints.NORTHWEST.

weightx, weighty

Used to determine how to distribute space; this is important for specifying resizing behavior. Unless you specify a weight for at least one component in a row (weightx) and column (weighty), all the components clump together in the center of their container. This is because when the weight is zero (the default), the GridBagLayout puts any extra space between its grid of cells and the edges of the container.

The following figure shows ten components (all buttons) managed by a GridBagLayout:

Error Index

All the components have `fill=GridBagConstraints.BOTH`. In addition, the components have the following non-default constraints:

- Button1, Button2, Button3: `weightx=1.0`
- Button4: `weightx=1.0, gridwidth=GridBagConstraints.REMAINDER`
- Button5: `gridwidth=GridBagConstraints.REMAINDER`
- Button6: `gridwidth=GridBagConstraints.RELATIVE`
- Button7: `gridwidth=GridBagConstraints.REMAINDER`
- Button8: `gridheight=2, weighty=1.0,`
- Button9, Button 10: `gridwidth=GridBagConstraints.REMAINDER`

Here is the code that implements the example shown above:

```
import java.awt.*;
import java.util.*;
import java.applet.Applet;
public class GridBagEx1 extends Applet {
    protected void makebutton(String name,
                               GridBagConstraints c) {
        Button button = new Button(name);
        gridbag.setConstraints(button, c);
        add(button);
    }
    public void init() {
        GridBagLayout gridbag = new GridBagLayout();
        GridBagConstraints c = new GridBagConstraints();
        setFont(new Font("Helvetica", Font.PLAIN, 14));
        setLayout(gridbag);
        c.fill = GridBagConstraints.BOTH;
        c.weightx = 1.0;
```

```

        makebutton("Button1", gridbag, c);
        makebutton("Button2", gridbag, c);
        makebutton("Button3", gridbag, c);
        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        makebutton("Button4", gridbag, c);
        c.weightx = 0.0;           //reset to the default
        makebutton("Button5", gridbag, c); //another row
        c.gridwidth = GridBagConstraints.RELATIVE; //next-to-last in row
        makebutton("Button6", gridbag, c);
        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        makebutton("Button7", gridbag, c);
        c.gridwidth = 1;           //reset to the default
        c.gridheight = 2;
        c.weighty = 1.0;
        makebutton("Button8", gridbag, c);
        c.weightx = 0.0;           //reset to the default
        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        c.gridheight = 1;          //reset to the default
        makebutton("Button9", gridbag, c);
        makebutton("Button10", gridbag, c);
        resize(300, 100);
    }
    public static void main(String args[]) {
        Frame f = new Frame("GridBag Layout Example");
        GridBagEx1 ex1 = new GridBagEx1();
        ex1.init();
        f.add("Center", ex1);
        f.pack();
        f.resize(f.preferredSize());
        f.show();
    }
}

```

Error Index

Error Index

MAXGRIDSIZE

Error Index

MINSIZE

Error Index

PREFERRED_SIZE

Error Index

columnWeights

Error Index

columnWidths

Error Index

comptable

Error Index

defaultConstraints

Error Index

layoutInfo

Error Index

rowHeights

Error Index

rowWeights

Error Index

Error Index

GridBagLayout()

Creates a gridbag layout.

Error Index

Error Index

AdjustForGravity(GridBagConstraints, Rectangle)

Error Index

ArrangeGrid(Container)

Error Index

DumpConstraints (GridBagConstraints)

Print the layout constraints.

Error Index

DumpLayoutInfo(GridBagLayoutInfo)

Print the layout information.

Error Index

GetLayoutInfo(Container, int)

Error Index

GetMinSize (Container, GridBagLayoutInfo)

Error Index

addLayoutComponent (String, Component)

Adds the specified component with the specified name to the layout.

Error Index

getConstraints(Component)

Retrieves the constraints for the specified component.

Error Index

getLayoutDimensions()

Error Index

getLayoutOrigin()

Error Index

getLayoutWeights()

Error Index

layoutContainer(Container)

Lays out the container in the specified panel.

Error Index

location(int, int)

Error Index

lookupConstraints(Component)

Retrieves the constraints for the specified component.

Error Index

minimumLayoutSize(Container)

Returns the minimum dimensions needed to layout the components contained in the specified panel.

Error Index

preferredLayoutSize(Container)

Returns the preferred dimensions for this layout given the components in the specified panel.

Error Index

removeLayoutComponent(Component)

Removes the specified component from the layout.

Error Index

setConstraints(Component, GridBagConstraints)

Sets the constraints for the specified component.

Error Index

toString()

Returns the String representation of this GridLayout's values.

Error Index

Error Index

MAXGRIDSZIE

protected final static int MAXGRIDSZIE

Error Index

MINSIZE

protected final static int MINSIZE

Error Index **PREFERRED_SIZE**

protected final static int PREFERRED_SIZE

Error Index **comptable**

protected Hashtable comptable

Error Index **defaultConstraints**

protected GridBagConstraints defaultConstraints

Error Index **layoutInfo**

protected GridBagLayoutInfo layoutInfo

Error Index **columnWidths**

public int columnWidths[]

Error Index **rowHeights**

public int rowHeights[]

Error Index **columnWeights**

public double columnWeights[]

Error Index **rowWeights**

public double rowWeights[]

Error Index

Error Index

GridBagLayout

```
public GridBagLayout()
```

Creates a gridbag layout.

Error Index

Error Index

setConstraints

```
public void setConstraints(Component comp,  
                           GridBagConstraints constraints)
```

Sets the constraints for the specified component.

Parameters:

comp - the component to be modified

constraints - the constraints to be applied

Error Index

getConstraints

```
public GridBagConstraints getConstraints(Component comp)
```

Retrieves the constraints for the specified component. A copy of the constraints is returned.

Parameters:

comp - the component to be queried

Error Index

lookupConstraints

```
protected GridBagConstraints lookupConstraints(Component comp)
```

Retrieves the constraints for the specified component. The return value is not a copy, but is the actual constraints class used by the layout mechanism.

Parameters:

comp - the component to be queried

Error Index **getLayoutOrigin**

```
public Point getLayoutOrigin()
```

Error Index **getLayoutDimensions**

```
public int[][] getLayoutDimensions()
```

Error Index **getLayoutWeights**

```
public double[][] getLayoutWeights()
```

Error Index **location**

```
public Point location(int x,  
                      int y)
```

Error Index **addLayoutComponent**

```
public void addLayoutComponent(String name,  
                               Component comp)
```

Adds the specified component with the specified name to the layout.

Parameters:

name - the name of the component
comp - the component to be added

Error Index **removeLayoutComponent**

```
public void removeLayoutComponent(Component comp)
```

Removes the specified component from the layout. Does not apply.

Parameters:

comp - the component to be removed

Error Index

preferredLayoutSize

public Dimension preferredLayoutSize(Container parent)

Returns the preferred dimensions for this layout given the components in the specified panel.

Parameters:

parent - the component which needs to be laid out

See Also:

minimumLayoutSize

Error Index

minimumLayoutSize

public Dimension minimumLayoutSize(Container parent)

Returns the minimum dimensions needed to layout the components contained in the specified panel.

Parameters:

parent - the component which needs to be laid out

See Also:

preferredLayoutSize

Error Index

layoutContainer

public void layoutContainer(Container parent)

Lays out the container in the specified panel.

Parameters:

parent - the specified component being laid out

See Also:

Container

Error Index

toString

public String toString()

Returns the String representation of this GridLayout's values.

Overrides:

toString in class Object

Error Index

DumpLayoutInfo

protected void DumpLayoutInfo(GridBagLayoutInfo s)

Print the layout information. Useful for debugging.

Error Index

DumpConstraints

protected void DumpConstraints(GridBagConstraints constraints)

Print the layout constraints. Useful for debugging.

Error Index

GetLayoutInfo

protected GridBagLayoutInfo GetLayoutInfo(Container parent,
int sizeflag)

Error Index

AdjustForGravity

protected void AdjustForGravity(GridBagConstraints constraints,
Rectangle r)

Error Index

GetMinSize

protected Dimension GetMinSize(Container parent,
GridBagLayoutInfo info)

Error Index

ArrangeGrid

protected void ArrangeGrid(Container parent)

Class java.awt.GridLayout

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.GridLayout

[java.lang.Object](#)

|

+----java.awt.GridLayout

```
public class GridLayout
  extends Object
  implements LayoutManager
```

A layout manager for a container that lays out grids.

Error Index

Error Index

[**GridLayout**](#)(int, int)

Creates a grid layout with the specified rows and columns.

Error Index

[**GridLayout**](#)(int, int, int, int)

Creates a grid layout with the specified rows, columns, horizontal gap, and vertical gap.

Error Index

Error Index

[**addLayoutComponent**](#)(String, Component)

Adds the specified component with the specified name to the layout.

Error Index

[**layoutContainer**](#)(Container)

Lays out the container in the specified panel.

Error Index

[**minimumLayoutSize**](#)(Container)

Returns the minimum dimensions needed to layout the components contained in the specified panel.

Error Index

[**preferredLayoutSize**](#)(Container)

Returns the preferred dimensions for this layout given the components in the specified panel.

Error Index

removeLayoutComponent(Component)

Removes the specified component from the layout.

Error Index

toString()

Returns the String representation of this GridLayout's values.

Error Index

Error Index

GridLayout

```
public GridLayout(int rows,  
                  int cols)
```

Creates a grid layout with the specified rows and columns.

Parameters:

rows - the rows

cols - the columns

Error Index

GridLayout

```
public GridLayout(int rows,  
                  int cols,  
                  int hgap,  
                  int vgap)
```

Creates a grid layout with the specified rows, columns, horizontal gap, and vertical gap.

Parameters:

rows - the rows; zero means 'any number.'

cols - the columns; zero means 'any number.' Only one of 'rows' and 'cols' can be zero, not both.

hgap - the horizontal gap variable

vgap - the vertical gap variable

Throws: IllegalArgumentException

If the rows and columns are invalid.

Error Index

Error Index

addLayoutComponent

```
public void addLayoutComponent(String name,  
                               Component comp)
```

Adds the specified component with the specified name to the layout.

Parameters:

name - the name of the component
comp - the component to be added

Error Index

removeLayoutComponent

```
public void removeLayoutComponent(Component comp)
```

Removes the specified component from the layout. Does not apply.

Parameters:

comp - the component to be removed

Error Index

preferredLayoutSize

```
public Dimension preferredLayoutSize(Container parent)
```

Returns the preferred dimensions for this layout given the components in the specified panel.

Parameters:

parent - the component which needs to be laid out

See Also:

minimumLayoutSize

Error Index

minimumLayoutSize

```
public Dimension minimumLayoutSize(Container parent)
```

Returns the minimum dimensions needed to layout the components contained in the specified panel.

Parameters:

parent - the component which needs to be laid out

See Also:

preferredLayoutSize

Error Index layoutContainer

public void layoutContainer(Container parent)

Lays out the container in the specified panel.

Parameters:

parent - the specified component being laid out

See Also:

Container

Error Index toString

public String toString()

Returns the String representation of this GridLayout's values.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.ColorModel

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.ColorModel

[java.lang.Object](#)

|

+----[java.awt.image.ColorModel](#)

public class **ColorModel**
extends [Object](#)

A class that encapsulates the methods for translating from pixel values to alpha, red, green, and blue color components for an image. This class is abstract.

See Also:

[IndexColorModel](#), [DirectColorModel](#)

Error Index

Error Index

[pixel_bits](#)

Error Index

Error Index

[ColorModel\(int\)](#)

Constructs a ColorModel which describes a pixel of the specified number of bits.

Error Index

Error Index

[getAlpha\(int\)](#)

The subclass must provide a function which provides the alpha color component for the specified pixel.

Error Index

getBlue(int)

The subclass must provide a function which provides the blue color component for the specified pixel.

Error Index

getGreen(int)

The subclass must provide a function which provides the green color component for the specified pixel.

Error Index

getPixelSize()

Returns the number of bits per pixel described by this ColorModel.

Error Index

getRGB(int)

Returns the color of the pixel in the default RGB color model.

Error Index

getRGBdefault()

Return a ColorModel which describes the default format for integer RGB values used throughout the AWT image interfaces.

Error Index

getRed(int)

The subclass must provide a function which provides the red color component for the specified pixel.

Error Index

Error Index

pixel_bits

protected int pixel_bits

Error Index

Error Index

ColorModel

public ColorModel(int bits)

Constructs a ColorModel which describes a pixel of the specified number of bits.

Error Index

Error Index getRGBdefault

```
public static ColorModel getRGBdefault()
```

Return a ColorModel which describes the default format for integer RGB values used throughout the AWT image interfaces. The format for the RGB values is an integer with 8 bits each of alpha, red, green, and blue color components ordered correspondingly from the most significant byte to the least significant byte, as in: 0xAARRGGBB

Error Index getPixelSize

```
public int getPixelSize()
```

Returns the number of bits per pixel described by this ColorModel.

Error Index getRed

```
public abstract int getRed(int pixel)
```

The subclass must provide a function which provides the red color component for the specified pixel.

Returns:

The red color component ranging from 0 to 255

Error Index getGreen

```
public abstract int getGreen(int pixel)
```

The subclass must provide a function which provides the green color component for the specified pixel.

Returns:

The green color component ranging from 0 to 255

Error Index getBlue

```
public abstract int getBlue(int pixel)
```

The subclass must provide a function which provides the blue color component for the specified pixel.

Returns:

The blue color component ranging from 0 to 255

Error Index **getAlpha**

```
public abstract int getAlpha(int pixel)
```

The subclass must provide a function which provides the alpha color component for the specified pixel.

Returns:

The alpha transparency value ranging from 0 to 255

Error Index **getRGB**

```
public int getRGB(int pixel)
```

Returns the color of the pixel in the default RGB color model.

See Also:

[getRGBdefault](#)

Class java.awt.image.CropImageFilter

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.CropImageFilter

[java.lang.Object](#)

```
|
+----java.awt.image.ImageFilter
      |
      +----java.awt.image.CropImageFilter
```

public class **CropImageFilter**
extends [ImageFilter](#)

An ImageFilter class for cropping images. This class extends the basic ImageFilter Class to extract a given rectangular region of an existing Image and provide a source for a new image containing just the extracted region. It is meant to be used in conjunction with a FilteredImageSource object to produce cropped versions of existing images.

See Also:

[FilteredImageSource](#), [ImageFilter](#)

Error Index

Error Index

[CropImageFilter](#)(int, int, int, int)

Constructs a CropImageFilter that extracts the absolute rectangular region of pixels from its source Image as specified by the x, y, w, and h parameters.

Error Index

Error Index

[setDimensions](#)(int, int)

Override the source image's dimensions and pass the dimensions of the rectangular cropped region to the ImageConsumer.

Error Index

[setPixels](#) (int, int, int, int, ColorModel, byte[], int, int)

Determine whether the delivered byte pixels intersect the region to be extracted and passes through only that subset of pixels that appear in the output region.

Error Index

setPixels (int, int, int, int, ColorModel, int[], int, int)

Determine if the delivered int pixels intersect the region to be extracted and pass through only that subset of pixels that appear in the output region.

Error Index

setProperties(Hashtable)

Passes along the properties from the source object after adding a property indicating the cropped region.

Error Index

Error Index

CropImageFilter

```
public CropImageFilter(int x,  
    int y,  
    int w,  
    int h)
```

Constructs a CropImageFilter that extracts the absolute rectangular region of pixels from its source Image as specified by the x, y, w, and h parameters.

Parameters:

- x - the x location of the top of the rectangle to be extracted
- y - the y location of the top of the rectangle to be extracted
- w - the width of the rectangle to be extracted
- h - the height of the rectangle to be extracted

Error Index

Error Index

setProperties

```
public void setProperties(Hashtable props)
```

Passes along the properties from the source object after adding a property indicating the cropped region.

Overrides:

setProperties in class ImageFilter

Error Index

setDimensions

```
public void setDimensions(int w,  
                          int h)
```

Override the source image's dimensions and pass the dimensions of the rectangular cropped region to the ImageConsumer.

Overrides:

setDimensions in class ImageFilter

See Also:

ImageConsumer

Error Index

setPixels

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

Determine whether the delivered byte pixels intersect the region to be extracted and passes through only that subset of pixels that appear in the output region.

Overrides:

setPixels in class ImageFilter

Error Index

setPixels

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

Determine if the delivered int pixels intersect the region to be extracted and pass through only that subset of pixels that appear in the output region.

Overrides:

setPixels in class ImageFilter

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.DirectColorModel

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.DirectColorModel

[java.lang.Object](#)

```
|
+----java.awt.image.ColorModel
      |
      +----java.awt.image.DirectColorModel
```

public class **DirectColorModel**
extends [ColorModel](#)

A ColorModel class that specifies a translation from pixel values to alpha, red, green, and blue color components for pixels which have the color components embedded directly in the bits of the pixel itself. This color model is similar to an X11 TrueColor visual.

Many of the methods in this class are final. This is because the underlying native graphics code makes assumptions about the layout and operation of this class and those assumptions are reflected in the implementations of the methods here that are marked final. You can subclass this class for other reasons, but you cannot override or modify the behaviour of those methods.

See Also:

[ColorModel](#)

Error Index

Error Index

[DirectColorModel](#)(int, int, int, int)

Constructs a DirectColorModel from the given masks specifying which bits in the pixel contain the red, green and blue color components.

Error Index

[DirectColorModel](#)(int, int, int, int, int)

Constructs a DirectColorModel from the given masks specifying which bits in the pixel contain the alpha, red, green and blue color components.

Error Index

Error Index

getAlpha(int)

Returns the alpha transparency value for the specified pixel in the range 0-255.

Error Index

getAlphaMask()

Returns the mask indicating which bits in a pixel contain the alpha transparency component.

Error Index

getBlue(int)

Returns the blue color component for the specified pixel in the range 0-255.

Error Index

getBlueMask()

Returns the mask indicating which bits in a pixel contain the blue color component.

Error Index

getGreen(int)

Returns the green color component for the specified pixel in the range 0-255.

Error Index

getGreenMask()

Returns the mask indicating which bits in a pixel contain the green color component.

Error Index

getRGB(int)

Returns the color of the pixel in the default RGB color model.

Error Index

getRed(int)

Returns the red color component for the specified pixel in the range 0-255.

Error Index

getRedMask()

Returns the mask indicating which bits in a pixel contain the red color component.

Error Index

Error Index

DirectColorModel

```
public DirectColorModel(int bits,  
                        int rmask,  
                        int gmask,  
                        int bmask)
```

Constructs a DirectColorModel from the given masks specifying which bits in the pixel contain the red, green and blue color components. Pixels described by this color model will all have alpha components of 255 (fully opaque). All of the bits in each mask must be contiguous and fit in the specified number of least significant bits of the integer.

Error Index

DirectColorModel

```
public DirectColorModel(int bits,  
    int rmask,  
    int gmask,  
    int bmask,  
    int amask)
```

Constructs a DirectColorModel from the given masks specifying which bits in the pixel contain the alpha, red, green and blue color components. All of the bits in each mask must be contiguous and fit in the specified number of least significant bits of the integer.

Error Index

Error Index **getRedMask**

```
public final int getRedMask()
```

Returns the mask indicating which bits in a pixel contain the red color component.

Error Index **getGreenMask**

```
public final int getGreenMask()
```

Returns the mask indicating which bits in a pixel contain the green color component.

Error Index **getBlueMask**

```
public final int getBlueMask()
```

Returns the mask indicating which bits in a pixel contain the blue color component.

Error Index **getAlphaMask**

```
public final int getAlphaMask()
```

Returns the mask indicating which bits in a pixel contain the alpha transparency component.

Error Index **getRed**

```
public final int getRed(int pixel)
```

Returns the red color component for the specified pixel in the range 0-255.

Overrides:

getRed in class ColorModel

Error Index **getGreen**

public final int getGreen(int pixel)

Returns the green color component for the specified pixel in the range 0-255.

Overrides:

getGreen in class ColorModel

Error Index **getBlue**

public final int getBlue(int pixel)

Returns the blue color component for the specified pixel in the range 0-255.

Overrides:

getBlue in class ColorModel

Error Index **getAlpha**

public final int getAlpha(int pixel)

Return the alpha transparency value for the specified pixel in the range 0-255.

Overrides:

getAlpha in class ColorModel

Error Index **getRGB**

public final int getRGB(int pixel)

Returns the color of the pixel in the default RGB color model.

Overrides:

getRGB in class ColorModel

See Also:

[getRGBdefault](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.FilteredImageSource

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.FilteredImageSource

[java.lang.Object](#)

|

+----java.awt.image.FilteredImageSource

```
public class FilteredImageSource  
    extends Object  
    implements ImageProducer
```

This class is an implementation of the ImageProducer interface which takes an existing image and a filter object and uses them to produce image data for a new filtered version of the original image. Here is an example which filters an image by swapping the red and blue compents:

```
Image src = getImage("doc:///demo/images/duke/T1.gif");  
ImageFilter colorfilter = new RedBlueSwapFilter();  
Image img = createImage(new FilteredImageSource(src.getSource(),  
                                         colorfilter));
```

See Also:

[ImageProducer](#)

Error Index

Error Index

FilteredImageSource(ImageProducer, ImageFilter)

Constructs an ImageProducer object from an existing ImageProducer and a filter object.

Error Index

Error Index

addConsumer(ImageConsumer)

Adds an ImageConsumer to the list of consumers interested in data for this image.

Error Index

isConsumer(ImageConsumer)

Determines whether an ImageConsumer is on the list of consumers currently interested in data for this image.

Error Index

removeConsumer(ImageConsumer)

Removes an ImageConsumer from the list of consumers interested in data for this image.

Error Index

requestTopDownLeftRightResend(ImageConsumer)

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order.

Error Index

startProduction(ImageConsumer)

Adds an ImageConsumer to the list of consumers interested in data for this image, and immediately starts delivery of the image data through the ImageConsumer interface.

Error Index

Error Index

FilteredImageSource

```
public FilteredImageSource(ImageProducer orig,  
                           ImageFilter imgf)
```

Constructs an ImageProducer object from an existing ImageProducer and a filter object.

See Also:

ImageFilter, createImage

Error Index

Error Index

addConsumer

```
public synchronized void addConsumer(ImageConsumer ic)
```

Adds an ImageConsumer to the list of consumers interested in data for this image.

See Also:

ImageConsumer

Error Index isConsumer

public synchronized boolean isConsumer(ImageConsumer ic)

Determines whether an ImageConsumer is on the list of consumers currently interested in data for this image.

Returns:

true if the ImageConsumer is on the list; false otherwise

See Also:

ImageConsumer

Error Index removeConsumer

public synchronized void removeConsumer(ImageConsumer ic)

Removes an ImageConsumer from the list of consumers interested in data for this image.

See Also:

ImageConsumer

Error Index startProduction

public void startProduction(ImageConsumer ic)

Adds an ImageConsumer to the list of consumers interested in data for this image, and immediately starts delivery of the image data through the ImageConsumer interface.

See Also:

ImageConsumer

Error Index requestTopDownLeftRightResend

public void requestTopDownLeftRightResend(ImageConsumer ic)

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order. The request is handed to the ImageFilter for further processing, since the ability to preserve the pixel ordering depends on the filter.

See Also:

ImageConsumer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Image

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Image

[java.lang.Object](#)
|
+----java.awt.Image

public class **Image**
extends [Object](#)

The image class is an abstract class. The image must be obtained in a platform specific way.

Error Index

Error Index

[UndefinedProperty](#)

The UndefinedProperty object should be returned whenever a property which was not defined for a particular image is fetched.

Error Index

Error Index

[Image\(\)](#)

Error Index

Error Index

[flush\(\)](#)

Flushes all resources being used by this Image object.

Error Index

[getGraphics\(\)](#)

Gets a graphics object to draw into this image.

Error Index

getHeight(ImageObserver)

Gets the actual height of the image.

Error Index

getProperty(String, ImageObserver)

Gets a property of the image by name.

Error Index

getSource()

Gets the object that produces the pixels for the image.

Error Index

getWidth(ImageObserver)

Gets the actual width of the image.

Error Index

Error Index

UndefinedProperty

public final static Object UndefinedProperty

The UndefinedProperty object should be returned whenever a property which was not defined for a particular image is fetched.

Error Index

Error Index

Image

public Image()

Error Index

Error Index

getWidth

public abstract int getWidth(ImageObserver observer)

Gets the actual width of the image. If the width is not known yet then the ImageObserver will be notified later and -1 will be returned.

See Also:

getHeight, ImageObserver

Error Index

getHeight

```
public abstract int getHeight(ImageObserver observer)
```

Gets the actual height of the image. If the height is not known yet then the ImageObserver will be notified later and -1 will be returned.

See Also:

getWidth, ImageObserver

Error Index

getSource

```
public abstract ImageProducer getSource()
```

Gets the object that produces the pixels for the image. This is used by the Image filtering classes and by the image conversion and scaling code.

See Also:

ImageProducer

Error Index

getGraphics

```
public abstract Graphics getGraphics()
```

Gets a graphics object to draw into this image. This will only work for off-screen images.

See Also:

Graphics

Error Index

getProperty

```
public abstract Object getProperty(String name,  
                                     ImageObserver observer)
```

Gets a property of the image by name. Individual property names are defined by the various image formats. If a property is not defined for a particular image, this method will return the UndefinedProperty object. If the properties for this image are not yet known, then this method will return null and the ImageObserver object will be notified later. The property name "comment" should be used to store an optional comment which can be presented to the user as a description of the

image, its source, or its author.

See Also:

[ImageObserver](#), [UndefinedProperty](#)

Error Index flush

public abstract void flush()

Flushes all resources being used by this Image object. This includes any pixel data that is being cached for rendering to the screen as well as any system resources that are being used to store data or pixels for the image. The image is reset to a state similar to when it was first created so that if it is again rendered, the image data will have to be recreated or fetched again from its source.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.image.ImageConsumer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.image.ImageConsumer

public interface **ImageConsumer**
extends [Object](#)

The interface for objects expressing interest in image data through the ImageProducer interfaces. When a consumer is added to an image producer, the producer delivers all of the data about the image using the method calls defined in this interface.

See Also:

[ImageProducer](#)

Error Index

Error Index

COMPLETESCANLINES

The pixels will be delivered in (multiples of) complete scanlines at a time.

Error Index

IMAGEABORTED

The image creation process was deliberately aborted.

Error Index

IMAGEERROR

An error was encountered while producing the image.

Error Index

RANDOMPIXELORDER

The pixels will be delivered in a random order.

Error Index

SINGLEFRAME

The image contain a single static image.

Error Index

SINGLEFRAMEDONE

One frame of the image is complete but there are more frames to be delivered.

Error Index

SINGLEPASS

The pixels will be delivered in a single pass.

Error Index

STATICIMAGEDONE

The image is complete and there are no more pixels or frames to be delivered.

Error Index

TOPDOWNLEFTRIGHT

The pixels will be delivered in top-down, left-to-right order.

Error Index

Error Index

imageComplete(int)

The imageComplete method is called when the ImageProducer is finished delivering all of the pixels that the source image contains, or when a single frame of a multi-frame animation has been completed, or when an error in loading or producing the image has occurred.

Error Index

setColorModel(ColorModel)

The ColorModel object used for the majority of the pixels reported using the setPixels method calls.

Error Index

setDimensions(int, int)

The dimensions of the source image are reported using the setDimensions method call.

Error Index

setHints(int)

The ImageProducer can deliver the pixels in any order, but the ImageConsumer may be able to scale or convert the pixels to the destination ColorModel more efficiently or with higher quality if it knows some information about how the pixels will be delivered up front.

Error Index

setPixels (int, int, int, int, ColorModel, byte[], int, int)

The pixels of the image are delivered using one or more calls to the setPixels method.

Error Index

setPixels (int, int, int, int, ColorModel, int[], int, int)

The pixels of the image are delivered using one or more calls to the setPixels method.

Error Index

setProperties(Hashtable)

Sets the extensible list of properties associated with this image.

Error Index

Error Index

RANDOMPIXELORDER

```
public final static int RANDOMPIXELORDER
```

The pixels will be delivered in a random order. This tells the ImageConsumer not to use any optimizations that depend on the order of pixel delivery, which should be the default assumption in the absence of any call to the setHints method.

See Also:

setHints

Error Index TOPDOWNLEFTRIGHT

public final static int TOPDOWNLEFTRIGHT

The pixels will be delivered in top-down, left-to-right order.

See Also:

[setHints](#)

Error Index COMPLETESCANLINES

public final static int COMPLETESCANLINES

The pixels will be delivered in (multiples of) complete scanlines at a time.

See Also:

[setHints](#)

Error Index SINGLEPASS

public final static int SINGLEPASS

The pixels will be delivered in a single pass. Each pixel will appear in only one call to any of the setPixels methods. An example of an image format which does not meet this criterion is a progressive JPEG image which defines pixels in multiple passes, each more refined than the previous.

See Also:

[setHints](#)

Error Index SINGLEFRAME

public final static int SINGLEFRAME

The image contain a single static image. The pixels will be defined in calls to the setPixels methods and then the imageComplete method will be called with the STATICIMAGEDONE flag after which no more image data will be delivered. An example of an image type which would not meet these criteria would be the output of a video feed, or the representation of a 3D rendering being manipulated by the user. The end of each frame in those types of images will be indicated by calling imageComplete with the SINGLEFRAMEDONE flag.

See Also:

[setHints](#), [imageComplete](#)

Error Index **IMAGEERROR**

public final static int IMAGEERROR

An error was encountered while producing the image.

See Also:

[imageComplete](#)

Error Index **SINGLEFRAMEDONE**

public final static int SINGLEFRAMEDONE

One frame of the image is complete but there are more frames to be delivered.

See Also:

[imageComplete](#)

Error Index **STATICIMAGEDONE**

public final static int STATICIMAGEDONE

The image is complete and there are no more pixels or frames to be delivered.

See Also:

[imageComplete](#)

Error Index **IMAGEABORTED**

public final static int IMAGEABORTED

The image creation process was deliberately aborted.

See Also:

[imageComplete](#)

Error Index

Error Index

setDimensions

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

The dimensions of the source image are reported using the setDimensions method call.

Error Index

setProperties

```
public abstract void setProperties(Hashtable props)
```

Sets the extensible list of properties associated with this image.

Error Index

setColorModel

```
public abstract void setColorModel(ColorModel model)
```

The ColorModel object used for the majority of the pixels reported using the setPixels method calls. Note that each set of pixels delivered using setPixels contains its own ColorModel object, so no assumption should be made that this model will be the only one used in delivering pixel values. A notable case where multiple ColorModel objects may be seen is a filtered image when for each set of pixels that it filters, the filter determines whether the pixels can be sent on untouched, using the original ColorModel, or whether the pixels should be modified (filtered) and passed on using a ColorModel more convenient for the filtering process.

See Also:

ColorModel

Error Index

setHints

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

The ImageProducer can deliver the pixels in any order, but the ImageConsumer may be able to scale or convert the pixels to the destination ColorModel more efficiently or with higher quality if it knows some information about how the pixels will be delivered up front. The setHints method should be called before any calls to any of the setPixels methods with a bit mask of hints about the manner in which the pixels will be delivered. If the ImageProducer does not follow the guidelines for the indicated hint, the results are undefined.

Error Index **setPixels**

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

The pixels of the image are delivered using one or more calls to the setPixels method. Each call specifies the location and size of the rectangle of source pixels that are contained in the array of pixels. The specified ColorModel object should be used to convert the pixels into their corresponding color and alpha components. Pixel (m,n) is stored in the pixels array at index (n * scansize + m + off). The pixels delivered using this method are all stored as bytes.

See Also:

ColorModel

Error Index **setPixels**

```
public abstract void setPixels(int x,  
    int y,  
    int w,  
    int h,  
    ColorModel model,  
    int pixels[],  
    int off,  
    int scansize)
```

The pixels of the image are delivered using one or more calls to the setPixels method. Each call specifies the location and size of the rectangle of source pixels that are contained in the array of pixels. The specified ColorModel object should be used to convert the pixels into their corresponding color and alpha components. Pixel (m,n) is stored in the pixels array at index (n * scansize + m + off). The pixels delivered using this method are all stored as ints.

See Also:

ColorModel

Error Index **imageComplete**

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

The imageComplete method is called when the ImageProducer is finished delivering all of the pixels that the source image contains, or when a single frame of a multi-frame animation has been

completed, or when an error in loading or producing the image has occurred. The ImageConsumer should remove itself from the list of consumers registered with the ImageProducer at this time, unless it is interested in successive frames.

See Also:

[removeConsumer](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.ImageFilter

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.ImageFilter

[java.lang.Object](#)

|

+----java.awt.image.ImageFilter

```
public class ImageFilter
  extends Object
  implements ImageConsumer, Cloneable
```

This class implements a filter for the set of interface methods that are used to deliver data from an ImageProducer to an ImageConsumer. It is meant to be used in conjunction with a FilteredImageSource object to produce filtered versions of existing images. It is a base class that provides the calls needed to implement a "Null filter" which has no effect on the data being passed through. Filters should subclass this class and override the methods which deal with the data that needs to be filtered and modify it as necessary.

See Also:

[FilteredImageSource](#), [ImageConsumer](#)

Error Index

Error Index

consumer

The consumer of the particular image data stream for which this instance of the ImageFilter is filtering data.

Error Index

Error Index

ImageFilter()

Error Index

Error Index

Clones this object.

clone()

Error Index

Returns a unique instance of an ImageFilter object which will actually perform the filtering for the specified ImageConsumer.

getFilterInstance(ImageConsumer)

Error Index

Filters the information provided in the imageComplete method of the ImageConsumer interface.

imageComplete(int)

Error Index

Responds to a request for a TopDownLeftRight (TDLR) ordered resend of the pixel data from an ImageConsumer.

resendTopDownLeftRight(ImageProducer)

Error Index

Filter the information provided in the setColorModel method of the ImageConsumer interface.

setColorModel(ColorModel)

Error Index

Filters the information provided in the setDimensions method of the ImageConsumer interface.

setDimensions(int, int)

Error Index

Filters the information provided in the setHints method of the ImageConsumer interface.

setHints(int)

Error Index

Filters the information provided in the setPixels method of the ImageConsumer interface which takes an array of bytes.

setPixels (int, int, int, int, ColorModel, byte[], int, int)

Error Index

Filters the information provided in the setPixels method of the ImageConsumer interface which takes an array of integers.

setPixels (int, int, int, int, ColorModel, int[], int, int)

Error Index

Passes the properties from the source object along after adding a property indicating the stream of filters it has been run through.

setProperties(Hashtable)

Error Index

Error Index

consumer

protected ImageConsumer consumer

The consumer of the particular image data stream for which this instance of the ImageFilter is

filtering data. It is not initialized during the constructor, but rather during the `getFilterInstance()` method call when the `FilteredImageSource` is creating a unique instance of this object for a particular image data stream.

See Also:

[getFilterInstance](#), [ImageConsumer](#)

Error Index

Error Index

ImageFilter

```
public ImageFilter()
```

Error Index

Error Index

getFilterInstance

```
public ImageFilter getFilterInstance(ImageConsumer ic)
```

Returns a unique instance of an `ImageFilter` object which will actually perform the filtering for the specified `ImageConsumer`. The default implementation just clones this object.

Error Index

setDimensions

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

Filters the information provided in the `setDimensions` method of the `ImageConsumer` interface.

See Also:

[setDimensions](#)

Error Index

setProperties

```
public void setProperties(Hashtable props)
```

Passes the properties from the source object along after adding a property indicating the stream of filters it has been run through.

Error Index **setColorModel**

```
public void setColorModel(ColorModel model)
```

Filter the information provided in the setColorModel method of the ImageConsumer interface.

See Also:

setColorModel

Error Index **setHints**

```
public void setHints(int hints)
```

Filters the information provided in the setHints method of the ImageConsumer interface.

See Also:

setHints

Error Index **setPixels**

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

Filters the information provided in the setPixels method of the ImageConsumer interface which takes an array of bytes.

See Also:

setPixels

Error Index **setPixels**

```
public void setPixels(int x,  
    int y,  
    int w,  
    int h,  
    ColorModel model,  
    int pixels[],
```

```
int off,  
int scansize)
```

Filters the information provided in the setPixels method of the ImageConsumer interface which takes an array of integers.

See Also:

[setPixels](#)

Error Index

imageComplete

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

Filters the information provided in the imageComplete method of the ImageConsumer interface.

See Also:

[imageComplete](#)

Error Index

resendTopDownLeftRight

```
public void resendTopDownLeftRight(ImageProducer ip)
```

Responds to a request for a TopDownLeftRight (TDLR) ordered resend of the pixel data from an ImageConsumer. The ImageFilter can respond to this request in one of three ways.

1. If the filter can determine that it will forward the pixels in TDLR order if its upstream producer object sends them in TDLR order, then the request is automatically forwarded by default to the indicated ImageProducer using this filter as the requesting ImageConsumer, so no override is necessary.
2. If the filter can resend the pixels in the right order on its own (presumably because the generated pixels have been saved in some sort of buffer), then it can override this method and simply resend the pixels in TDLR order as specified in the ImageProducer API.
3. If the filter simply returns from this method then the request will be ignored and no resend will occur.

@see ImageProducer#requestTopDownLeftRightResend

Parameters:

ip - The ImageProducer that is feeding this instance of the filter - also the ImageProducer that the request should be forwarded to if necessary.

Error Index

clone

```
public Object clone()
```


Clones this object.

Overrides:

clone in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.image.ImageObserver

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.image.ImageObserver

public interface **ImageObserver**
extends [Object](#)

An asynchronous update interface for receiving notifications about Image information as the Image is constructed.

Error Index

Error Index

ABORT

An image which was being tracked asynchronously was aborted before production was complete.

Error Index

ALLBITS

A static image which was previously drawn is now complete and can be drawn again in its final form.

Error Index

ERROR

An image which was being tracked asynchronously has encountered an error.

Error Index

FRAMEBITS

Another complete frame of a multi-frame image which was previously drawn is now available to be drawn again.

Error Index

HEIGHT

The height of the base image is now available and can be taken from the height argument to the imageUpdate callback method.

Error Index

PROPERTIES

The properties of the image are now available.

Error Index

SOMEBITS

More pixels needed for drawing a scaled variation of the image are available.

Error Index

WIDTH

The width of the base image is now available and can be taken from the width argument to the imageUpdate callback method.

Error Index

Error Index

imageUpdate(Image, int, int, int, int, int)

This method is called when information about an image which was previously requested using an asynchronous interface becomes available.

Error Index

Error Index WIDTH

public final static int WIDTH

The width of the base image is now available and can be taken from the width argument to the imageUpdate callback method.

See Also:

getWidth, imageUpdate

Error Index HEIGHT

public final static int HEIGHT

The height of the base image is now available and can be taken from the height argument to the imageUpdate callback method.

See Also:

getHeight, imageUpdate

Error Index PROPERTIES

public final static int PROPERTIES

The properties of the image are now available.

See Also:

getProperty, imageUpdate

Error Index **SOMEBITS**

public final static int SOMEBITS

More pixels needed for drawing a scaled variation of the image are available. The bounding box of the new pixels can be taken from the x, y, width, and height arguments to the imageUpdate callback method.

See Also:

[drawImage](#), [imageUpdate](#)

Error Index **FRAMEBITS**

public final static int FRAMEBITS

Another complete frame of a multi-frame image which was previously drawn is now available to be drawn again. The x, y, width, and height arguments to the imageUpdate callback method should be ignored.

See Also:

[drawImage](#), [imageUpdate](#)

Error Index **ALLBITS**

public final static int ALLBITS

A static image which was previously drawn is now complete and can be drawn again in its final form. The x, y, width, and height arguments to the imageUpdate callback method should be ignored.

See Also:

[drawImage](#), [imageUpdate](#)

Error Index **ERROR**

public final static int ERROR

An image which was being tracked asynchronously has encountered an error. No further information will become available and drawing the image will fail. As a convenience, the ABORT flag will be indicated at the same time to indicate that the image production was aborted.

See Also:

[imageUpdate](#)

Error Index ABORT

public final static int ABORT

An image which was being tracked asynchronously was aborted before production was complete. No more information will become available without further action to trigger another image production sequence. If the ERROR flag was not also set in this image update, then accessing any of the data in the image will restart the production again, probably from the beginning.

See Also:

[imageUpdate](#)

Error Index

Error Index imageUpdate

```
public abstract boolean imageUpdate(Image img,  
                                     int infoflags,  
                                     int x,  
                                     int y,  
                                     int width,  
                                     int height)
```

This method is called when information about an image which was previously requested using an asynchronous interface becomes available. Asynchronous interfaces are method calls such as `getWidth(ImageObserver)` and `drawImage(img, x, y, ImageObserver)` which take an `ImageObserver` object as an argument. Those methods register the caller as interested either in information about the overall image itself (in the case of `getWidth(ImageObserver)`) or about an output version of an image (in the case of the `drawImage(img, x, y, [w, h,] ImageObserver)` call).

This method should return true if further updates are needed or false if the required information has been acquired. The image which was being tracked is passed in using the `img` argument. Various constants are combined to form the `infoflags` argument which indicates what information about the image is now available. The interpretation of the `x`, `y`, `width`, and `height` arguments depends on the contents of the `infoflags` argument.

See Also:

[getWidth](#), [getHeight](#), [drawImage](#)

Interface java.awt.image.ImageProducer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.image.ImageProducer

public interface **ImageProducer**
extends [Object](#)

The interface for objects which can produce the image data for Images. Each image contains an ImageProducer which is used to reconstruct the image whenever it is needed, for example, when a new size of the Image is scaled, or when the width or height of the Image is being requested.

See Also:

[ImageConsumer](#)

Error Index

Error Index

[addConsumer](#)(ImageConsumer)

This method is used to register an ImageConsumer with the ImageProducer for access to the image data during a later reconstruction of the Image.

Error Index

[isConsumer](#)(ImageConsumer)

This method determines if a given ImageConsumer object is currently registered with this ImageProducer as one of its consumers.

Error Index

[removeConsumer](#)(ImageConsumer)

This method removes the given ImageConsumer object from the list of consumers currently registered to receive image data.

Error Index

[requestTopDownLeftRightResend](#)(ImageConsumer)

This method is used by an ImageConsumer to request that the ImageProducer attempt to resend the image data one more time in TOPDOWNLEFTRIGHT order so that higher quality conversion algorithms which depend on receiving pixels in order can be used to produce a better output version of the image.

Error Index

[startProduction](#)(ImageConsumer)

This method both registers the given ImageConsumer object as a consumer and starts an immediate reconstruction of the image data which will then be delivered to this consumer and any other consumer which may have already been registered with the producer.

Error Index

Error Index **addConsumer**

public abstract void addConsumer(ImageConsumer ic)

This method is used to register an ImageConsumer with the ImageProducer for access to the image data during a later reconstruction of the Image. The ImageProducer may, at its discretion, start delivering the image data to the consumer using the ImageConsumer interface immediately, or when the next available image reconstruction is triggered by a call to the startProduction method.

See Also:

startProduction

Error Index **isConsumer**

public abstract boolean isConsumer(ImageConsumer ic)

This method determines if a given ImageConsumer object is currently registered with this ImageProducer as one of its consumers.

Error Index **removeConsumer**

public abstract void removeConsumer(ImageConsumer ic)

This method removes the given ImageConsumer object from the list of consumers currently registered to receive image data. It is not considered an error to remove a consumer that is not currently registered. The ImageProducer should stop sending data to this consumer as soon as is feasible.

Error Index **startProduction**

public abstract void startProduction(ImageConsumer ic)

This method both registers the given ImageConsumer object as a consumer and starts an immediate reconstruction of the image data which will then be delivered to this consumer and any other consumer which may have already been registered with the producer. This method differs from the addConsumer method in that a reproduction of the image data should be triggered as soon as possible.

See Also:

addConsumer

Error Index

requestTopDownLeftRightResend

public abstract void requestTopDownLeftRightResend(ImageConsumer ic)

This method is used by an ImageConsumer to request that the ImageProducer attempt to resend the image data one more time in TOPDOWNLEFTRIGHT order so that higher quality conversion algorithms which depend on receiving pixels in order can be used to produce a better output version of the image. The ImageProducer is free to ignore this call if it cannot resend the data in that order. If the data can be resent, then the ImageProducer should respond by executing the following minimum set of ImageConsumer method calls:

```
ic.setHints(TOPDOWNLEFTRIGHT | );  
ic.setPixels(...);    // As many times as needed  
ic.imageComplete();
```

See Also:

[setHints](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.IndexColorModel

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.IndexColorModel

[java.lang.Object](#)

```
|
+----java.awt.image.ColorModel
|
+----java.awt.image.IndexColorModel
```

public class **IndexColorModel**
extends [ColorModel](#)

A ColorModel class that specifies a translation from pixel values to alpha, red, green, and blue color components for pixels which represent indices into a fixed colormap. An optional transparent pixel value can be supplied which indicates a completely transparent pixel, regardless of any alpha value recorded for that pixel value. This color model is similar to an X11 PseudoColor visual.

Many of the methods in this class are final. The reason for this is that the underlying native graphics code makes assumptions about the layout and operation of this class and those assumptions are reflected in the implementations of the methods here that are marked final. You can subclass this class for other reasons, but you cannot override or modify the behaviour of those methods.

See Also:

[ColorModel](#)

Error Index

Error Index

[IndexColorModel](#)(int, int, byte[], byte[], byte[])

Constructs an IndexColorModel from the given arrays of red, green, and blue components.

Error Index

[IndexColorModel](#) (int, int, byte[], byte[], byte[], int)

Constructs an IndexColorModel from the given arrays of red, green, and blue components.

Error Index

[IndexColorModel](#) (int, int, byte[], byte[], byte[], byte[])

Constructs an IndexColorModel from the given arrays of red, green, blue and alpha components.

Error Index

[IndexColorModel](#)(int, int, byte[], int, boolean)

Constructs an IndexColorModel from a single arrays of packed red, green, blue and optional alpha

components.

Error Index

IndexColorModel(int, int, byte[], int, boolean, int)

Constructs an IndexColorModel from a single arrays of packed red, green, blue and optional alpha components.

Error Index

Error Index

getAlpha(int)

Returns the alpha transparency value for the specified pixel in the range 0-255.

Error Index

getAlphas(byte[])

Copies the array of alpha transparency values into the given array.

Error Index

getBlue(int)

Returns the blue color component for the specified pixel in the range 0-255.

Error Index

getBlues(byte[])

Copies the array of blue color components into the given array.

Error Index

getGreen(int)

Returns the green color component for the specified pixel in the range 0-255.

Error Index

getGreens(byte[])

Copies the array of green color components into the given array.

Error Index

getMapSize()

Returns the size of the color component arrays in this IndexColorModel.

Error Index

getRGB(int)

Returns the color of the pixel in the default RGB color model.

Error Index

getRed(int)

Returns the red color component for the specified pixel in the range 0-255.

Error Index

getReds(byte[])

Copies the array of red color components into the given array.

Error Index

getTransparentPixel()

Returns the index of the transparent pixel in this IndexColorModel or -1 if there is no transparent pixel.

Error Index

Error Index

IndexColorModel

```
public IndexColorModel(int bits,  
    int size,  
    byte r[],  
    byte g[],  
    byte b[])
```

Constructs an IndexColorModel from the given arrays of red, green, and blue components. Pixels described by this color model will all have alpha components of 255 (fully opaque). All of the arrays specifying the color components must have at least the specified number of entries.

Parameters:

- bits - The number of bits each pixel occupies.
- size - The size of the color component arrays.
- r - The array of red color components.
- g - The array of green color components.
- b - The array of blue color components.

Error Index

IndexColorModel

```
public IndexColorModel(int bits,  
    int size,  
    byte r[],  
    byte g[],  
    byte b[],  
    int trans)
```

Constructs an IndexColorModel from the given arrays of red, green, and blue components. Pixels described by this color model will all have alpha components of 255 (fully opaque), except for the indicated transparent pixel. All of the arrays specifying the color components must have at least the specified number of entries.

Parameters:

- bits - The number of bits each pixel occupies.
- size - The size of the color component arrays.
- r - The array of red color components.
- g - The array of green color components.
- b - The array of blue color components.
- trans - The index of the transparent pixel.

Error Index

IndexColorModel

```
public IndexColorModel(int bits,
    int size,
    byte r[],
    byte g[],
    byte b[],
    byte a[])
```

Constructs an IndexColorModel from the given arrays of red, green, blue and alpha components. All of the arrays specifying the color components must have at least the specified number of entries.

Parameters:

bits - The number of bits each pixel occupies.
 size - The size of the color component arrays.
 r - The array of red color components.
 g - The array of green color components.
 b - The array of blue color components.
 a - The array of alpha value components.

Error Index IndexColorModel

```
public IndexColorModel(int bits,
    int size,
    byte cmap[],
    int start,
    boolean hasalpha)
```

Constructs an IndexColorModel from a single arrays of packed red, green, blue and optional alpha components. The array must have enough values in it to fill all of the needed component arrays of the specified size.

Parameters:

bits - The number of bits each pixel occupies.
 size - The size of the color component arrays.
 cmap - The array of color components.
 start - The starting offset of the first color component.
 hasalpha - Indicates whether alpha values are contained in the cmap array.

Error Index IndexColorModel

```
public IndexColorModel(int bits,
    int size,
    byte cmap[],
    int start,
    boolean hasalpha,
    int trans)
```

Constructs an IndexColorModel from a single arrays of packed red, green, blue and optional alpha

components. The specified transparent index represents a pixel which will be considered entirely transparent regardless of any alpha value specified for it. The array must have enough values in it to fill all of the needed component arrays of the specified size.

Parameters:

bits - The number of bits each pixel occupies.

size - The size of the color component arrays.

cmap - The array of color components.

start - The starting offset of the first color component.

hasalpha - Indicates whether alpha values are contained in the cmap array.

trans - The index of the fully transparent pixel.

Error Index

Error Index

getMapSize

```
public final int getMapSize()
```

Returns the size of the color component arrays in this IndexColorModel.

Error Index

getTransparentPixel

```
public final int getTransparentPixel()
```

Returns the index of the transparent pixel in this IndexColorModel or -1 if there is no transparent pixel.

Error Index

getReds

```
public final void getReds(byte r[])
```

Copies the array of red color components into the given array. Only the initial entries of the array as specified by getMapSize() are written.

Error Index

getGreens

```
public final void getGreens(byte g[])
```

Copies the array of green color components into the given array. Only the initial entries of the array as specified by getMapSize() are written.

Error Index getBlues

```
public final void getBlues(byte b[])
```

Copies the array of blue color components into the given array. Only the initial entries of the array as specified by getMapSize() will be written.

Error Index getAlphas

```
public final void getAlphas(byte a[])
```

Copies the array of alpha transparency values into the given array. Only the initial entries of the array as specified by getMapSize() will be written.

Error Index getRed

```
public final int getRed(int pixel)
```

Returns the red color component for the specified pixel in the range 0-255.

Overrides:

getRed in class ColorModel

Error Index getGreen

```
public final int getGreen(int pixel)
```

Returns the green color component for the specified pixel in the range 0-255.

Overrides:

getGreen in class ColorModel

Error Index getBlue

```
public final int getBlue(int pixel)
```

Returns the blue color component for the specified pixel in the range 0-255.

Overrides:

getBlue in class ColorModel

Error Index `getAlpha`

```
public final int getAlpha(int pixel)
```

Returns the alpha transparency value for the specified pixel in the range 0-255.

Overrides:

`getAlpha` in class `ColorModel`

Error Index `getRGB`

```
public final int getRGB(int pixel)
```

Returns the color of the pixel in the default RGB color model.

Overrides:

`getRGB` in class `ColorModel`

See Also:

`getRGBdefault`

Class java.awt.image.MemoryImageSource

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.MemoryImageSource

[java.lang.Object](#)

|

+----java.awt.image.MemoryImageSource

```
public class MemoryImageSource
    extends Object
    implements ImageProducer
```

This class is an implementation of the ImageProducer interface which uses an array to produce pixel values for an Image. Here is an example which calculates a 100x100 image representing a fade from black to blue along the X axis and a fade from black to red along the Y axis:

```
int w = 100;
int h = 100;
int pix[] = new int[w * h];
int index = 0;
for (int y = 0; y
```

See Also:

[ImageProducer](#)

Error Index

Error Index

MemoryImageSource(int, int, ColorModel, byte[], int, int)

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

Error Index

MemoryImageSource(int, int, ColorModel, byte[], int, int, Hashtable)

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

Error Index

MemoryImageSource(int, int, ColorModel, int[], int, int)

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

Error Index

MemoryImageSource(int, int, ColorModel, int[], int, int, Hashtable)

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

Error Index

MemoryImageSource(int, int, int[], int, int)

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

Error Index

MemoryImageSource(int, int, int[], int, int, Hashtable)

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

Error Index

Error Index

addConsumer(ImageConsumer)

Adds an ImageConsumer to the list of consumers interested in data for this image.

Error Index

isConsumer(ImageConsumer)

Determine if an ImageConsumer is on the list of consumers currently interested in data for this image.

Error Index

removeConsumer(ImageConsumer)

Remove an ImageConsumer from the list of consumers interested in data for this image.

Error Index

requestTopDownLeftRightResend(ImageConsumer)

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order.

Error Index

startProduction(ImageConsumer)

Adds an ImageConsumer to the list of consumers interested in data for this image, and immediately start delivery of the image data through the ImageConsumer interface.

Error Index

Error Index

MemoryImageSource

```
public MemoryImageSource(int w,  
    int h,  
    ColorModel cm,  
    byte pix[],  
    int off,  
    int scan)
```

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

See Also:

[createImage](#)

Error Index

MemoryImageSource

```
public MemoryImageSource(int w,  
    int h,  
    ColorModel cm,  
    byte pix[],  
    int off,  
    int scan,  
    Hashtable props)
```

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

See Also:

[createImage](#)

Error Index

MemoryImageSource

```
public MemoryImageSource(int w,  
    int h,  
    ColorModel cm,  
    int pix[],  
    int off,  
    int scan)
```

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

See Also:

[createImage](#)

Error Index

MemoryImageSource

```
public MemoryImageSource(int w,  
    int h,  
    ColorModel cm,  
    int pix[],  
    int off,  
    int scan,  
    Hashtable props)
```

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

See Also:

[createImage](#)

Error Index

MemoryImageSource

```
public MemoryImageSource(int w,  
    int h,  
    int pix[],  
    int off,  
    int scan)
```

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

See Also:

[createImage](#), [getRGBdefault](#)

Error Index

MemoryImageSource

```
public MemoryImageSource(int w,  
    int h,  
    int pix[],  
    int off,  
    int scan,  
    Hashtable props)
```

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

See Also:

[createImage](#), [getRGBdefault](#)

Error Index

Error Index

addConsumer

```
public synchronized void addConsumer(ImageConsumer ic)
```

Adds an ImageConsumer to the list of consumers interested in data for this image.

See Also:

[ImageConsumer](#)

Error Index

isConsumer

public synchronized boolean isConsumer(ImageConsumer ic)

Determine if an ImageConsumer is on the list of consumers currently interested in data for this image.

Returns:

true if the ImageConsumer is on the list; false otherwise

See Also:

ImageConsumer

Error Index

removeConsumer

public synchronized void removeConsumer(ImageConsumer ic)

Remove an ImageConsumer from the list of consumers interested in data for this image.

See Also:

ImageConsumer

Error Index

startProduction

public void startProduction(ImageConsumer ic)

Adds an ImageConsumer to the list of consumers interested in data for this image, and immediately start delivery of the image data through the ImageConsumer interface.

See Also:

[ImageConsumer](#)

Error Index

requestTopDownLeftRightResend

```
public void requestTopDownLeftRightResend(ImageConsumer ic)
```

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order.

See Also:

[ImageConsumer](#)

Class java.awt.image.PixelGrabber

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.PixelGrabber

[java.lang.Object](#)

|
+----java.awt.image.PixelGrabber

```
public class PixelGrabber  
extends Object  
implements ImageConsumer
```

The PixelGrabber class implements an ImageConsumer which can be attached to an Image or ImageProducer object to retrieve a subset of the pixels in that image. Here is an example:

```
public abstract void handlesinglepixel(int x, int y, int pixel);  
public void handlepixels(Image img, int x, int y, int w, int h) {  
    int[] pixels = new int[w * h];  
    PixelGrabber pg = new PixelGrabber(img, x, y, w, h, pixels, 0, w);  
    try {  
        pg.grabPixels();  
    } catch (InterruptedException e) {  
        System.err.println("interrupted waiting for pixels!");  
        return;  
    }  
    if ((pg.status() & ImageObserver.ABORT) != 0) {  
        System.err.println("image fetch aborted or errored");  
        return;  
    }  
    for (int j = 0; j
```

Error Index

Error Index

PixelGrabber(Image, int, int, int, int, int[], int, int)

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the specified image into the given array.

Error Index

PixelGrabber(ImageProducer, int, int, int, int, int[], int, int)

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the image produced by the specified ImageProducer into the given array.

Error Index

Error Index

grabPixels()

Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered.

Error Index

grabPixels(long)

Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered or until the specified timeout has elapsed.

Error Index

imageComplete(int)

The imageComplete method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setColorModel(ColorModel)

The setColorModel method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setDimensions(int, int)

The setDimensions method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setHints(int)

The setHints method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setPixels(int, int, int, int, ColorModel, byte[], int, int)

The setPixels method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setPixels(int, int, int, int, ColorModel, int[], int, int)

The setPixels method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setProperties(Hashtable)

The setProperties method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

status()

Return the status of the pixels.

Error Index

Error Index

PixelGrabber

```
public PixelGrabber(Image img,  
    int x,  
    int y,  
    int w,  
    int h,  
    int pix[],  
    int off,  
    int scansize)
```

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the specified image into the given array. The pixels are stored into the array in the default RGB ColorModel. The RGB data for pixel (i, j) where (i, j) is inside the rectangle (x, y, w, h) is stored in the array at `pix[(j - y) * scansize + (i - x) + off]`.

Parameters:

img - the image to retrieve pixels from

x - the x coordinate of the upper left corner of the rectangle of pixels to retrieve from the image, relative to the default (unscaled) size of the image

y - the y coordinate of the upper left corner of the rectangle of pixels to retrieve from the image

w - the width of the rectangle of pixels to retrieve

h - the height of the rectangle of pixels to retrieve

pix - the array of integers which are to be used to hold the RGB pixels retrieved from the image

off - the offset into the array of where to store the first pixel

scansize - the distance from one row of pixels to the next in the array

See Also:

[getRGBdefault](#)

Error Index

PixelGrabber

```
public PixelGrabber(ImageProducer ip,  
    int x,  
    int y,  
    int w,  
    int h,  
    int pix[],  
    int off,  
    int scansize)
```

Create a PixelGrabber object to grab the (x, y, w, h) rectangular section of pixels from the image produced by the specified ImageProducer into the given array.

The pixels are stored into the array in the default RGB ColorModel. The RGB data for pixel (i, j) where (i, j) is inside the rectangle (x, y, w, h) is stored in the array at `pix[(j - y) * scansize + (i - x) + off]`.

Parameters:

img - the image to retrieve pixels from

x - the x coordinate of the upper left corner of the rectangle of pixels to retrieve from the image, relative to the default (unscaled) size of the image

y - the y coordinate of the upper left corner of the rectangle of pixels to retrieve from the image

w - the width of the rectangle of pixels to retrieve

h - the height of the rectangle of pixels to retrieve

pix - the array of integers which are to be used to hold the RGB pixels retrieved from the image

off - the offset into the array of where to store the first pixel

scansize - the distance from one row of pixels to the next in the array

See Also:

[getRGBdefault](#)

Error Index

Error Index

grabPixels

public boolean grabPixels() throws [InterruptedException](#)

Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered.

Returns:

true if the pixels were successfully grabbed, false on abort, error or timeout

Throws: [InterruptedException](#)

Another thread has interrupted this thread.

Error Index

grabPixels

public synchronized boolean grabPixels(long ms) throws InterruptedException

Request the Image or ImageProducer to start delivering pixels and wait for all of the pixels in the rectangle of interest to be delivered or until the specified timeout has elapsed.

Parameters:

ms - the number of milliseconds to wait for the image pixels to arrive before timing out

Returns:

true if the pixels were successfully grabbed, false on abort, error or timeout

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index

status

public synchronized int status()

Return the status of the pixels. The ImageObserver flags representing the available pixel information are returned.

Returns:

the bitwise OR of all relevant ImageObserver flags

See Also:

ImageObserver

Error Index

setDimensions

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

The setDimensions method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setHints

```
public void setHints(int hints)
```

The setHints method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setProperties

```
public void setProperties(Hashtable props)
```

The setProperties method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setColorModel

```
public void setColorModel(ColorModel model)
```

The setColorModel method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setPixels

```
public void setPixels(int srcX,  
    int srcY,  
    int srcW,  
    int srcH,  
    ColorModel model,  
    byte pixels[],  
    int srcOff,  
    int srcScan)
```

The setPixels method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

setPixels

```
public void setPixels(int srcX,  
    int srcY,
```

```
int srcW,  
int srcH,  
ColorModel model,  
int pixels[],  
int srcOff,  
int srcScan)
```

The setPixels method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

Error Index

imageComplete

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

The imageComplete method is part of the ImageConsumer API which this class must implement to retrieve the pixels.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.RGBImageFilter

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.image.RGBImageFilter

```
java.lang.Object
|
+----java.awt.image.ImageFilter
|
+----java.awt.image.RGBImageFilter
```

```
public class RGBImageFilter
extends ImageFilter
```

This class provides an easy way to create an `ImageFilter` which modifies the pixels of an image in the default RGB `ColorModel`. It is meant to be used in conjunction with a `FilteredImageSource` object to produce filtered versions of existing images. It is an abstract class that provides the calls needed to channel all of the pixel data through a single method which converts pixels one at a time in the default RGB `ColorModel` regardless of the `ColorModel` being used by the `ImageProducer`. The only method which needs to be defined to create a useable image filter is the `filterRGB` method. Here is an example of a definition of a filter which swaps the red and blue components of an image:

```
class RedBlueSwapFilter extends RGBImageFilter {
    public RedBlueSwapFilter() {
        // The filter's operation does not depend on the
        // pixel's location, so IndexColorModels can be
        // filtered directly.
        canFilterIndexColorModel = true;
    }
    public int filterRGB(int x, int y, int rgb) {
        return ((rgb & 0xff00ff00)
            | ((rgb & 0xff0000) >> 16)
            | (rgb & 0xff)
```

See Also:

[FilteredImageSource](#), [ImageFilter](#), [getRGBdefault](#)

Error Index

Error Index

canFilterIndexColorModel

This boolean indicates whether or not it is acceptable to apply the color filtering of the filterRGB method to the color table entries of an IndexColorModel object in lieu of pixel by pixel filtering.

Error Index

newmodel

Error Index

origmodel

Error Index

Error Index

RGBImageFilter()

Error Index

Error Index

filterIndexColorModel(IndexColorModel)

Filters an IndexColorModel object by running each entry in its color tables through the filterRGB function that RGBImageFilter subclasses must provide.

Error Index

filterRGB(int, int, int)

Subclasses must specify a method to convert a single input pixel in the default RGB ColorModel to a single output pixel.

Error Index

filterRGBPixels(int, int, int, int, int[], int, int)

Filters a buffer of pixels in the default RGB ColorModel by passing them one by one through the filterRGB method.

Error Index

setColorModel(ColorModel)

If the ColorModel is an IndexColorModel, and the subclass has set the canFilterIndexColorModel flag to true, we substitute a filtered version of the color model here and wherever that original ColorModel object appears in the setPixels methods.

Error Index

setPixels(int, int, int, int, ColorModel, byte[], int, int)

If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel.

Error Index

setPixels(int, int, int, int, ColorModel, int[], int, int)

If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel, otherwise converts the buffer of integer pixels to the default RGB ColorModel and passes the converted buffer to the filterRGBPixels method to be converted one by one.

Error Index

substituteColorModel(ColorModel, ColorModel)

Registers two ColorModel objects for substitution.

Error Index

Error Index

origmodel

protected ColorModel origmodel

Error Index

newmodel

protected ColorModel newmodel

Error Index

canFilterIndexColorModel

protected boolean canFilterIndexColorModel

This boolean indicates whether or not it is acceptable to apply the color filtering of the filterRGB method to the color table entries of an IndexColorModel object in lieu of pixel by pixel filtering. Subclasses should set this variable to true in their constructor if their filterRGB method does not depend on the coordinate of the pixel being filtered.

See Also:

[substituteColorModel](#), [filterRGB](#), [IndexColorModel](#)

Error Index

Error Index

RGBImageFilter

public RGBImageFilter()

Error Index

Error Index

setColorModel

public void setColorModel([ColorModel](#) model)

If the ColorModel is an IndexColorModel, and the subclass has set the canFilterIndexColorModel flag to true, we substitute a filtered version of the color model here and wherever that original ColorModel object appears in the setPixels methods. Otherwise overrides the default ColorModel used by the ImageProducer and specifies the default RGB ColorModel instead.

Overrides:

setColorModel in class ImageFilter

See Also:

ImageConsumer, getRGBdefault

Error Index

substituteColorModel

```
public void substituteColorModel(ColorModel oldcm,  
                                ColorModel newcm)
```

Registers two ColorModel objects for substitution. If the oldcm is encountered during any of the setPixels methods, the newcm is substituted and the pixels passed through untouched (but with the new ColorModel object).

Parameters:

oldcm - the ColorModel object to be replaced on the fly

newcm - the ColorModel object to replace oldcm on the fly

Error Index

filterIndexColorModel

```
public IndexColorModel filterIndexColorModel(IndexColorModel icm)
```

Filters an IndexColorModel object by running each entry in its color tables through the filterRGB function that RGBImageFilter subclasses must provide. Uses coordinates of -1 to indicate that a color table entry is being filtered rather than an actual pixel value.

Parameters:

icm - the IndexColorModel object to be filtered

Returns:

a new IndexColorModel representing the filtered colors

Error Index

filterRGBPixels

```
public void filterRGBPixels(int x,  
    int y,  
    int w,  
    int h,  
    int pixels[],  
    int off,  
    int scansize)
```

Filters a buffer of pixels in the default RGB ColorModel by passing them one by one through the filterRGB method.

See Also:

[getRGBdefault](#), [filterRGB](#)

Error Index

setPixels

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

If the [ColorModel](#) object is the same one that has already been converted, then simply passes the pixels through with the converted [ColorModel](#). Otherwise converts the buffer of byte pixels to the default RGB [ColorModel](#) and passes the converted buffer to the [filterRGBPixels](#) method to be converted one by one.

Overrides:

[setPixels](#) in class [ImageFilter](#)

See Also:

[getRGBdefault](#), [filterRGBPixels](#)

Error Index

setPixels

```
public void setPixels(int x,  
    int y,
```

```
int w,  
int h,  
ColorModel model,  
int pixels[],  
int off,  
int scansize)
```

If the `ColorModel` object is the same one that has already been converted, then simply passes the pixels through with the converted `ColorModel`, otherwise converts the buffer of integer pixels to the default RGB `ColorModel` and passes the converted buffer to the `filterRGBPixels` method to be converted one by one. Converts a buffer of integer pixels to the default RGB `ColorModel` and passes the converted buffer to the `filterRGBPixels` method.

Overrides:

[setPixels](#) in class [ImageFilter](#)

See Also:

[getRGBdefault](#), [filterRGBPixels](#)

Error Index

filterRGB

```
public abstract int filterRGB(int x,  
int y,  
int rgb)
```

Subclasses must specify a method to convert a single input pixel in the default RGB `ColorModel` to a single output pixel.

See Also:

[getRGBdefault](#), [filterRGBPixels](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Insets

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Insets

[java.lang.Object](#)

|
+----java.awt.Insets

public class **Insets**
extends [Object](#)
implements [Cloneable](#)

The insets of a container. This class is used to layout containers.

See Also:

[LayoutManager](#), [Container](#)

Error Index

Error Index

bottom

The inset from the bottom.

Error Index

left

The inset from the left.

Error Index

right

The inset from the right.

Error Index

top

The inset from the top.

Error Index

Error Index

Insets(int, int, int, int)

Constructs and initializes a new Inset with the specified top, left, bottom, and right insets.

Error Index

Error Index

clone()

Creates a clone of the object.

Error Index

toString()

Returns a String object representing this Inset's values.

Error Index

Error Index

top

public int top

The inset from the top.

Error Index

left

public int left

The inset from the left.

Error Index

bottom

public int bottom

The inset from the bottom.

Error Index

right

public int right

The inset from the right.

Error Index

Error Index Insets

```
public Insets(int top,  
             int left,  
             int bottom,  
             int right)
```

Constructs and initializes a new Inset with the specified top, left, bottom, and right insets.

Parameters:

- top - the inset from the top
- left - the inset from the left
- bottom - the inset from the bottom
- right - the inset from the right

Error Index

Error Index toString

```
public String toString()
```

Returns a String object representing this Inset's values.

Overrides:

toString in class Object

Error Index clone

```
public Object clone()
```

Creates a clone of the object.

Overrides:

clone in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Label

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Label

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Label
```

```
public class Label
    extends Component
```

A component that displays a single line of read-only text.

Error Index

Error Index

The center alignment.

CENTER

Error Index

The left alignment.

LEFT

Error Index

The right alignment.

RIGHT

Error Index

Error Index

Constructs an empty label.

Label()

Error Index

Constructs a new label with the specified String of text.

Label(String)

Error Index

Label(String, int)

Constructs a new label with the specified String of text and the specified alignment.

Error Index

Error Index

addNotify()

Creates the peer for this label.

Error Index

getAlignment()

Gets the current alignment of this label.

Error Index

getText()

Gets the text of this label.

Error Index

paramString()

Returns the parameter String of this label.

Error Index

setAlignment(int)

Sets the alignment for this label to the specified alignment.

Error Index

setText(String)

Sets the text for this label to the specified text.

Error Index

Error Index

LEFT

public final static int LEFT

The left alignment.

Error Index

CENTER

public final static int CENTER

The center alignment.

Error Index RIGHT

```
public final static int RIGHT
```

The right alignment.

Error Index

Error Index Label

```
public Label()
```

Constructs an empty label.

Error Index Label

```
public Label(String label)
```

Constructs a new label with the specified String of text.

Parameters:

label - the text that makes up the label

Error Index Label

```
public Label(String label,  
            int alignment)
```

Constructs a new label with the specified String of text and the specified alignment.

Parameters:

label - the String that makes up the label

alignment - the alignment value

Error Index

Error Index `addNotify`

`public synchronized void addNotify()`

Creates the peer for this label. The peer allows us to modify the appearance of the label without changing its functionality.

Overrides:

`addNotify` in class `Component`

Error Index `getAlignment`

`public int getAlignment()`

Gets the current alignment of this label.

See Also:

`setAlignment`

Error Index `setAlignment`

`public void setAlignment(int alignment)`

Sets the alignment for this label to the specified alignment.

Parameters:

alignment - the alignment value

Throws: `IllegalArgumentException`

If an improper alignment was given.

See Also:

`getAlignment`

Error Index `getText`

`public String getText()`

Gets the text of this label.

See Also:

`setText`

Error Index `setText`

`public void setText(String label)`

Sets the text for this label to the specified text.

Parameters:

label - the text that makes up the label

See Also:

getText

Error Index `paramString`

`protected String paramString()`

Returns the parameter String of this label.

Overrides:

paramString in class Component

Interface java.awt.LayoutManager

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.LayoutManager

public interface **LayoutManager**
extends [Object](#)

Defines the interface for classes that know how to layout Containers.

See Also:

[Container](#)

Error Index

Error Index

[addLayoutComponent](#)(String, Component)

Adds the specified component with the specified name to the layout.

Error Index

[layoutContainer](#)(Container)

Lays out the container in the specified panel.

Error Index

[minimumLayoutSize](#)(Container)

Calculates the minimum size dimensions for the specified panel given the components in the specified parent container.

Error Index

[preferredLayoutSize](#)(Container)

Calculates the preferred size dimensions for the specified panel given the components in the specified parent container.

Error Index

[removeLayoutComponent](#)(Component)

Removes the specified component from the layout.

Error Index

Error Index

addLayoutComponent

public abstract void addLayoutComponent([String](#) name,
[Component](#) comp)

Adds the specified component with the specified name to the layout.

Parameters:

name - the component name

comp - the component to be added

Error Index removeLayoutComponent

public abstract void removeLayoutComponent(Component comp)

Removes the specified component from the layout.

Parameters:

comp - the component to be removed

Error Index preferredLayoutSize

public abstract Dimension preferredLayoutSize(Container parent)

Calculates the preferred size dimensions for the specified panel given the components in the specified parent container.

Parameters:

parent - the component to be laid out

See Also:

minimumLayoutSize

Error Index minimumLayoutSize

public abstract Dimension minimumLayoutSize(Container parent)

Calculates the minimum size dimensions for the specified panel given the components in the specified parent container.

Parameters:

parent - the component to be laid out

See Also:

preferredLayoutSize

Error Index layoutContainer

public abstract void layoutContainer([Container](#) parent)

Lays out the container in the specified panel.

Parameters:

parent - the component which needs to be laid out

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.List

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.List

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.List
```

```
public class List
extends Component
```

A scrolling list of text items.

Error Index

Error Index

List()

Creates a new scrolling list initialized with no visible Lines or multiple selections.

Error Index

List(int, boolean)

Creates a new scrolling list initialized with the specified number of visible lines and a boolean stating whether multiple selections are allowed or not.

Error Index

Error Index

addItem(String)

Adds the specified item to the end of scrolling list.

Error Index

addItem(String, int)

Adds the specified item to the end of scrolling list.

Error Index

addNotify()

Creates the peer for the list.

Error Index

allowsMultipleSelections()

Returns true if this list allows multiple selections.

Error Index

clear()

Clears the list.

Error Index

countItems()

Returns the number of items in the list.

Error Index

dellItem(int)

Delete an item from the list.

Error Index

dellItems(int, int)

Delete multiple items from the list.

Error Index

deselect(int)

Deselects the item at the specified index.

Error Index

getItem(int)

Gets the item associated with the specified index.

Error Index

getRows()

Returns the number of visible lines in this list.

Error Index

getSelectedIndex()

Get the selected item on the list or -1 if no item is selected.

Error Index

getSelectedIndexes()

Returns the selected indexes on the list.

Error Index

getSelectedItem()

Returns the selected item on the list or null if no item is selected.

Error Index

getSelectedItems()

Returns the selected items on the list.

Error Index

getVisibleIndex()

Gets the index of the item that was last made visible by the method makeVisible.

Error Index

isSelected(int)

Returns true if the item at the specified index has been selected; false otherwise.

Error Index

makeVisible(int)

Forces the item at the specified index to be visible.

Error Index

minimumSize(int)

Returns the minimum dimensions needed for the amount of rows in the list.

Error Index

minimumSize()

Returns the minimum dimensions needed for the list.

Error Index

paramString()

Returns the parameter String of this list.

Error Index

preferredSize(int)

Returns the preferred dimensions needed for the list with the specified amount of rows.

Error Index

preferredSize()

Returns the preferred dimensions needed for the list.

Error Index

removeNotify()

Removes the peer for this list.

Error Index

replaceItem(String, int)

Replaces the item at the given index.

Error Index

select(int)

Selects the item at the specified index.

Error Index

setMultipleSelections(boolean)

Sets whether this list should allow multiple selections or not.

Error Index

Error Index

List

public List()

Creates a new scrolling list initialized with no visible Lines or multiple selections.

Error Index

List

public List(int rows,
boolean multipleSelections)

Creates a new scrolling list initialized with the specified number of visible lines and a boolean stating

whether multiple selections are allowed or not.

Parameters:

rows - the number of items to show.

multipleSelections - if true then multiple selections are allowed.

Error Index

Error Index

addNotify

```
public synchronized void addNotify()
```

Creates the peer for the list. The peer allows us to modify the list's appearance without changing its functionality.

Overrides:

addNotify in class Component

Error Index

removeNotify

```
public synchronized void removeNotify()
```

Removes the peer for this list. The peer allows us to modify the list's appearance without changing its functionality.

Overrides:

removeNotify in class Component

Error Index

countItems

```
public int countItems()
```

Returns the number of items in the list.

See Also:

getItem

Error Index

getItem

```
public String getItem(int index)
```

Gets the item associated with the specified index.

Parameters:

index - the position of the item

See Also:

countItems

Error Index addItem

```
public synchronized void addItem(String item)
```

Adds the specified item to the end of scrolling list.

Parameters:

item - the item to be added

Error Index addItem

```
public synchronized void addItem(String item,  
int index)
```

Adds the specified item to the end of scrolling list.

Parameters:

item - the item to be added

index - the position at which to put in the item. The index is zero-based. If index is -1 then the item is added to the end. If index is greater than the number of items in the list, the item gets added at the end.

Error Index replaceItem

```
public synchronized void replaceItem(String newValue,  
int index)
```

Replaces the item at the given index.

Parameters:

newValue - the new value to replace the existing item

index - the position of the item to replace

Error Index clear

public synchronized void clear()

Clears the list.

See Also:

delItem, delItems

Error Index delItem

public synchronized void delItem(int position)

Delete an item from the list.

Error Index delItems

public synchronized void delItems(int start,
int end)

Delete multiple items from the list.

Error Index getSelectedIndex

public synchronized int getSelectedIndex()

Get the selected item on the list or -1 if no item is selected.

See Also:

select, deselect, isSelected

Error Index getSelectedIndexes

public synchronized int[] getSelectedIndexes()

Returns the selected indexes on the list.

See Also:

select, deselect, isSelected

Error Index **getSelectedItem**

public synchronized String getItem()

Returns the selected item on the list or null if no item is selected.

See Also:

select, deselect, isSelected

Error Index **getSelectedItems**

public synchronized String[] getItem()

Returns the selected items on the list.

See Also:

select, deselect, isSelected

Error Index **select**

public synchronized void select(int index)

Selects the item at the specified index.

Parameters:

index - the position of the item to select

See Also:

getItem, deselect, isSelected

Error Index **deselect**

public synchronized void deselect(int index)

Deselects the item at the specified index.

Parameters:

index - the position of the item to deselect

See Also:

select, getItem, isSelected

Error Index **isSelected**

```
public synchronized boolean isSelected(int index)
```

Returns true if the item at the specified index has been selected; false otherwise.

Parameters:

index - the item to be checked

See Also:

[select](#), [deselect](#), [isSelected](#)

Error Index **getRows**

```
public int getRows()
```

Returns the number of visible lines in this list.

Error Index **allowsMultipleSelections**

```
public boolean allowsMultipleSelections()
```

Returns true if this list allows multiple selections.

See Also:

[setMultipleSelections](#)

Error Index **setMultipleSelections**

```
public void setMultipleSelections(boolean v)
```

Sets whether this list should allow multiple selections or not.

Parameters:

v - the boolean to allow multiple selections

See Also:

[allowsMultipleSelections](#)

Error Index **getVisibleIndex**

```
public int getVisibleIndex()
```

Gets the index of the item that was last made visible by the method `makeVisible`.

Error Index `makeVisible`

```
public void makeVisible(int index)
```

Forces the item at the specified index to be visible.

Parameters:

index - the position of the item

See Also:

[getVisibleIndex](#)

Error Index `preferredSize`

```
public Dimension preferredSize(int rows)
```

Returns the preferred dimensions needed for the list with the specified amount of rows.

Parameters:

rows - amount of rows in list.

Error Index `preferredSize`

```
public Dimension preferredSize()
```

Returns the preferred dimensions needed for the list.

Returns:

the preferred size with the specified number of rows if the row size is greater than 0.

Overrides:

[preferredSize](#) in class [Component](#)

Error Index `minimumSize`

```
public Dimension minimumSize(int rows)
```

Returns the minimum dimensions needed for the amount of rows in the list.

Parameters:

rows - minimum amount of rows in the list

Error Index `minimumSize`

public Dimension minimumSize()

Returns the minimum dimensions needed for the list.

Returns:

the preferred size with the specified number of rows if the row size is greater than zero.

Overrides:

minimumSize in class Component

Error Index `paramString`

protected String paramString()

Returns the parameter String of this list.

Overrides:

paramString in class Component

Class java.awt.MediaTracker

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.MediaTracker

[java.lang.Object](#)

|
+----java.awt.MediaTracker

```
public class MediaTracker  
extends Object
```

A utility class to track the status of a number of media objects. Media objects could include images as well as audio clips, though currently only images are supported. To use it, simply create an instance and then call `addImage()` for each image to be tracked. Each image can be assigned a unique ID for identification purposes. The IDs control the priority order in which the images are fetched as well as identifying unique subsets of the images that can be waited on independently. Here is an example:

```
import java.applet.Applet;  
import java.awt.Color;  
import java.awt.Image;  
import java.awt.Graphics;  
import java.awt.MediaTracker;  
public class ImageBlaster extends Applet implements Runnable {  
    MediaTracker tracker;  
    Image bg;  
    Image anim[] = new Image[5];  
    int index;  
    Thread animator;  
    // Get the images for the background (id == 0) and the animation  
    // frames (id == 1) and add them to the MediaTracker  
    public void init() {  
        tracker = new MediaTracker(this);  
        bg = getImage(getDocumentBase(), "images/background.gif");  
        tracker.addImage(bg, 0);  
        for (int i = 0; i = anim.length) {  
            index = 0;  
        }  
    }  
    repaint();  
}  
// The background image fills our frame so we don't need to clear  
// the applet on repaints, just call the paint method.  
public void update(Graphics g) {  
    paint(g);  
}
```

```

    }
    // Paint a large red rectangle if there are any errors loading the
    // images. Otherwise always paint the background so that it appears
    // incrementally as it is loading. Finally, only paint the current
    // animation frame if all of the frames (id == 1) are done loading
    // so that we don't get partial animations.
    public void paint(Graphics g) {
        if ((tracker.statusAll() & MediaTracker.ERROR) != 0) {
            g.setColor(Color.red);
            g.fillRect(0, 0, size().width, size().height);
            return;
        }
        g.drawImage(bg, 0, 0, this);
        if ((tracker.statusID(1) & MediaTracker.COMPLETE) != 0) {
            g.drawImage(anim[index], 10, 10, this);
        }
    }
}

```

Error Index

Error Index

ABORTED

Flag indicating the download of some media was aborted.

Error Index

COMPLETE

Flag indicating the download of media completed successfully.

Error Index

ERROR

Flag indicating the download of some media encountered an error.

Error Index

LOADING

Flag indicating some media is currently being loaded.

Error Index

Error Index

MediaTracker(Component)

Creates a Media tracker to track images for a given Component.

Error Index

Error Index

addImage(Image, int)

Adds an image to the list of images being tracked.

Error Index

addImage(Image, int, int, int)

Adds a scaled image to the list of images being tracked.

Error Index

checkAll()

Checks to see if all images have finished loading but does not start loading the images if they are not already loading.

Error Index

checkAll(boolean)

Checks to see if all images have finished loading.

Error Index

checkID(int)

Checks to see if all images tagged with the indicated ID have finished loading, but does not start loading the images if they are not already loading.

Error Index

checkID(int, boolean)

Checks to see if all images tagged with the indicated ID have finished loading.

Error Index

getErrorsAny()

Returns a list of all media that have encountered an error.

Error Index

getErrorsID(int)

Returns a list of media with the specified ID that have encountered an error.

Error Index

isErrorAny()

Checks the error status of all of the images.

Error Index

isErrorID(int)

Checks the error status of all of the images with the specified ID.

Error Index

statusAll(boolean)

Returns the boolean OR of the status of all of the media being tracked.

Error Index

statusID(int, boolean)

Returns the boolean OR of the status of all of the media with a given ID.

Error Index

waitForAll()

Starts loading all images.

Error Index

waitForAll(long)

Starts loading all images.

Error Index

waitForID(int)

Starts loading all images with the specified ID and waits until they have finished loading or receive an error.

Error Index

waitForID(int, long)

Starts loading all images with the specified ID.

Error Index

Error Index

LOADING

public final static int LOADING

Flag indicating some media is currently being loaded.

See Also:

[statusAll](#), [statusID](#)

Error Index

ABORTED

public final static int ABORTED

Flag indicating the download of some media was aborted.

See Also:

[statusAll](#), [statusID](#)

Error Index

ERRORED

public final static int ERRORED

Flag indicating the download of some media encountered an error.

See Also:

[statusAll](#), [statusID](#)

Error Index

COMPLETE

public final static int COMPLETE

Flag indicating the download of media completed successfully.

See Also:

statusAll, statusID

Error Index

Error Index **MediaTracker**

```
public MediaTracker(Component comp)
```

Creates a Media tracker to track images for a given Component.

Parameters:

comp - the component on which the images will eventually be drawn

Error Index

Error Index **addImage**

```
public void addImage(Image image,  
                    int id)
```

Adds an image to the list of images being tracked. The image will eventually be rendered at its default (unscaled) size.

Parameters:

image - the image to be tracked

id - the identifier used to later track this image

Error Index **addImage**

```
public synchronized void addImage(Image image,  
                                  int id,  
                                  int w,  
                                  int h)
```

Adds a scaled image to the list of images being tracked. The image will eventually be rendered at the indicated size.

Parameters:

image - the image to be tracked
id - the identifier used to later track this image
w - the width that the image will be rendered at
h - the height that the image will be rendered at

Error Index **checkAll**

public boolean checkAll()

Checks to see if all images have finished loading but does not start loading the images if they are not already loading. If there is an error while loading or scaling an image then that image is considered "complete." Use `isErrorAny()` or `isErrorID()` to check for errors.

Returns:

true if all images have finished loading, were aborted or encountered an error

See Also:

[checkAll](#), [checkID](#), [isErrorAny](#), [isErrorID](#)

Error Index **checkAll**

public synchronized boolean checkAll(boolean load)

Checks to see if all images have finished loading. If load is true, starts loading any images that are not yet being loaded. If there is an error while loading or scaling an image then that image is considered "complete." Use `isErrorAny()` or `isErrorID()` to check for errors.

Parameters:

load - start loading the images if this parameter is true

Returns:

true if all images have finished loading, were aborted or encountered an error

See Also:

[isErrorAny](#), [isErrorID](#), [checkID](#), [checkAll](#)

Error Index **isErrorAny**

public synchronized boolean isErrorAny()

Checks the error status of all of the images.

Returns:

true if any of the images had an error during loading

See Also:

[isErrorID](#), [getErrorsAny](#)

Error Index **getErrorsAny**

public synchronized Object[] getErrorsAny()

Returns a list of all media that have encountered an error.

Returns:

an array of media objects or null if there are no errors

See Also:

isErrorAny, getErrorsID

Error Index **waitForAll**

public void waitForAll() throws InterruptedException

Starts loading all images. Waits until they have finished loading, are aborted, or it receives an error. If there is an error while loading or scaling an image then that image is considered "complete." Use isErrorAny() or statusAll() to check for errors.

Throws: InterruptedException

Another thread has interrupted this thread.

See Also:

waitForID, waitForAll, isErrorAny, isErrorID

Error Index **waitForAll**

public synchronized boolean waitForAll(long ms) throws InterruptedException

Starts loading all images. Waits until they have finished loading, are aborted, it receives an error, or until the specified timeout has elapsed. If there is an error while loading or scaling an image then that image is considered "complete." Use isErrorAny() or statusAll() to check for errors.

Parameters:

ms - the length of time to wait for the loading to complete

Returns:

true if all images were successfully loaded

Throws: InterruptedException

Another thread has interrupted this thread.

See Also:

waitForID, waitForAll, isErrorAny, isErrorID

Error Index **statusAll**

public int statusAll(boolean load)

Returns the boolean OR of the status of all of the media being tracked.

Parameters:

load - specifies whether to start the media loading

See Also:

[statusID](#), [LOADING](#), [ABORTED](#), [ERRORED](#), [COMPLETE](#)

Error Index [checkID](#)

public boolean checkID(int id)

Checks to see if all images tagged with the indicated ID have finished loading, but does not start loading the images if they are not already loading. If there is an error while loading or scaling an image then that image is considered "complete." Use [isErrorAny\(\)](#) or [isErrorID\(\)](#) to check for errors.

Parameters:

id - the identifier used to determine which images to check

Returns:

true if all tagged images have finished loading, were aborted, or an error occurred.

See Also:

[checkID](#), [checkAll](#), [isErrorAny](#), [isErrorID](#)

Error Index [checkID](#)

public synchronized boolean checkID(int id,
boolean load)

Checks to see if all images tagged with the indicated ID have finished loading. If load is true, starts loading any images with that ID that are not yet being loaded. If there is an error while loading or scaling an image then that image is considered "complete." Use [isErrorAny\(\)](#) or [isErrorID\(\)](#) to check for errors.

Parameters:

id - the identifier used to determine which images to check

load - start loading the images if this parameter is true

Returns:

true if all tagged images have finished loading, were aborted, or an error occurred

See Also:

[checkID](#), [checkAll](#), [isErrorAny](#), [isErrorID](#)

Error Index [isErrorID](#)

public synchronized boolean isErrorID(int id)

Checks the error status of all of the images with the specified ID.

Parameters:

id - the identifier used to determine which images to check

Returns:

true if any of the tagged images had an error during loading

See Also:

[isErrorAny](#), [getErrorsID](#)

Error Index

getErrorsID

public synchronized [Object\[\]](#) getErrorsID(int id)

Returns a list of media with the specified ID that have encountered an error.

Parameters:

id - the identifier used to determine which images to return

Returns:

an array of media objects or null if there are no errors

See Also:

[isErrorID](#), [getErrorsAny](#)

Error Index

waitForID

public void waitForID(int id) throws [InterruptedException](#)

Starts loading all images with the specified ID and waits until they have finished loading or receive an error. If there is an error while loading or scaling an image then that image is considered "complete." Use [statusID\(\)](#) or [isErrorID\(\)](#) to check for errors.

Parameters:

id - the identifier used to determine which images to wait for

Throws: [InterruptedException](#)

Another thread has interrupted this thread.

See Also:

[waitForAll](#), [waitForID](#), [isErrorAny](#), [isErrorID](#)

Error Index

waitForID

public synchronized boolean waitForID(int id,
long ms) throws [InterruptedException](#)

Starts loading all images with the specified ID. Waits until they have finished loading, an error occurs, or the specified timeout has elapsed. If there is an error while loading or scaling an image then that image is considered "complete." Use `statusID` or `isErrorID` to check for errors.

Parameters:

id - the identifier used to determine which images to wait for

ms - the length of time to wait for the loading to complete

Throws: [InterruptedException](#)

Another thread has interrupted this thread.

See Also:

[waitForAll](#), [waitForID](#), [isErrorAny](#), [isErrorID](#)

Error Index `statusID`

```
public int statusID(int id,  
                   boolean load)
```

Returns the boolean OR of the status of all of the media with a given ID.

Parameters:

id - the identifier used to determine which images to check

load - specifies whether to start the media loading

See Also:

[statusAll](#), [LOADING](#), [ABORTED](#), [ERRORED](#), [COMPLETE](#)

Class java.awt.Menu

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Menu

```
java.lang.Object
|
+----java.awt.MenuComponent
      |
      +----java.awt.MenuItem
            |
            +----java.awt.Menu
```

```
public class Menu
  extends MenuItem
  implements MenuContainer
```

A Menu that is a component of a menu bar.

Error Index

Error Index

Menu(String)

Constructs a new Menu with the specified label.

Error Index

Menu(String, boolean)

Constructs a new Menu with the specified label.

Error Index

Error Index

add(MenuItem)

Adds the specified item to this menu.

Error Index

add(String)

Adds an item with with the specified label to this menu.

Error Index

addNotify()

Creates the menu's peer.

Error Index

addSeparator()

Adds a separator line, or a hyphen, to the menu at the current position.

Error Index

countItems()

Returns the number of elements in this menu.

Error Index

getItem(int)

Returns the item located at the specified index of this menu.

Error Index

isTearOff()

Returns true if this is a tear-off menu.

Error Index

remove(int)

Deletes the item from this menu at the specified index.

Error Index

remove(MenuComponent)

Deletes the specified item from this menu.

Error Index

removeNotify()

Removes the menu's peer.

Error Index

Error Index

Menu

```
public Menu(String label)
```

Constructs a new Menu with the specified label. This menu can not be torn off - the menu will still appear on screen after the the mouse button has been released.

Parameters:

label - the label to be added to this menu

Error Index

Menu

```
public Menu(String label,  
            boolean tearOff)
```

Constructs a new Menu with the specified label. If tearOff is true, the menu can be torn off - the menu will still appear on screen after the the mouse button has been released.

Parameters:

label - the label to be added to this menu

tearOff - the boolean indicating whether or not the menu will be able to be torn off.

Error Index

Error Index **addNotify**

```
public synchronized void addNotify()
```

Creates the menu's peer. The peer allows us to modify the appearance of the menu without changing its functionality.

Overrides:

addNotify in class MenuItem

Error Index **removeNotify**

```
public synchronized void removeNotify()
```

Removes the menu's peer. The peer allows us to modify the appearance of the menu without changing its functionality.

Overrides:

removeNotify in class MenuComponent

Error Index **isTearOff**

```
public boolean isTearOff()
```

Returns true if this is a tear-off menu.

Error Index **countItems**

```
public int countItems()
```

Returns the number of elements in this menu.

Error Index **getItem**

```
public MenuItem getItem(int index)
```

Returns the item located at the specified index of this menu.

Parameters:

index - the position of the item to be returned

Error Index **add**

```
public synchronized MenuItem add(MenuItem mi)
```

Adds the specified item to this menu.

Parameters:

mi - the item to be added

Error Index **add**

```
public void add(String label)
```

Adds an item with with the specified label to this menu.

Parameters:

label - the text on the item

Error Index **addSeparator**

```
public void addSeparator()
```

Adds a separator line, or a hyphen, to the menu at the current position.

Error Index **remove**

```
public synchronized void remove(int index)
```

Deletes the item from this menu at the specified index.

Parameters:

index - the position of the item to be removed

Error Index remove

public synchronized void remove(MenuComponent item)

Deletes the specified item from this menu.

Parameters:

item - the item to be removed from the menu

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.MenuBar

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.MenuBar

[java.lang.Object](#)

```
|
+----java.awt.MenuComponent
      |
      +----java.awt.MenuBar
```

public class **MenuBar**
extends [MenuComponent](#)
implements [MenuContainer](#)

A class that encapsulates the platform's concept of a menu bar bound to a Frame. In order to associate the MenuBar with an actual Frame, the `Frame.setMenuBar()` method should be called.

See Also:

[setMenuBar](#)

Error Index

Error Index

[MenuBar\(\)](#)

Creates a new menu bar.

Error Index

Error Index

[add\(Menu\)](#)

Adds the specified menu to the menu bar.

Error Index

[addNotify\(\)](#)

Creates the menu bar's peer.

Error Index

[countMenus\(\)](#)

Counts the number of menus on the menu bar.

Error Index

getHelpMenu()

Gets the help menu on the menu bar.

Error Index

getMenu(int)

Gets the specified menu.

Error Index

remove(int)

Removes the menu located at the specified index from the menu bar.

Error Index

remove(MenuComponent)

Removes the specified menu from the menu bar.

Error Index

removeNotify()

Removes the menu bar's peer.

Error Index

setHelpMenu(Menu)

Sets the help menu to the specified menu on the menu bar.

Error Index

Error Index

MenuBar

public MenuBar()

Creates a new menu bar.

Error Index

Error Index

addNotify

public synchronized void addNotify()

Creates the menu bar's peer. The peer allows us to change the appearance of the menu bar without changing any of the menu bar's functionality.

Error Index

removeNotify

public void removeNotify()

Removes the menu bar's peer. The peer allows us to change the appearance of the menu bar without changing any of the menu bar's functionality.

Overrides:

removeNotify in class MenuComponent

Error Index **getHelpMenu**

```
public Menu getHelpMenu()
```

Gets the help menu on the menu bar.

Error Index **setHelpMenu**

```
public synchronized void setHelpMenu(Menu m)
```

Sets the help menu to the specified menu on the menu bar.

Parameters:

m - the menu to be set

Error Index **add**

```
public synchronized Menu add(Menu m)
```

Adds the specified menu to the menu bar.

Parameters:

m - the menu to be added to the menu bar

Error Index **remove**

```
public synchronized void remove(int index)
```

Removes the menu located at the specified index from the menu bar.

Parameters:

index - the position of the menu to be removed

Error Index remove

public synchronized void remove(MenuComponent m)

Removes the specified menu from the menu bar.

Parameters:

m - the menu to be removed

Error Index countMenus

public int countMenus()

Counts the number of menus on the menu bar.

Error Index getMenu

public Menu getMenu(int i)

Gets the specified menu.

Parameters:

i - the menu to be returned

Class java.awt.MenuComponent

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.MenuComponent

[java.lang.Object](#)

|

+----java.awt.MenuComponent

```
public class MenuComponent
    extends Object
```

The super class of all menu related components.

Error Index

Error Index

MenuComponent()

Error Index

Error Index

getFont()

Gets the font used for this MenuItem.

Error Index

getParent()

Returns the parent container.

Error Index

getPeer()

Gets the MenuComponent's peer.

Error Index

paramString()

Returns the String parameter of this MenuComponent.

Error Index

postEvent(Event)

Posts the specified event to the menu.

Error Index

removeNotify()

Removes the menu component's peer.

Error Index

setFont(Font)

Sets the font to be used for this MenuItem to the specified font.

Error Index

toString()

Returns the String representation of this MenuComponent's values.

Error Index

Error Index

MenuComponent

```
public MenuComponent()
```

Error Index

Error Index

getParent

```
public MenuContainer getParent()
```

Returns the parent container.

Error Index

getPeer

```
public MenuComponentPeer getPeer()
```

Gets the MenuComponent's peer. The peer allows us to modify the appearance of the menu component without changing the functionality of the menu component.

Error Index

getFont

```
public Font getFont()
```

Gets the font used for this MenuItem.

Returns:

the font if one is used; null otherwise.

Error Index setFont

```
public void setFont(Font f)
```

Sets the font to be used for this MenuItem to the specified font.

Parameters:

f - the font to be set

Error Index removeNotify

```
public void removeNotify()
```

Removes the menu component's peer. The peer allows us to modify the appearance of the menu component without changing the functionality of the menu component.

Error Index postEvent

```
public boolean postEvent(Event evt)
```

Posts the specified event to the menu.

Parameters:

evt - the event which is to take place

Error Index paramString

```
protected String paramString()
```

Returns the String parameter of this MenuComponent.

Error Index toString

```
public String toString()
```

Returns the String representation of this MenuComponent's values.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.MenuContainer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.MenuContainer

public interface **MenuContainer**
extends [Object](#)

The super class of all menu related containers.

Error Index

Error Index

[getFont\(\)](#)

Error Index

[postEvent](#)([Event](#))

Error Index

[remove](#)([MenuComponent](#))

Error Index

Error Index

[getFont](#)

public abstract [Font](#) [getFont](#)()

Error Index

[postEvent](#)

public abstract boolean [postEvent](#)([Event](#) evt)

Error Index

[remove](#)

public abstract void [remove](#)([MenuComponent](#) comp)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.MenuItem

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.MenuItem

```
java.lang.Object
|
+----java.awt.MenuComponent
      |
      +----java.awt.MenuItem
```

```
public class MenuItem
    extends MenuComponent
```

A String item that represents a choice in a menu.

Error Index

Error Index

[**MenuItem**](#)(String)

Constructs a new MenuItem with the specified label.

Error Index

Error Index

[**addNotify**](#)()

Creates the menu item's peer.

Error Index

[**disable**](#)()

Makes this menu item unselectable by the user.

Error Index

[**enable**](#)()

Makes this menu item selectable by the user.

Error Index

[**enable**](#)(boolean)

Conditionally enables a component.

Error Index

getLabel()

Gets the label for this menu item.

Error Index

isEnabled()

Checks whether the menu item is enabled.

Error Index

paramString()

Returns the String parameter of the menu item.

Error Index

setLabel(String)

Sets the label to be the specified label.

Error Index

Error Index

MenuItem

public MenuItem(String label)

Constructs a new MenuItem with the specified label.

Parameters:

label - the label for this menu item. Note that "-" is reserved to mean a separator between menu items.

Error Index

Error Index

addNotify

public synchronized void addNotify()

Creates the menu item's peer. The peer allows us to modify the appearance of the menu item without changing its functionality.

Error Index

getLabel

public String getLabel()

Gets the label for this menu item.

Error Index

setLabel

```
public void setLabel(String label)
```

Sets the label to be the specified label.

Parameters:

label - the label for this menu item

Error Index

isEnabled

```
public boolean isEnabled()
```

Checks whether the menu item is enabled.

Error Index

enable

```
public void enable()
```

Makes this menu item selectable by the user.

Error Index

enable

```
public void enable(boolean cond)
```

Conditionally enables a component.

Parameters:

cond - enabled if true; disabled otherwise.

See Also:

enable, disable

Error Index

disable

```
public void disable()
```

Makes this menu item unselectable by the user.

Error Index paramString

public String paramString()

Returns the String parameter of the menu item.

Overrides:

paramString in class MenuComponent

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Panel

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Panel

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Container
            |
            +----java.awt.Panel
```

```
public class Panel
  extends Container
```

A Panel Container class. This produces a generic container.

Error Index

Error Index

Panel()

Creates a new panel.

Error Index

Error Index

addNotify()

Creates the Panel's peer.

Error Index

Error Index

Panel

```
public Panel()
```

Creates a new panel. The default layout for all panels is FlowLayout.

Error Index

Error Index **addNotify**

public synchronized void addNotify()

Creates the Panel's peer. The peer allows you to modify the appearance of the panel without changing its functionality.

Overrides:

addNotify in class Container

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ButtonPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ButtonPeer

public interface **ButtonPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

setLabel(String)

Error Index

Error Index

setLabel

public abstract void setLabel([String](#) label)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.CanvasPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.CanvasPeer

public interface **CanvasPeer**

extends [Object](#)

extends [ComponentPeer](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface

java.awt.peer.CheckboxMenuItemPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.CheckboxMenuItemPeer

public interface **CheckboxMenuItemPeer**
extends [Object](#)
extends [MenuItemPeer](#)

Error Index

Error Index

setState(boolean)

Error Index

Error Index

setState

public abstract void setState(boolean t)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.CheckboxPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.CheckboxPeer

public interface **CheckboxPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

setCheckboxGroup([CheckboxGroup](#))

Error Index

setLabel([String](#))

Error Index

setState(boolean)

Error Index

Error Index

setState

public abstract void setState(boolean state)

Error Index

setCheckboxGroup

public abstract void setCheckboxGroup([CheckboxGroup](#) g)

Error Index

setLabel

public abstract void setLabel([String](#) label)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ChoicePeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ChoicePeer

public interface **ChoicePeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

addItem(String, int)

Error Index

select(int)

Error Index

Error Index

addItem

public abstract void addItem([String](#) item,
int index)

Error Index

select

public abstract void select(int index)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ComponentPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ComponentPeer

public interface **ComponentPeer**
extends [Object](#)

Error Index

Error Index

[checkImage](#)(Image, int, int, ImageObserver)

Error Index

[createImage](#)(ImageProducer)

Error Index

[createImage](#)(int, int)

Error Index

[disable](#)()

Error Index

[dispose](#)()

Error Index

[enable](#)()

Error Index

[getColorModel](#)()

Error Index

[getFontMetrics](#)(Font)

Error Index

[getGraphics](#)()

Error Index

[getToolkit](#)()

Error Index

[handleEvent](#)(Event)

Error Index

[hide](#)()

Error Index

[minimumSize](#)()

Error Index

nextFocus()

Error Index

paint(Graphics)

Error Index

preferredSize()

Error Index

prepareImage (Image, int, int, ImageObserver)

Error Index

print(Graphics)

Error Index

repaint (long, int, int, int, int)

Error Index

requestFocus()

Error Index

reshape(int, int, int, int)

Error Index

setBackground(Color)

Error Index

setFont(Font)

Error Index

setForeground(Color)

Error Index

show()

Error Index

Error Index

show

public abstract void show()

Error Index

hide

public abstract void hide()

Error Index

enable

public abstract void enable()

Error Index **disable**

public abstract void disable()

Error Index **paint**

public abstract void paint(Graphics g)

Error Index **repaint**

public abstract void repaint(long tm,
int x,
int y,
int width,
int height)

Error Index **print**

public abstract void print(Graphics g)

Error Index **reshape**

public abstract void reshape(int x,
int y,
int width,
int height)

Error Index **handleEvent**

public abstract boolean handleEvent(Event e)

Error Index **minimumSize**

public abstract Dimension minimumSize()

Error Index preferredSize

public abstract Dimension preferredSize()

Error Index getColorModel

public abstract ColorModel getColorModel()

Error Index getToolkit

public abstract Toolkit getToolkit()

Error Index getGraphics

public abstract Graphics getGraphics()

Error Index getFontMetrics

public abstract FontMetrics getFontMetrics(Font font)

Error Index dispose

public abstract void dispose()

Error Index setForeground

public abstract void setForeground(Color c)

Error Index setBackground

public abstract void setBackground(Color c)

setFont

ont f)

requestFocus

cus()

nextFocus

O

createImage

age(ImageProducer producer)

createImage

```
age(int width,
```

prepareImage

```
Image(Image img,
Observer o)
```

checkImage

(Image img,
over o)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ContainerPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ContainerPeer

public interface **ContainerPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

[insets\(\)](#)

Error Index

Error Index

[insets](#)

public abstract [Insets](#) insets()

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.DialogPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.DialogPeer

public interface **DialogPeer**

extends [Object](#)

extends [WindowPeer](#)

Error Index

Error Index

[setResizable](#)(boolean)

Error Index

[setTitle](#)(String)

Error Index

Error Index

setTitle

public abstract void setTitle([String](#) title)

Error Index

setResizable

public abstract void setResizable(boolean resizable)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.FileDialogPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.FileDialogPeer

public interface **FileDialogPeer**
extends [Object](#)
extends [DialogPeer](#)

Error Index

Error Index

setDirectory(String)

Error Index

setFile(String)

Error Index

setFilenameFilter (FilenameFilter)

Error Index

Error Index

setFile

public abstract void setFile([String](#) file)

Error Index

setDirectory

public abstract void setDirectory([String](#) dir)

Error Index

setFilenameFilter

public abstract void setFilenameFilter([FilenameFilter](#) filter)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.FramePeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.FramePeer

public interface **FramePeer**
extends [Object](#)
extends [WindowPeer](#)

Error Index

Error Index

setCursor(int)

Error Index

setIconImage(Image)

Error Index

setMenuBar(MenuBar)

Error Index

setResizable(boolean)

Error Index

setTitle(String)

Error Index

Error Index

setTitle

public abstract void setTitle([String](#) title)

Error Index

setIconImage

public abstract void setIconImage([Image](#) im)

Error Index

setMenuBar

public abstract void setMenuBar([MenuBar](#) mb)

Error Index **setResizable**

public abstract void setResizable(boolean resizeable)

Error Index **setCursor**

public abstract void setCursor(int cursorType)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.LabelPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.LabelPeer

public interface **LabelPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

[setAlignment](#)(int)

Error Index

[setText](#)(String)

Error Index

Error Index

setText

public abstract void setText([String](#) label)

Error Index

setAlignment

public abstract void setAlignment(int alignment)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ListPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ListPeer

public interface **ListPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

[addItem](#)(String, int)

Error Index

[clear](#)()

Error Index

[delItems](#)(int, int)

Error Index

[deselect](#)(int)

Error Index

[getSelectedIndexes](#)()

Error Index

[makeVisible](#)(int)

Error Index

[minimumSize](#)(int)

Error Index

[preferredSize](#)(int)

Error Index

[select](#)(int)

Error Index

[setMultipleSelections](#)(boolean)

Error Index

Error Index

[getSelectedIndexes](#)

```
public abstract int[] getSelectedIndexes()
```

Error Index addItem

```
public abstract void addItem(String item,  
                             int index)
```

Error Index delItems

```
public abstract void delItems(int start,  
                              int end)
```

Error Index clear

```
public abstract void clear()
```

Error Index select

```
public abstract void select(int index)
```

Error Index deselect

```
public abstract void deselect(int index)
```

Error Index makeVisible

```
public abstract void makeVisible(int index)
```

Error Index setMultipleSelections

```
public abstract void setMultipleSelections(boolean v)
```

Error Index preferredSize

public abstract [Dimension](#) preferredSize(int v)

Error Index **minimumSize**

public abstract [Dimension](#) minimumSize(int v)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuBarPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuBarPeer

public interface **MenuBarPeer**
extends [Object](#)
extends [MenuComponentPeer](#)

Error Index

Error Index

[addHelpMenu](#)(Menu)

Error Index

[addMenu](#)(Menu)

Error Index

[delMenu](#)(int)

Error Index

Error Index

addMenu

public abstract void addMenu([Menu](#) m)

Error Index

delMenu

public abstract void delMenu(int index)

Error Index

addHelpMenu

public abstract void addHelpMenu([Menu](#) m)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface

java.awt.peer.MenuComponentPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuComponentPeer

public interface **MenuComponentPeer**
extends [Object](#)

Error Index

Error Index

[dispose\(\)](#)

Error Index

Error Index

dispose

public abstract void dispose()

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuItemPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuitemPeer

public interface **MenuitemPeer**
extends [Object](#)
extends [MenuComponentPeer](#)

Error Index

Error Index

[disable\(\)](#)

Error Index

[enable\(\)](#)

Error Index

[setLabel\(String\)](#)

Error Index

Error Index

[setLabel](#)

public abstract void setLabel([String](#) label)

Error Index

[enable](#)

public abstract void enable()

Error Index

[disable](#)

public abstract void disable()

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.MenuPeer

public interface **MenuPeer**
extends [Object](#)
extends [MenuItemPeer](#)

Error Index

Error Index

[addItem](#)(MenuItem)

Error Index

[addSeparator](#)()

Error Index

[delItem](#)(int)

Error Index

Error Index

addSeparator

public abstract void addSeparator()

Error Index

addItem

public abstract void addItem([MenuItem](#) item)

Error Index

delItem

public abstract void delItem(int index)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface `java.awt.peer.PanelPeer`

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface `java.awt.peer.PanelPeer`

public interface **PanelPeer**

extends [Object](#)

extends [ContainerPeer](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ScrollbarPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.ScrollbarPeer

public interface **ScrollbarPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

[setLineIncrement](#)(int)

Error Index

[setPageIncrement](#)(int)

Error Index

[setValue](#)(int)

Error Index

[setValues](#)(int, int, int, int)

Error Index

Error Index

setValue

public abstract void setValue(int value)

Error Index

setValues

public abstract void setValues(int value,
int visible,
int minimum,
int maximum)

Error Index **setLineIncrement**

public abstract void setLineIncrement(int l)

Error Index **setPageIncrement**

public abstract void setPageIncrement(int l)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.TextAreaPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.TextAreaPeer

public interface **TextAreaPeer**
extends [Object](#)
extends [TextComponentPeer](#)

Error Index

Error Index

[insertText](#)(String, int)

Error Index

[minimumSize](#)(int, int)

Error Index

[preferredSize](#)(int, int)

Error Index

[replaceText](#)(String, int, int)

Error Index

Error Index

insertText

public abstract void insertText([String](#) txt,
int pos)

Error Index

replaceText

public abstract void replaceText([String](#) txt,
int start,
int end)

Error Index preferredSize

```
public abstract Dimension preferredSize(int rows,  
int cols)
```

Error Index minimumSize

```
public abstract Dimension minimumSize(int rows,  
int cols)
```

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface

java.awt.peer.TextComponentPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.TextComponentPeer

public interface **TextComponentPeer**
extends [Object](#)
extends [ComponentPeer](#)

Error Index

Error Index

[getSelectionEnd\(\)](#)

Error Index

[getSelectionStart\(\)](#)

Error Index

[getText\(\)](#)

Error Index

[select](#)(int, int)

Error Index

[setEditable](#)(boolean)

Error Index

[setText](#)(String)

Error Index

Error Index

setEditable

public abstract void setEditable(boolean editable)

Error Index

getText

public abstract String getText()

Error Index **setText**

public abstract void setText(String l)

Error Index **getSelectionStart**

public abstract int getSelectionStart()

Error Index **getSelectionEnd**

public abstract int getSelectionEnd()

Error Index **select**

public abstract void select(int selStart,
int selEnd)

Interface java.awt.peer.TextFieldPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.TextFieldPeer

public interface **TextFieldPeer**
extends [Object](#)
extends [TextComponentPeer](#)

Error Index

Error Index

[minimumSize](#)(int)

Error Index

[preferredSize](#)(int)

Error Index

[setEchoCharacter](#)(char)

Error Index

Error Index

setEchoCharacter

public abstract void setEchoCharacter(char c)

Error Index

preferredSize

public abstract [Dimension](#) preferredSize(int cols)

Error Index

minimumSize

public abstract [Dimension](#) minimumSize(int cols)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.WindowPeer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.awt.peer.WindowPeer

public interface **WindowPeer**
extends [Object](#)
extends [ContainerPeer](#)

Error Index

Error Index

[toBack\(\)](#)

Error Index

[toFront\(\)](#)

Error Index

Error Index

toFront

public abstract void toFront()

Error Index

toBack

public abstract void toBack()

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Point

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Point

[java.lang.Object](#)
|
+----java.awt.Point

public class **Point**
extends [Object](#)

An x,y coordinate.

Error Index

Error Index x
The x coordinate.

Error Index y
The y coordinate.

Error Index

Error Index **Point**(int, int)
Constructs and initializes a Point from the specified x and y coordinates.

Error Index

Error Index **equals**(Object)
Checks whether two pointers are equal.

Error Index

hashCode()

Returns the hashCode for this Point.

Error Index

move(int, int)

Moves the point.

Error Index

toString()

Returns the String representation of this Point's coordinates.

Error Index

translate(int, int)

Translates the point.

Error Index

Error Index_x

public int x

The x coordinate.

Error Index_y

public int y

The y coordinate.

Error Index

Error Index_{Point}

public Point(int x,
int y)

Constructs and initializes a Point from the specified x and y coordinates.

Parameters:

- x - the x coordinate
- y - the y coordinate

Error Index

Error Index `move`

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

Moves the point.

Error Index `translate`

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

Translates the point.

Error Index `hashCode`

```
public int hashCode()
```

Returns the hashcode for this Point.

Overrides:

hashCode in class Object

Error Index `equals`

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

Checks whether two pointers are equal.

Overrides:

equals in class Object

Error Index `toString`

```
public String toString()
```

Returns the String representation of this Point's coordinates.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Polygon

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Polygon

[java.lang.Object](#)

|

+----java.awt.Polygon

```
public class Polygon
    extends Object
```

A polygon consists of a list of x and y coordinates.

Error Index

Error Index

npoints

The total number of points.

Error Index

xpoints

The array of x coordinates.

Error Index

ypoints

The array of y coordinates.

Error Index

Error Index

Polygon()

Creates an empty polygon.

Error Index

Polygon(int[], int[], int)

Constructs and initializes a Polygon from the specified parameters.

Error Index

Error Index

addPoint(int, int)

Appends a point to a polygon.

Error Index

getBoundingBox()

Determines the area spanned by this Polygon.

Error Index

inside(int, int)

Determines whether the point (x,y) is inside the Polygon.

Error Index

Error Index

npoints

public int npoints

The total number of points.

Error Index

xpoints

public int xpoints[]

The array of x coordinates.

Error Index

ypoints

public int ypoints[]

The array of y coordinates.

Error Index

Error Index

Polygon

public Polygon()

Creates an empty polygon.

Error Index Polygon

```
public Polygon(int xpoints[],
               int ypoints[],
               int npoints)
```

Constructs and initializes a Polygon from the specified parameters.

Parameters:

xpoints - the array of x coordinates
ypoints - the array of y coordinates
npoints - the total number of points in the Polygon

Error Index

Error Index addPoint

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

Appends a point to a polygon. If inside(x, y) or another operation that calculates the bounding box has already been performed, this method updates the bounds accordingly.

Parameters:

x - the x coordinate of the point
y - the y coordinate of the point

Error Index getBoundingBox

```
public Rectangle getBoundingBox()
```

Determines the area spanned by this Polygon.

Returns:

a Rectangle defining the bounds of the Polygon.

Error Index inside


```
public boolean inside(int x,  
                      int y)
```

Determines whether the point (x,y) is inside the Polygon. Uses an even-odd insideness rule (also known as an alternating rule).

Parameters:

x - the X coordinate of the point to be tested
y - the Y coordinate of the point to be tested

Based on code by Hanpeter van Vliet .

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Rectangle

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Rectangle

[java.lang.Object](#)

|

+----java.awt.Rectangle

```
public class Rectangle
    extends Object
```

A rectangle defined by x, y, width and height.

Error Index

Error Index

height

The height of the rectangle.

Error Index

width

The width of the rectangle.

Error Index

x

The x coordinate of the rectangle.

Error Index

y

The y coordinate of the rectangle.

Error Index

Error Index

Rectangle()

Constructs a new rectangle.

Error Index

Rectangle(int, int, int, int)

Constructs and initializes a rectangle with the specified parameters.

Error Index

Rectangle(int, int)

Constructs a rectangle and initializes it with the specified width and height parameters.

Error Index

Rectangle(Point, Dimension)

Constructs a rectangle and initializes it to a specified point and dimension.

Error Index

Rectangle(Point)

Constructs a rectangle and initializes it to the specified point.

Error Index

Rectangle(Dimension)

Constructs a rectangle and initializes it to the specified width and height.

Error Index

Error Index

add(int, int)

Adds a point to a rectangle.

Error Index

add(Point)

Adds a point to a rectangle.

Error Index

add(Rectangle)

Adds a rectangle to a rectangle.

Error Index

equals(Object)

Checks whether two rectangles are equal.

Error Index

grow(int, int)

Grows the rectangle horizontally and vertically.

Error Index

hashCode()

Returns the hashcode for this Rectangle.

Error Index

inside(int, int)

Checks if the specified point lies inside a rectangle.

Error Index

intersection(Rectangle)

Computes the intersection of two rectangles.

Error Index

intersects(Rectangle)

Checks if two rectangles intersect.

Error Index

isEmpty()

Determines whether the rectangle is empty.

Error Index

move(int, int)

Moves the rectangle.

Error Index

reshape(int, int, int, int)

Reshapes the rectangle.

Error Index

resize(int, int)

Resizes the rectangle.

Error Index

toString()

Returns the String representation of this Rectangle's values.

Error Index

translate(int, int)

Translates the rectangle.

Error Index

union(Rectangle)

Computes the union of two rectangles.

Error Index

Error Index_x

public int x

The x coordinate of the rectangle.

Error Index_y

public int y

The y coordinate of the rectangle.

Error Index_{width}

public int width

The width of the rectangle.

Error Index height

```
public int height
```

The height of the rectangle.

Error Index

Error Index Rectangle

```
public Rectangle()
```

Constructs a new rectangle.

Error Index Rectangle

```
public Rectangle(int x,  
                 int y,  
                 int width,  
                 int height)
```

Constructs and initializes a rectangle with the specified parameters.

Parameters:

x - the x coordinate
y - the y coordinate
width - the width of the rectangle
height - the height of the rectangle

Error Index Rectangle

```
public Rectangle(int width,  
                 int height)
```

Constructs a rectangle and initializes it with the specified width and height parameters.

Parameters:

width - the width of the rectangle
height - the height of the rectangle

Error Index Rectangle

```
public Rectangle(Point p,  
    Dimension d)
```

Constructs a rectangle and initializes it to a specified point and dimension.

Parameters:

p - the point
d - dimension

Error Index Rectangle

```
public Rectangle(Point p)
```

Constructs a rectangle and initializes it to the specified point.

Parameters:

p - the value of the x and y coordinate

Error Index Rectangle

```
public Rectangle(Dimension d)
```

Constructs a rectangle and initializes it to the specified width and height.

Parameters:

d - the value of the width and height

Error Index

Error Index reshape

```
public void reshape(int x,  
    int y,  
    int width,  
    int height)
```

Reshapes the rectangle.

Error Index **move**

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

Moves the rectangle.

Error Index **translate**

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

Translates the rectangle.

Error Index **resize**

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

Resizes the rectangle.

Error Index **inside**

```
public boolean inside(int x,  
                    int y)
```

Checks if the specified point lies inside a rectangle.

Parameters:

x - the x coordinate
y - the y coordinate

Error Index **intersects**

```
public boolean intersects(Rectangle r)
```

Checks if two rectangles intersect.

Error Index **intersection**

```
public Rectangle intersection(Rectangle r)
```

Computes the intersection of two rectangles.

Error Index union

```
public Rectangle union(Rectangle r)
```

Computes the union of two rectangles.

Error Index add

```
public void add(int newx,  
               int newy)
```

Adds a point to a rectangle. This results in the smallest rectangle that contains both the rectangle and the point.

Error Index add

```
public void add(Point pt)
```

Adds a point to a rectangle. This results in the smallest rectangle that contains both the rectangle and the point.

Error Index add

```
public void add(Rectangle r)
```

Adds a rectangle to a rectangle. This results in the union of the two rectangles.

Error Index grow

```
public void grow(int h,  
               int v)
```

Grows the rectangle horizontally and vertically.

Error Index isEmpty


```
public boolean isEmpty()
```

Determines whether the rectangle is empty.

Error Index hashCode

```
public int hashCode()
```

Returns the hashCode for this Rectangle.

Overrides:

hashCode in class Object

Error Index equals

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

Checks whether two rectangles are equal.

Overrides:

equals in class Object

Error Index toString

```
public String toString()
```

Returns the String representation of this Rectangle's values.

Overrides:

toString in class Object

Class java.awt.Scrollbar

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Scrollbar

```
java.lang.Object
|
+----java.awt.Component
|
+----java.awt.Scrollbar
```

```
public class Scrollbar
    extends Component
```

A Scrollbar component.

Error Index

Error Index

HORIZONTAL

The horizontal Scrollbar variable.

Error Index

VERTICAL

The vertical Scrollbar variable.

Error Index

Error Index

Scrollbar()

Constructs a new vertical Scrollbar.

Error Index

Scrollbar(int)

Constructs a new Scrollbar with the specified orientation.

Error Index

Scrollbar(int, int, int, int, int)

Constructs a new Scrollbar with the specified orientation, value, page size, and minimum and maximum values.

Error Index

Error Index

addNotify()

Creates the Scrollbar's peer.

Error Index

getLineIncrement()

Gets the line increment for this scrollbar.

Error Index

getMaximum()

Returns the maximum value of this Scrollbar.

Error Index

getMinimum()

Returns the minimum value of this Scrollbar.

Error Index

getOrientation()

Returns the orientation for this Scrollbar.

Error Index

getPageIncrement()

Gets the page increment for this scrollbar.

Error Index

getValue()

Returns the current value of this Scrollbar.

Error Index

getVisible()

Returns the visible amount of the Scrollbar.

Error Index

paramString()

Returns the String parameters for this Scrollbar.

Error Index

setLineIncrement(int)

Sets the line increment for this scrollbar.

Error Index

setPageIncrement(int)

Sets the page increment for this scrollbar.

Error Index

setValue(int)

Sets the value of this Scrollbar to the specified value.

Error Index

setValues(int, int, int, int)

Sets the values for this Scrollbar.

Error Index

Error Index HORIZONTAL

```
public final static int HORIZONTAL
```

The horizontal Scrollbar variable.

Error Index VERTICAL

```
public final static int VERTICAL
```

The vertical Scrollbar variable.

Error Index

Error Index Scrollbar

```
public Scrollbar()
```

Constructs a new vertical Scrollbar.

Error Index Scrollbar

```
public Scrollbar(int orientation)
```

Constructs a new Scrollbar with the specified orientation.

Parameters:

orientation - either Scrollbar.HORIZONTAL or Scrollbar.VERTICAL

Throws: IllegalArgumentException

When an illegal scrollbar orientation is given.

Error Index Scrollbar

```
public Scrollbar(int orientation,  
                int value,  
                int visible,
```

```
int minimum,  
int maximum)
```

Constructs a new Scrollbar with the specified orientation, value, page size, and minimum and maximum values.

Parameters:

orientation - either Scrollbar.HORIZONTAL or Scrollbar.VERTICAL

value - the scrollbar's value

visible - the size of the visible portion of the scrollable area. The scrollbar will use this value when paging up or down by a page.

minimum - the minimum value of the scrollbar

maximum - the maximum value of the scrollbar

Error Index

Error Index

addNotify

```
public synchronized void addNotify()
```

Creates the Scrollbar's peer. The peer allows you to modify the appearance of the Scrollbar without changing any of its functionality.

Overrides:

addNotify in class Component

Error Index

getOrientation

```
public int getOrientation()
```

Returns the orientation for this Scrollbar.

Error Index

getValue

```
public int getValue()
```

Returns the current value of this Scrollbar.

See Also:

getMinimum, getMaximum

Error Index setValue

```
public void setValue(int value)
```

Sets the value of this Scrollbar to the specified value.

Parameters:

value - the new value of the Scrollbar. If this value is below the current minimum or above the current maximum, it becomes the new one of those values, respectively.

See Also:

[getValue](#)

Error Index getMinimum

```
public int getMinimum()
```

Returns the minimum value of this Scrollbar.

See Also:

[getMaximum](#), [getValue](#)

Error Index getMaximum

```
public int getMaximum()
```

Returns the maximum value of this Scrollbar.

See Also:

[getMinimum](#), [getValue](#)

Error Index getVisible

```
public int getVisible()
```

Returns the visible amount of the Scrollbar.

Error Index setLineIncrement

```
public void setLineIncrement(int l)
```

Sets the line increment for this scrollbar. This is the value that will be added (subtracted) when the user hits the line down (up) gadgets.

Error Index **getLineIncrement**

```
public int getLineIncrement()
```

Gets the line increment for this scrollbar.

Error Index **setPageIncrement**

```
public void setPageIncrement(int l)
```

Sets the page increment for this scrollbar. This is the value that will be added (subtracted) when the user hits the page down (up) gadgets.

Error Index **getPageIncrement**

```
public int getPageIncrement()
```

Gets the page increment for this scrollbar.

Error Index **setValues**

```
public void setValues(int value,  
    int visible,  
    int minimum,  
    int maximum)
```

Sets the values for this Scrollbar.

Parameters:

value - is the position in the current window.
visible - is the amount visible per page
minimum - is the minimum value of the scrollbar
maximum - is the maximum value of the scrollbar

Error Index **paramString**

```
protected String paramString()
```

Returns the String parameters for this Scrollbar.

Overrides:

paramString in class Component

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.TextArea

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.TextArea

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.TextComponent
            |
            +----java.awt.TextArea
```

```
public class TextArea
    extends TextComponent
```

A TextArea object is a multi-line area that displays text. It can be set to allow editing or read-only modes.

Error Index

Error Index

[TextArea\(\)](#)

Constructs a new TextArea.

Error Index

[TextArea\(int, int\)](#)

Constructs a new TextArea with the specified number of rows and columns.

Error Index

[TextArea\(String\)](#)

Constructs a new TextArea with the specified text displayed.

Error Index

[TextArea\(String, int, int\)](#)

Constructs a new TextArea with the specified text and number of rows and columns.

Error Index

Error Index

[addNotify\(\)](#)

Creates the TextArea's peer.

Error Index

appendText(String)

Appends the given text to the end.

Error Index

getColumnns()

Returns the number of columns in the TextArea.

Error Index

getRows()

Returns the number of rows in the TextArea.

Error Index

insertText(String, int)

Inserts the specified text at the specified position.

Error Index

minimumSize(int, int)

Returns the specified minimum size Dimensions of the TextArea.

Error Index

minimumSize()

Returns the minimum size Dimensions of the TextArea.

Error Index

paramString()

Returns the String of parameters for this TextArea.

Error Index

preferredSize(int, int)

Returns the specified row and column Dimensions of the TextArea.

Error Index

preferredSize()

Returns the preferred size Dimensions of the TextArea.

Error Index

replaceText(String, int, int)

Replaces text from the indicated start to end position with the new text specified.

Error Index

Error Index

TextArea

public TextArea()

Constructs a new TextArea.

Error Index

TextArea

public TextArea(int rows,

int cols)

Constructs a new TextArea with the specified number of rows and columns.

Parameters:

rows - the number of rows

cols - the number of columns

Error Index **TextArea**

public TextArea(String text)

Constructs a new TextArea with the specified text displayed.

Parameters:

text - the text to be displayed

Error Index **TextArea**

public TextArea(String text,
int rows,
int cols)

Constructs a new TextArea with the specified text and number of rows and columns.

Parameters:

text - the text to be displayed

rows - the number of rows

cols - the number of cols

Error Index

Error Index **addNotify**

public synchronized void addNotify()

Creates the TextArea's peer. The peer allows us to modify the appearance of the TextArea without changing any of its functionality.

Overrides:

addNotify in class Component

Error Index

insertText

```
public void insertText(String str,  
                      int pos)
```

Inserts the specified text at the specified position.

Parameters:

str - the text to insert.

pos - the position at which to insert.

See Also:

setText, replaceText

Error Index

appendText

```
public void appendText(String str)
```

Appends the given text to the end.

Parameters:

str - the text to insert

See Also:

insertText

Error Index

replaceText

```
public void replaceText(String str,  
                       int start,  
                       int end)
```

Replaces text from the indicated start to end position with the new text specified.

Parameters:

str - the text to use as the replacement.

start - the start position.

end - the end position.

See Also:

insertText, replaceText

Error Index

getRows

```
public int getRows()
```

Returns the number of rows in the TextArea.

Error Index `getColumns`

```
public int getColumns()
```

Returns the number of columns in the TextArea.

Error Index `preferredSize`

```
public Dimension preferredSize(int rows,  
                                int cols)
```

Returns the specified row and column Dimensions of the TextArea.

Parameters:

rows - the preferred rows amount

cols - the preferred columns amount

Error Index `preferredSize`

```
public Dimension preferredSize()
```

Returns the preferred size Dimensions of the TextArea.

Overrides:

preferredSize in class Component

Error Index `minimumSize`

```
public Dimension minimumSize(int rows,  
                                int cols)
```

Returns the specified minimum size Dimensions of the TextArea.

Parameters:

rows - the minimum row size

cols - the minimum column size

Error Index `minimumSize`

public Dimension minimumSize()

Returns the minimum size Dimensions of the TextArea.

Overrides:

minimumSize in class Component

Error Index `paramString`

protected String paramString()

Returns the String of parameters for this TextArea.

Overrides:

paramString in class TextComponent

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.TextComponent

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.TextComponent

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.TextComponent
```

```
public class TextComponent
    extends Component
```

A TextComponent is a component that allows the editing of some text.

Error Index

Error Index

[getSelectedText\(\)](#)

Returns the selected text contained in this TextComponent.

Error Index

[getSelectionEnd\(\)](#)

Returns the selected text's end position.

Error Index

[getSelectionStart\(\)](#)

Returns the selected text's start position.

Error Index

[getText\(\)](#)

Returns the text contained in this TextComponent.

Error Index

[isEditable\(\)](#)

Returns the boolean indicating whether this TextComponent is editable or not.

Error Index

[paramString\(\)](#)

Returns the String of parameters for this TextComponent.

Error Index

[removeNotify\(\)](#)

Removes the TextComponent's peer.

Error Index

select(int, int)

Selects the text found between the specified start and end locations.

Error Index

selectAll()

Selects all the text in the TextComponent.

Error Index

setEditable(boolean)

Sets the specified boolean to indicate whether or not this TextComponent should be editable.

Error Index

setText(String)

Sets the text of this TextComponent to the specified text.

Error Index

Error Index

removeNotify

public synchronized void removeNotify()

Removes the TextComponent's peer. The peer allows us to modify the appearance of the TextComponent without changing its functionality.

Overrides:

removeNotify in class Component

Error Index

setText

public void setText(String t)

Sets the text of this TextComponent to the specified text.

Parameters:

t - the new text to be set

See Also:

getText

Error Index

getText

public String getText()

Returns the text contained in this TextComponent.

See Also:
[setText](#)

Error Index **getSelectedText**

```
public String getSelectedText()
```

Returns the selected text contained in this TextComponent.

See Also:
[setText](#)

Error Index **isEditable**

```
public boolean isEditable()
```

Returns the boolean indicating whether this TextComponent is editable or not.

See Also:
[setEditable](#)

Error Index **setEditable**

```
public void setEditable(boolean t)
```

Sets the specified boolean to indicate whether or not this TextComponent should be editable.

Parameters:
t - the boolean to be set

See Also:
[isEditable](#)

Error Index **getSelectionStart**

```
public int getSelectionStart()
```

Returns the selected text's start position.

Error Index **getSelectionEnd**

```
public int getSelectionEnd()
```

Returns the selected text's end position.

Error Index **select**

```
public void select(int selStart,  
                  int selEnd)
```

Selects the text found between the specified start and end locations.

Parameters:

selStart - the start position of the text

selEnd - the end position of the text

Error Index **selectAll**

```
public void selectAll()
```

Selects all the text in the TextComponent.

Error Index **paramString**

```
protected String paramString()
```

Returns the String of parameters for this TextComponent.

Overrides:

paramString in class Component

Class java.awt.TextField

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.TextField

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.TextComponent
            |
            +----java.awt.TextField
```

```
public class TextField
    extends TextComponent
```

TextField is a component that allows the editing of a single line of text.

Error Index

Error Index

[TextField\(\)](#)

Constructs a new TextField.

Error Index

[TextField\(int\)](#)

Constructs a new TextField initialized with the specified columns.

Error Index

[TextField\(String\)](#)

Constructs a new TextField initialized with the specified text.

Error Index

[TextField\(String, int\)](#)

Constructs a new TextField initialized with the specified text and columns.

Error Index

Error Index

[addNotify\(\)](#)

Creates the TextField's peer.

Error Index

echoCharIsSet()

Returns true if this TextField has a character set for echoing.

Error Index

getColumnns()

Returns the number of columns in this TextField.

Error Index

getEchoChar()

Returns the character to be used for echoing.

Error Index

minimumSize(int)

Returns the minimum size Dimensions needed for this TextField with the specified amount of columns.

Error Index

minimumSize()

Returns the minimum size Dimensions needed for this TextField.

Error Index

paramString()

Returns the String of parameters for this TExtField.

Error Index

preferredSize(int)

Returns the preferred size Dimensions needed for this TextField with the specified amount of columns.

Error Index

preferredSize()

Returns the preferred size Dimensions needed for this TextField.

Error Index

setEchoCharacter(char)

Sets the echo character for this TextField.

Error Index

Error Index

TextField

public TextField()

Constructs a new TextField.

Error Index

TextField

public TextField(int cols)

Constructs a new TextField initialized with the specified columns.

Parameters:

cols - the number of columns

Error Index TextField

```
public TextField(String text)
```

Constructs a new TextField initialized with the specified text.

Parameters:

text - the text to be displayed

Error Index TextField

```
public TextField(String text,  
                int cols)
```

Constructs a new TextField initialized with the specified text and columns.

Parameters:

text - the text to be displayed

cols - the number of columns

Error Index

Error Index addNotify

```
public synchronized void addNotify()
```

Creates the TextField's peer. The peer allows us to modify the appearance of the TextField without changing its functionality.

Overrides:

addNotify in class Component

Error Index getEchoChar

```
public char getEchoChar()
```

Returns the character to be used for echoing.

See Also:

setEchoCharacter, echoCharIsSet

Error Index `echoCharIsSet`

```
public boolean echoCharIsSet()
```

Returns true if this TextField has a character set for echoing.

See Also:

setEchoCharacter, getEchoChar

Error Index `getColumns`

```
public int getColumns()
```

Returns the number of columns in this TextField.

Error Index `setEchoCharacter`

```
public void setEchoCharacter(char c)
```

Sets the echo character for this TextField. This is useful for fields where the user input shouldn't be echoed to the screen, as in the case of a TextField that represents a password.

Parameters:

c - the echo character for this TextField

See Also:

echoCharIsSet, getEchoChar

Error Index `preferredSize`

```
public Dimension preferredSize(int cols)
```

Returns the preferred size Dimensions needed for this TextField with the specified amount of columns.

Parameters:

cols - the number of columns in this TextField

Error Index preferredSize

public Dimension preferredSize()

Returns the preferred size Dimensions needed for this TextField.

Overrides:

preferredSize in class Component

Error Index minimumSize

public Dimension minimumSize(int cols)

Returns the minimum size Dimensions needed for this TextField with the specified amount of columns.

Parameters:

cols - the number of columns in this TextField

Error Index minimumSize

public Dimension minimumSize()

Returns the minimum size Dimensions needed for this TextField.

Overrides:

minimumSize in class Component

Error Index paramString

protected String paramString()

Returns the String of parameters for this TextField.

Overrides:

paramString in class TextComponent

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Toolkit

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Toolkit

[java.lang.Object](#)
|
+----java.awt.Toolkit

```
public class Toolkit  
extends Object
```

An AWT toolkit. It is used to bind the abstract AWT classes to a particular native toolkit implementation.

Error Index

Error Index

[Toolkit\(\)](#)

Error Index

Error Index

[checkImage](#)(Image, int, int, ImageObserver)

Returns the status of the construction of the indicated method at the indicated width and height for the default screen.

Error Index

[createButton](#)(Button)

Uses the specified Peer interface to create a new Button.

Error Index

[createCanvas](#)(Canvas)

Uses the specified Peer interface to create a new Canvas.

Error Index

[createCheckbox](#)(Checkbox)

Uses the specified Peer interface to create a new Checkbox.

Error Index

[createCheckboxMenuItem](#)(CheckboxMenuItem)

Uses the specified Peer interface to create a new CheckboxMenuItem.

Error Index

createChoice(Choice)

Uses the specified Peer interface to create a new Choice.

Error Index

createDialog(Dialog)

Uses the specified Peer interface to create a new Dialog.

Error Index

createFileDialog(FileDialog)

Uses the specified Peer interface to create a new FileDialog.

Error Index

createFrame(Frame)

Uses the specified Peer interface to create a new Frame.

Error Index

createImage(ImageProducer)

Creates an image with the specified image producer.

Error Index

createLabel(Label)

Uses the specified Peer interface to create a new Label.

Error Index

createList(List)

Uses the specified Peer interface to create a new List.

Error Index

createMenu(Menu)

Uses the specified Peer interface to create a new Menu.

Error Index

createMenuBar(MenuBar)

Uses the specified Peer interface to create a new MenuBar.

Error Index

createMenuItem(MenuItem)

Uses the specified Peer interface to create a new MenuItem.

Error Index

createPanel(Panel)

Uses the specified Peer interface to create a new Panel.

Error Index

createScrollbar(Scrollbar)

Uses the specified Peer interface to create a new Scrollbar.

Error Index

createTextArea(TextArea)

Uses the specified Peer interface to create a new TextArea.

Error Index

createTextField(TextField)

Uses the specified Peer interface to create a new TextField.

Error Index

createWindow(Window)

Uses the specified Peer interface to create a new Window.

Error Index

getColorModel()

Returns the ColorModel of the screen.

Error Index

getDefaultToolkit()

Returns the default toolkit.

Error Index

getFontList()

Returns the names of the available fonts.

Error Index

getFontMetrics(Font)

Returns the screen metrics of the font.

Error Index

getImage(String)

Returns an image which gets pixel data from the specified file.

Error Index

getImage(URL)

Returns an image which gets pixel data from the specified URL.

Error Index

getScreenResolution()

Returns the screen resolution in dots-per-inch.

Error Index

getScreenSize()

Gets the size of the screen.

Error Index

prepareImage(Image, int, int, ImageObserver)

Prepares an image for rendering on the default screen at the specified width and height.

Error Index

sync()

Syncs the graphics state; useful when doing animation.

Error Index

Error Index

Toolkit

public Toolkit()

Error Index

Error Index

createButton

protected abstract ButtonPeer createButton(Button target)

Uses the specified Peer interface to create a new Button.

Parameters:

target - the Button to be created

Error Index createTextField

protected abstract TextFieldPeer createTextField(TextField target)

Uses the specified Peer interface to create a new TextField.

Parameters:

target - the TextField to be created

Error Index createLabel

protected abstract LabelPeer createLabel(Label target)

Uses the specified Peer interface to create a new Label.

Parameters:

target - the Label to be created

Error Index createList

protected abstract ListPeer createList(List target)

Uses the specified Peer interface to create a new List.

Parameters:

target - the List to be created

Error Index createCheckbox

protected abstract CheckboxPeer createCheckbox(Checkbox target)

Uses the specified Peer interface to create a new Checkbox.

Parameters:

target - the Checkbox to be created

Error Index createScrollbar

protected abstract ScrollbarPeer createScrollbar(Scrollbar target)

Uses the specified Peer interface to create a new Scrollbar.

Parameters:

target - the Scrollbar to be created

Error Index createTextArea

protected abstract TextAreaPeer createTextArea(TextArea target)

Uses the specified Peer interface to create a new TextArea.

Parameters:

target - the TextArea to be created

Error Index createChoice

protected abstract ChoicePeer createChoice(Choice target)

Uses the specified Peer interface to create a new Choice.

Parameters:

target - the Choice to be created

Error Index createFrame

protected abstract FramePeer createFrame(Frame target)

Uses the specified Peer interface to create a new Frame.

Parameters:

target - the Frame to be created

Error Index createCanvas

protected abstract CanvasPeer createCanvas(Canvas target)

Uses the specified Peer interface to create a new Canvas.

Parameters:

target - the Canvas to be created

Error Index createPanel

protected abstract PanelPeer createPanel(Panel target)

Uses the specified Peer interface to create a new Panel.

Parameters:

target - the Panel to be created

Error Index createWindow

protected abstract WindowPeer createWindow(Window target)

Uses the specified Peer interface to create a new Window.

Parameters:

target - the Window to be created

Error Index createDialog

protected abstract DialogPeer createDialog(Dialog target)

Uses the specified Peer interface to create a new Dialog.

Parameters:

target - the Dialog to be created

Error Index createMenuBar

protected abstract MenuBarPeer createMenuBar(MenuBar target)

Uses the specified Peer interface to create a new MenuBar.

Parameters:

target - the MenuBar to be created

Error Index createMenu

protected abstract MenuPeer createMenu(Menu target)

Uses the specified Peer interface to create a new Menu.

Parameters:

target - the Menu to be created

Error Index createMenuItem

protected abstract MenuItemPeer createMenuItem(MenuItem target)

Uses the specified Peer interface to create a new MenuItem.

Parameters:

target - the MenuItem to be created

Error Index createFileDialog

protected abstract FileDialogPeer createFileDialog(FileDialog target)

Uses the specified Peer interface to create a new FileDialog.

Parameters:

target - the FileDialog to be created

Error Index createCheckboxMenuItem

protected abstract CheckboxMenuItemPeer createCheckboxMenuItem(CheckboxMenuItem target)

Uses the specified Peer interface to create a new CheckboxMenuItem.

Parameters:

target - the CheckboxMenuItem to be created

Error Index **getScreenSize**

```
public abstract Dimension getScreenSize()
```

Gets the size of the screen.

Error Index **getScreenResolution**

```
public abstract int getScreenResolution()
```

Returns the screen resolution in dots-per-inch.

Error Index **getColorModel**

```
public abstract ColorModel getColorModel()
```

Returns the ColorModel of the screen.

Error Index **getFontList**

```
public abstract String[] getFontList()
```

Returns the names of the available fonts.

Error Index **getFontMetrics**

```
public abstract FontMetrics getFontMetrics(Font font)
```

Returns the screen metrics of the font.

Error Index **sync**

```
public abstract void sync()
```

Syncs the graphics state; useful when doing animation.

Error Index

getDefaultToolkit

public static synchronized Toolkit getDefaultToolkit()

Returns the default toolkit. This is controlled by the "awt.toolkit" property.

Throws: AWTError

Toolkit not found or could not be instantiated.

Error Index

getImage

public abstract Image getImage(String filename)

Returns an image which gets pixel data from the specified file.

Parameters:

filename - the file containing the pixel data in one of the recognized file formats

Error Index

getImage

public abstract Image getImage(URL url)

Returns an image which gets pixel data from the specified URL.

Parameters:

url - the URL to use in fetching the pixel data

Error Index

prepareImage

public abstract boolean prepareImage(Image image,
int width,
int height,
ImageObserver observer)

Prepares an image for rendering on the default screen at the specified width and height.

Error Index

checkImage

public abstract int checkImage(Image image,
int width,

```
int height,  
ImageObserver observer)
```

Returns the status of the construction of the indicated method at the indicated width and height for the default screen.

Error Index **createImage**

```
public abstract Image createImage(ImageProducer producer)
```

Creates an image with the specified image producer.

Parameters:

producer - the image producer to be used

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Window

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.awt.Window

```
java.lang.Object
|
+----java.awt.Component
      |
      +----java.awt.Container
            |
            +----java.awt.Window
```

```
public class Window
    extends Container
```

A Window is a top-level window with no borders and no menubar. It could be used to implement a pop-up menu. The default layout for a window is BorderLayout.

Error Index

Error Index

[Window](#)(Frame)

Constructs a new Window initialized to an invisible state.

Error Index

Error Index

[addNotify](#)()

Creates the Window's peer.

Error Index

[dispose](#)()

Disposes of the Window.

Error Index

[getToolkit](#)()

Returns the toolkit of this frame.

Error Index

[getWarningString](#)()

Gets the warning string for this window.

Error Index

pack()

Packs the components of the Window.

Error Index

show()

Shows the Window.

Error Index

toBack()

Sends the frame to the back of the Window.

Error Index

toFront()

Brings the frame to the front of the Window.

Error Index

Error Index

Window

public Window(Frame parent)

Constructs a new Window initialized to an invisible state. It behaves as a modal dialog in that it will block input to other windows when shown.

Parameters:

parent - the owner of the dialog

See Also:

resize, show

Error Index

Error Index

addNotify

public synchronized void addNotify()

Creates the Window's peer. The peer allows us to modify the appearance of the Window without changing its functionality.

Overrides:

addNotify in class Container

Error Index pack

public synchronized void pack()

Packs the components of the Window.

Error Index show

public synchronized void show()

Shows the Window. This will bring the window to the front if the window is already visible.

Overrides:

show in class Component

See Also:

hide

Error Index dispose

public synchronized void dispose()

Disposes of the Window. This method must be called to release the resources that are used for the window.

Error Index toFront

public void toFront()

Brings the frame to the front of the Window.

Error Index toBack

public void toBack()

Sends the frame to the back of the Window.

Error Index getToolkit

public Toolkit getToolkit()

Returns the toolkit of this frame.

Overrides:

getToolkit in class Component

See Also:

Toolkit

Error Index `getWarningString`

```
public final String getWarningString()
```

Gets the warning string for this window. This is a string that will be displayed somewhere in the visible area of windows that are not secure.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.BufferedInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.BufferedInputStream

```
java.lang.Object
|
+----java.io.InputStream
      |
      +----java.io.FilterInputStream
            |
            +----java.io.BufferedInputStream
```

```
public class BufferedInputStream
    extends FilterInputStream
```

A buffered input stream. This stream lets you read in characters from a stream without causing a read every time. The data is read into a buffer, subsequent reads result in a fast buffer access.

Error Index

Error Index

buf

The buffer where data is stored.

Error Index

count

The number of bytes in the buffer.

Error Index

marklimit

The maximum readahead allowed after a mark() before subsequent calls to reset() fail.

Error Index

markpos

The position in the buffer of the current mark.

Error Index

pos

The current position in the buffer.

Error Index

Error Index

BufferedInputStream(InputStream)

Creates a new buffered stream with a default buffer size.

Error Index

BufferedInputStream(InputStream, int)

Creates a new buffered stream with the specified buffer size.

Error Index

Error Index

available()

Returns the number of bytes that can be read without blocking.

Error Index

mark(int)

Marks the current position in the input stream.

Error Index

markSupported()

Returns a boolean indicating if this stream type supports mark/reset.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

reset()

Repositions the stream to the last marked position.

Error Index

skip(long)

Skips n bytes of input.

Error Index

Error Index

buf

protected byte buf[]

The buffer where data is stored.

Error Indexcount

protected int count

The number of bytes in the buffer.

Error Indexpos

protected int pos

The current position in the buffer.

Error Indexmarkpos

protected int markpos

The position in the buffer of the current mark. This mark is set to -1 if there is no current mark.

Error Indexmarklimit

protected int marklimit

The maximum readahead allowed after a mark() before subsequent calls to reset() fail.

Error Index

Error IndexBufferedInputStream

public BufferedInputStream(InputStream in)

Creates a new buffered stream with a default buffer size.

Parameters:

in - the input stream

Error IndexBufferedInputStream

```
public BufferedInputStream(InputStream in,  
                           int size)
```

Creates a new buffered stream with the specified buffer size.

Parameters:

in - the input stream
size - the buffer size

Error Index

Error Index read

```
public synchronized int read() throws IOException
```

Reads a byte of data. This method will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index read

```
public synchronized int read(byte b[],  
                             int off,  
                             int len) throws IOException
```

Reads into an array of bytes. Blocks until some input is available.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index skip

public synchronized long skip(long n) throws IOException

Skips n bytes of input.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Throws: IOException

If an I/O error has occurred.

Overrides:

skip in class FilterInputStream

Error Index available

public synchronized int available() throws IOException

Returns the number of bytes that can be read without blocking. This total is the number of bytes in the buffer and the number of bytes available from the input stream.

Returns:

the number of available bytes.

Overrides:

available in class FilterInputStream

Error Index mark

public synchronized void mark(int readlimit)

Marks the current position in the input stream. A subsequent call to the reset() method will reposition the stream at the last marked position so that subsequent reads will re-read the same bytes. The stream promises to allow readlimit bytes to be read before the mark position gets invalidated.

Parameters:

readlimit - the maximum limit of bytes allowed to be read before the mark position becomes invalid.

Overrides:

mark in class FilterInputStream

Error Index reset

public synchronized void reset() throws [IOException](#)

Repositions the stream to the last marked position. If the stream has not been marked, or if the mark has been invalidated, an [IOException](#) is thrown. Stream marks are intended to be used in situations where you need to read ahead a little to see what's in the stream. Often this is most easily done by invoking some general parser. If the stream is of the type handled by the parser, it just chugs along happily. If the stream is not of that type, the parser should toss an exception when it fails. If an exception gets tossed within readlimit bytes, the parser will allow the outer code to reset the stream and to try another parser.

Throws: [IOException](#)

If the stream has not been marked or if the mark has been invalidated.

Overrides:

[reset](#) in class [FilterInputStream](#)

Error Index

markSupported

public boolean markSupported()

Returns a boolean indicating if this stream type supports mark/reset.

Overrides:

[markSupported](#) in class [FilterInputStream](#)

Class java.io.BufferedOutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.BufferedOutputStream

```
java.lang.Object
|
+----java.io.OutputStream
      |
      +----java.io.FilterOutputStream
            |
            +----java.io.BufferedOutputStream
```

```
public class BufferedOutputStream
    extends FilterOutputStream
```

A buffered output stream. This stream lets you write characters to a stream without causing a write every time. The data is first written into a buffer. Data is written to the actual stream only when the buffer is full, or when the stream is flushed.

Error Index

Error Index

buf

The buffer where data is stored.

Error Index

count

The number of bytes in the buffer.

Error Index

Error Index

BufferedOutputStream(OutputStream)

Creates a new buffered stream with a default buffer size.

Error Index

BufferedOutputStream(OutputStream, int)

Creates a new buffered stream with the specified buffer size.

Error Index

Error Index

flush()

Flushes the stream.

Error Index

write(int)

Writes a byte.

Error Index

write(byte[], int, int)

Writes a subarray of bytes.

Error Index

Error Index

buf

protected byte buf[]

The buffer where data is stored.

Error Index

count

protected int count

The number of bytes in the buffer.

Error Index

Error Index

BufferedOutputStream

public BufferedOutputStream(OutputStream out)

Creates a new buffered stream with a default buffer size.

Parameters:

out - the output stream

Error Index

BufferedOutputStream

```
public BufferedOutputStream(OutputStream out,  
                           int size)
```

Creates a new buffered stream with the specified buffer size.

Parameters:

out - the output stream
size - the buffer size

Error Index

Error Index

write

```
public synchronized void write(int b) throws IOException
```

Writes a byte. This method will block until the byte is actually written.

Parameters:

b - the byte to be written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class FilterOutputStream

Error Index

write

```
public synchronized void write(byte b[],  
                               int off,  
                               int len) throws IOException
```

Writes a subarray of bytes.

Parameters:

b - the data to be written
off - the start offset in the data
len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class FilterOutputStream

Error Index flush

public synchronized void flush() throws [IOException](#)

Flushes the stream. This will write any buffered output bytes.

Throws: [IOException](#)

If an I/O error has occurred.

Overrides:

[flush](#) in class [FilterOutputStream](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.ByteArrayInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.ByteArrayInputStream

[java.lang.Object](#)

```
|
+----java.io.InputStream
      |
      +----java.io.ByteArrayInputStream
```

```
public class ByteArrayInputStream
    extends InputStream
```

This class implements a buffer that can be used as an InputStream.

Error Index

Error Index

buf

The buffer where data is stored.

Error Index

count

The number of characters to use in the buffer.

Error Index

pos

The current position in the buffer.

Error Index

Error Index

ByteArrayInputStream(byte[])

Creates an ByteArrayInputStream from the specified array of bytes.

Error Index

ByteArrayInputStream(byte[], int, int)

Creates an ByteArrayInputStream from the specified array of bytes.

Error Index

Error Index

available()

Returns the number of available bytes in the buffer.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

reset()

Resets the buffer to the beginning.

Error Index

skip(long)

Skips n bytes of input.

Error Index

Error Index_{buf}

protected byte buf[]

The buffer where data is stored.

Error Index_{pos}

protected int pos

The current position in the buffer.

Error Index_{count}

protected int count

The number of characters to use in the buffer.

Error Index

Error Index

ByteArrayInputStream

```
public ByteArrayInputStream(byte buf[])
```

Creates an ByteArrayInputStream from the specified array of bytes.

Parameters:

buf - the input buffer (not copied)

Error Index

ByteArrayInputStream

```
public ByteArrayInputStream(byte buf[],  
                             int offset,  
                             int length)
```

Creates an ByteArrayInputStream from the specified array of bytes.

Parameters:

buf - the input buffer (not copied)

offset - the offset of the first byte to read

length - the number of bytes to read

Error Index

Error Index

read

```
public synchronized int read()
```

Reads a byte of data.

Returns:

the byte read, or -1 if the end of the stream is reached.

Overrides:

read in class InputStream

Error Index

read

```
public synchronized int read(byte b[],
                             int off,
                             int len)
```

Reads into an array of bytes.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read; -1 is returned when the end of the stream is reached.

Overrides:

read in class InputStream

Error Index skip

```
public synchronized long skip(long n)
```

Skips n bytes of input.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Overrides:

skip in class InputStream

Error Index available

```
public synchronized int available()
```

Returns the number of available bytes in the buffer.

Overrides:

available in class InputStream

Error Index reset

```
public synchronized void reset()
```

Resets the buffer to the beginning.

Overrides:

reset in class InputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.ByteArrayOutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.ByteArrayOutputStream

[java.lang.Object](#)

```
|
+----java.io.OutputStream
      |
      +----java.io.ByteArrayOutputStream
```

```
public class ByteArrayOutputStream
    extends OutputStream
```

This class implements a buffer that can be used as an OutputStream. The buffer automatically grows when data is written to the stream. The data can be retrieved using `toByteArray()` and `toString()`.

Error Index

Error Index

buf

The buffer where data is stored.

Error Index

count

The number of bytes in the buffer.

Error Index

Error Index

ByteArrayOutputStream()

Creates a new ByteArrayOutputStream.

Error Index

ByteArrayOutputStream(int)

Creates a new ByteArrayOutputStream with the specified initial size.

Error Index

Error Index

reset()

Resets the buffer so that you can use it again without throwing away the already allocated buffer.

Error Index

size()

Returns the current size of the buffer.

Error Index

toByteArray()

Returns a copy of the input data.

Error Index

toString()

Converts input data to a string.

Error Index

toString(int)

Converts input data to a string.

Error Index

write(int)

Writes a byte to the buffer.

Error Index

write(byte[], int, int)

Writes bytes to the buffer.

Error Index

writeTo(OutputStream)

Writes the contents of the buffer to another stream.

Error Index

Error Index

buf

protected byte buf[]

The buffer where data is stored.

Error Index

count

protected int count

The number of bytes in the buffer.

Error Index

Error Index

ByteArrayOutputStream

```
public ByteArrayOutputStream()
```

Creates a new ByteArrayOutputStream.

Error Index

ByteArrayOutputStream

```
public ByteArrayOutputStream(int size)
```

Creates a new ByteArrayOutputStream with the specified initial size.

Parameters:

size - the initial size

Error Index

Error Index

write

```
public synchronized void write(int b)
```

Writes a byte to the buffer.

Parameters:

b - the byte

Overrides:

write in class OutputStream

Error Index

write

```
public synchronized void write(byte b[],  
                                int off,  
                                int len)
```

Writes bytes to the buffer.

Parameters:

b - the data to be written

off - the start offset in the data
len - the number of bytes that are written

Overrides:

write in class OutputStream

Error Index **writeTo**

public synchronized void writeTo(OutputStream out) throws IOException

Writes the contents of the buffer to another stream.

Parameters:

out - the output stream to write to

Error Index **reset**

public synchronized void reset()

Resets the buffer so that you can use it again without throwing away the already allocated buffer.

Error Index **toByteArray**

public synchronized byte[] toByteArray()

Returns a copy of the input data.

Error Index **size**

public int size()

Returns the current size of the buffer.

Error Index **toString**

public String toString()

Converts input data to a string.

Returns:

the string.

Overrides:

toString in class Object

Error Index **toString**

```
public String toString(int hibernate)
```

Converts input data to a string. The top 8 bits of each 16 bit Unicode character are set to hibernate.

Parameters:

hibernate - the bits set

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.io.DataInput

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.io.DataInput

public interface **DataInput**
extends [Object](#)

DataInput is an interface describing streams that can read input in a machine-independent format.

See Also:

[DataInputStream](#), [DataOutput](#)

Error Index

Error Index

Reads in a boolean.

[readBoolean\(\)](#)

Error Index

Reads an 8 bit byte.

[readByte\(\)](#)

Error Index

Reads a 16 bit char.

[readChar\(\)](#)

Error Index

Reads a 64 bit double.

[readDouble\(\)](#)

Error Index

Reads a 32 bit float.

[readFloat\(\)](#)

Error Index

Reads bytes, blocking until all bytes are read.

[readFully\(byte\[\]\)](#)

Error Index

Reads bytes, blocking until all bytes are read.

[readFully\(byte\[\], int, int\)](#)

Error Index

Reads a 32 bit int.

[readInt\(\)](#)

Error Index

[readLine\(\)](#)

Error Index

Reads a 64 bit long.

readLong()

Error Index

Reads a 16 bit short.

readShort()

Error Index

readUTF()

Error Index

readUnsignedByte()

Reads an unsigned 8 bit byte.

Error Index

readUnsignedShort()

Reads an unsigned 16 bit short.

Error Index

skipBytes(int)

Skips bytes, block until all bytes are skipped.

Error Index

Error Index

readFully

public abstract void readFully(byte b[]) throws IOException

Reads bytes, blocking until all bytes are read.

Parameters:

b - the buffer into which the data is read

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readFully

public abstract void readFully(byte b[],
int off,
int len) throws IOException

Reads bytes, blocking until all bytes are read.

Parameters:

b - the buffer into which the data is read

off - the start offset of the data

len - the maximum number of bytes to read

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index skipBytes

public abstract int skipBytes(int n) throws IOException

Skips bytes, block until all bytes are skipped.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index readBoolean

public abstract boolean readBoolean() throws IOException

Reads in a boolean.

Returns:

the boolean read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index readByte

public abstract byte readByte() throws IOException

Reads an 8 bit byte.

Returns:

the 8 bit byte read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readUnsignedByte

public abstract int readUnsignedByte() throws IOException

Reads an unsigned 8 bit byte.

Returns:

the 8 bit byte read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readShort

public abstract short readShort() throws IOException

Reads a 16 bit short.

Returns:

the 16 bit short read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readUnsignedShort

public abstract int readUnsignedShort() throws IOException

Reads an unsigned 16 bit short.

Returns:

the 16 bit short read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readChar

public abstract char readChar() throws IOException

Reads a 16 bit char.

Returns:

the 16 bit char read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readInt

public abstract int readInt() throws IOException

Reads a 32 bit int.

Returns:

the 32 bit integer read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readLong

public abstract long readLong() throws IOException

Reads a 64 bit long.

Returns:

the read 64 bit long.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readFloat

public abstract float readFloat() throws IOException

Reads a 32 bit float.

Returns:

the 32 bit float read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readDouble

public abstract double readDouble() throws IOException

Reads a 64 bit double.

Returns:

the 64 bit double read.

Throws: EOFException

If end of file is reached.

Throws: IOException

If other I/O error has occurred.

Error Index

readLine

public abstract String readLine() throws IOException

Error Index

readUTF

public abstract String readUTF() throws IOException

Class java.io.DataInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.DataInputStream

```
java.lang.Object
|
+----java.io.InputStream
      |
      +----java.io.FilterInputStream
            |
            +----java.io.DataInputStream
```

```
public class DataInputStream
    extends FilterInputStream
    implements DataInput
```

A data input stream that lets you read primitive Java data types from a stream in a portable way. Primitive data types are well understood types with associated operations. For example, Integers are considered primitive data types.

See Also:

[DataOutputStream](#)

Error Index

Error Index

DataInputStream(InputStream)

Creates a new DataInputStream.

Error Index

Error Index

read(byte[])

Reads data into an array of bytes.

Error Index

read(byte[], int, int)

Reads data into an array of bytes.

Error Index

Reads a boolean.

readBoolean()

Error Index

Reads an 8 bit byte.

readByte()

Error Index

Reads a 16 bit char.

readChar()

Error Index

Reads a 64 bit double.

readDouble()

Error Index

Reads a 32 bit float.

readFloat()

Error Index

Reads bytes, blocking until all bytes are read.

readFully(byte[])

Error Index

Reads bytes, blocking until all bytes are read.

readFully(byte[], int, int)

Error Index

Reads a 32 bit int.

readInt()

Error Index

Reads in a line that has been terminated by a \n, \r, \r\n or EOF.

readLine()

Error Index

Reads a 64 bit long.

readLong()

Error Index

Reads a 16 bit short.

readShort()

Error Index

Reads a UTF format String.

readUTF()

Error Index

Reads a UTF format String from the given input stream.

readUTF(DataInput)

Error Index

Reads an unsigned 8 bit byte.

readUnsignedByte()

Error Index

Reads 16 bit short.

readUnsignedShort()

Error Index

skipBytes(int)

Skips bytes, blocks until all bytes are skipped.

Error Index

Error Index

DataInputStream

public DataInputStream(InputStream in)

Creates a new DataInputStream.

Parameters:

in - the input stream

Error Index

Error Index

read

public final int read(byte b[]) throws IOException

Reads data into an array of bytes. This method blocks until some input is available.

Parameters:

b - the buffer into which the data is read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index

read

public final int read(byte b[],
int off,
int len) throws IOException

Reads data into an array of bytes. This method blocks until some input is available.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index readFully

public final void readFully(byte b[]) throws IOException

Reads bytes, blocking until all bytes are read.

Parameters:

b - the buffer into which the data is read

Throws: IOException

If an I/O error has occurred.

Throws: EOFException

If EOF reached before all bytes are read.

Error Index readFully

public final void readFully(byte b[],
int off,
int len) throws IOException

Reads bytes, blocking until all bytes are read.

Parameters:

b - the buffer into which the data is read

off - the start offset of the data

len - the maximum number of bytes read

Throws: IOException

If an I/O error has occurred.

Throws: EOFException

If EOF reached before all bytes are read.

Error Index skipBytes

public final int skipBytes(int n) throws IOException

Skips bytes, blocks until all bytes are skipped.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Throws: IOException

If an I/O error has occurred.

Error Index readBoolean

public final boolean readBoolean() throws IOException

Reads a boolean.

Returns:

the boolean read.

Error Index readByte

public final byte readByte() throws IOException

Reads an 8 bit byte.

Returns:

the 8 bit byte read.

Error Index readUnsignedByte

public final int readUnsignedByte() throws IOException

Reads an unsigned 8 bit byte.

Returns:

the 8 bit byte read.

Error Index readShort

public final short readShort() throws IOException

Reads a 16 bit short.

Returns:

the 16 bit short read.

Error Index

readUnsignedShort

public final int readUnsignedShort() throws IOException

Reads 16 bit short.

Returns:

the 16 bit short read.

Error Index

readChar

public final char readChar() throws IOException

Reads a 16 bit char.

Returns:

the read 16 bit char.

Error Index

readInt

public final int readInt() throws IOException

Reads a 32 bit int.

Returns:

the 32 bit integer read.

Error Index

readLong

public final long readLong() throws IOException

Reads a 64 bit long.

Returns:

the 64 bit long read.

Error Index

readFloat

public final float readFloat() throws IOException

Reads a 32 bit float.

Returns:

the read 32 bit float.

Error Index

readDouble

public final double readDouble() throws IOException

Reads a 64 bit double.

Returns:

the 64 bit double read.

Error Index

readLine

public final String readLine() throws IOException

Reads in a line that has been terminated by a \n, \r, \r\n or EOF.

Returns:

a String copy of the line.

Error Index

readUTF

public final String readUTF() throws IOException

Reads a UTF format String.

Returns:

the String.

Error Index

readUTF

public final static String readUTF(DataInput in) throws IOException

Reads a UTF format String from the given input stream.

Returns:

the String.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.io.DataOutput

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.io.DataOutput

public interface **DataOutput**
extends [Object](#)

DataOutput is an interface describing streams that can write output in a machine-independent format.

See Also:

[DataOutputStream](#), [DataInput](#)

Error Index

Error Index

Writes a byte.

[write](#)(int)

Error Index

Writes an array of bytes.

[write](#)(byte[])

Error Index

Writes a subarray of bytes.

[write](#)(byte[], int, int)

Error Index

Writes a boolean.

[writeBoolean](#)(boolean)

Error Index

Writes an 8 bit byte.

[writeByte](#)(int)

Error Index

Writes a String as a sequence of bytes.

[writeBytes](#)(String)

Error Index

Writes a 16 bit char.

[writeChar](#)(int)

Error Index

Writes a String as a sequence of chars.

[writeChars](#)(String)

Error Index

Writes a 64 bit double.

[writeDouble](#)(double)

Error Index

Writes a 32 bit float.

writeFloat(float)

Error Index

Writes a 32 bit int.

writeInt(int)

Error Index

Writes a 64 bit long.

writeLong(long)

Error Index

Writes a 16 bit short.

writeShort(int)

Error Index

Writes a String in UTF format.

writeUTF(String)

Error Index

Error Index

write

public abstract void write(int b) throws IOException

Writes a byte. Will block until the byte is actually written.

Parameters:

b - the byte to be written

Throws: IOException

If an I/O error has occurred.

Error Index

write

public abstract void write(byte b[]) throws IOException

Writes an array of bytes.

Parameters:

b - the data to be written

Throws: IOException

If an I/O error has occurred.

Error Index

write

```
public abstract void write(byte b[],  
    int off,  
    int len) throws IOException
```

Writes a subarray of bytes.

Parameters:

b - the data to be written
off - the start offset in the data
len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Error Index writeBoolean

```
public abstract void writeBoolean(boolean v) throws IOException
```

Writes a boolean.

Parameters:

v - the boolean to be written

Error Index writeByte

```
public abstract void writeByte(int v) throws IOException
```

Writes an 8 bit byte.

Parameters:

v - the byte value to be written

Error Index writeShort

```
public abstract void writeShort(int v) throws IOException
```

Writes a 16 bit short.

Parameters:

v - the short value to be written

Error Index writeChar

public abstract void writeChar(int v) throws IOException

Writes a 16 bit char.

Parameters:

v - the char value to be written

Error Index writeInt

public abstract void writeInt(int v) throws IOException

Writes a 32 bit int.

Parameters:

v - the integer value to be written

Error Index writeLong

public abstract void writeLong(long v) throws IOException

Writes a 64 bit long.

Parameters:

v - the long value to be written

Error Index writeFloat

public abstract void writeFloat(float v) throws IOException

Writes a 32 bit float.

Parameters:

v - the float value to be written

Error Index writeDouble

public abstract void writeDouble(double v) throws IOException

Writes a 64 bit double.

Parameters:

v - the double value to be written

Error Index writeBytes

public abstract void writeBytes(String s) throws IOException

Writes a String as a sequence of bytes.

Parameters:

s - the String of bytes to be written

Error Index writeChars

public abstract void writeChars(String s) throws IOException

Writes a String as a sequence of chars.

Parameters:

s - the String of chars to be written

Error Index writeUTF

public abstract void writeUTF(String str) throws IOException

Writes a String in UTF format.

Parameters:

str - the String in UTF format

Class java.io.DataOutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.DataOutputStream

```
java.lang.Object
|
+----java.io.OutputStream
      |
      +----java.io.FilterOutputStream
            |
            +----java.io.DataOutputStream
```

```
public class DataOutputStream
    extends FilterOutputStream
    implements DataOutput
```

This class lets you write primitive Java data types to a stream in a portable way. Primitive data types are well understood types with associated operations. For example, an Integer is considered to be a good primitive data type. The data can be converted back using a [DataInputStream](#).

Error Index

Error Index

written

The number of bytes written so far.

Error Index

Error Index

DataOutputStream(OutputStream)

Creates a new DataOutputStream.

Error Index

Error Index

Flushes the stream.

flush()

Error Index

Returns the number of bytes written.

size()

Error Index

Writes a byte.

write(int)

Error Index

Writes a sub array of bytes.

write(byte[], int, int)

Error Index

Writes a boolean.

writeBoolean(boolean)

Error Index

Writes an 8 bit byte.

writeByte(int)

Error Index

Writes a String as a sequence of bytes.

writeBytes(String)

Error Index

Writes a 16 bit char.

writeChar(int)

Error Index

Writes a String as a sequence of chars.

writeChars(String)

Error Index

Writes a 64 bit double.

writeDouble(double)

Error Index

Writes a 32 bit float.

writeFloat(float)

Error Index

Writes a 32 bit int.

writeInt(int)

Error Index

Writes a 64 bit long.

writeLong(long)

Error Index

Writes a 16 bit short.

writeShort(int)

Error Index

Writes a String in UTF format.

writeUTF(String)

Error Index

Error Index written

protected int written

The number of bytes written so far.

Error Index

Error Index DataOutputStream

public DataOutputStream(OutputStream out)

Creates a new DataOutputStream.

Parameters:

out - the output stream

Error Index

Error Index write

public synchronized void write(int b) throws IOException

Writes a byte. Will block until the byte is actually written.

Parameters:

b - the byte to be written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class FilterOutputStream

Error Index write


```
public synchronized void write(byte b[],
                               int off,
                               int len) throws IOException
```

Writes a sub array of bytes.

Parameters:

b - the data to be written
off - the start offset in the data
len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class FilterOutputStream

Error Index flush

```
public void flush() throws IOException
```

Flushes the stream. This will write any buffered output bytes.

Throws: IOException

If an I/O error has occurred.

Overrides:

flush in class FilterOutputStream

Error Index writeBoolean

```
public final void writeBoolean(boolean v) throws IOException
```

Writes a boolean.

Parameters:

v - the boolean to be written

Error Index writeByte

```
public final void writeByte(int v) throws IOException
```

Writes an 8 bit byte.

Parameters:

v - the byte value to be written

Error Index writeShort

public final void writeShort(int v) throws IOException

Writes a 16 bit short.

Parameters:

v - the short value to be written

Error Index writeChar

public final void writeChar(int v) throws IOException

Writes a 16 bit char.

Parameters:

v - the char value to be written

Error Index writeInt

public final void writeInt(int v) throws IOException

Writes a 32 bit int.

Parameters:

v - the integer value to be written

Error Index writeLong

public final void writeLong(long v) throws IOException

Writes a 64 bit long.

Parameters:

v - the long value to be written

Error Index writeFloat

public final void writeFloat(float v) throws IOException

Writes a 32 bit float.

Parameters:

v - the float value to be written

Error Index writeDouble

public final void writeDouble(double v) throws IOException

Writes a 64 bit double.

Parameters:

v - the double value to be written

Error Index writeBytes

public final void writeBytes(String s) throws IOException

Writes a String as a sequence of bytes.

Parameters:

s - the String of bytes to be written

Error Index writeChars

public final void writeChars(String s) throws IOException

Writes a String as a sequence of chars.

Parameters:

s - the String of chars to be written

Error Index writeUTF

public final void writeUTF(String str) throws IOException

Writes a String in UTF format.

Parameters:

str - the String in UTF format

Error Index size

public final int size()

Returns the number of bytes written.

Returns:

the number of bytes written thus far.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.EOFException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.EOFException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
                  |
                  +----java.io.EOFException
```

```
public class EOFException
    extends IOException
```

Signals that and EOF has been reached unexpectedly during input.

See Also:

[IOException](#), [DataInputStream](#)

Error Index

Error Index

EOFException()

Constructs an EOFException with no detail message.

Error Index

EOFException(String)

Constructs an EOFException with the specified detail message.

Error Index

Error Index

EOFException

```
public EOFException()
```

Constructs an EOFException with no detail message. A detail message is a String that describes this particular exception.

Error Index EOFException

```
public EOFException(String s)
```

Constructs an EOFException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.File

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.File

[java.lang.Object](#)
|
+----java.io.File

public class **File**
extends [Object](#)

This class represents a file name of the host file system. The file name can be relative or absolute. It must use the file name conventions of the host platform.

The intention is to provide an abstraction that deals with most of the system-dependent file name features such as the separator character, root, device name, etc. Not all features are currently fully implemented.

Note that whenever a file name or path is used it is assumed that the host's file name conventions are used.

Error Index

Error Index

[pathSeparator](#)

The system dependent path separator string.

Error Index

[pathSeparatorChar](#)

The system dependent path separator character.

Error Index

[separator](#)

The system dependent file separator String.

Error Index

[separatorChar](#)

The system dependent file separator character.

Error Index

Error Index

Creates a File object.

File(String)

Error Index

Creates a File object from the specified directory.

File(String, String)

Error Index

Creates a File object (given a directory File object).

File(File, String)

Error Index

Error Index

Returns a boolean indicating whether or not a readable file exists.

canRead()

Error Index

Returns a boolean indicating whether or not a writable file exists.

canWrite()

Error Index

Deletes the specified file.

delete()

Error Index

Compares this object against the specified object.

equals(Object)

Error Index

Returns a boolean indicating whether or not a file exists.

exists()

Error Index

Gets the absolute path of the file.

getAbsolutePath()

Error Index

Gets the name of the file.

getName()

Error Index

Gets the name of the parent directory.

getParent()

Error Index

Gets the path of the file.

getPath()

Error Index

Computes a hashcode for the file.

hashCode()

Error Index

isAbsolute()

Returns a boolean indicating whether the file name is absolute.

Error Index

isDirectory()

Returns a boolean indicating whether or not a directory file exists.

Error Index

isFile()

Returns a boolean indicating whether or not a normal file exists.

Error Index

lastModified()

Returns the last modification time.

Error Index

length()

Returns the length of the file.

Error Index

list()

Lists the files in a directory.

Error Index

list(FileNameFilter)

Uses the specified filter to list files in a directory.

Error Index

mkdir()

Creates a directory and returns a boolean indicating the success of the creation.

Error Index

mkdirs()

Creates all directories in this path.

Error Index

renameTo(File)

Renames a file and returns a boolean indicating whether or not this method was successful.

Error Index

toString()

Returns a String object representing this file's path.

Error Index

Error Index

separator

public final static String separator

The system dependent file separator String.

Error Index

separatorChar

public final static char separatorChar

The system dependent file separator character.

Error Index pathSeparator

public final static String pathSeparator

The system dependent path separator string.

Error Index pathSeparatorChar

public final static char pathSeparatorChar

The system dependent path separator character.

Error Index

Error Index File

public File(String path)

Creates a File object.

Parameters:

path - the file path

Throws: NullPointerException

If the file path is equal to null.

Error Index File

public File(String path,
String name)

Creates a File object from the specified directory.

Parameters:

path - the directory path

name - the file name

Error Index File

```
public File(File dir,  
           String name)
```

Creates a File object (given a directory File object).

Parameters:

dir - the directory
name - the file name

Error Index

Error Index getName

```
public String getName()
```

Gets the name of the file. This method does not include the directory.

Returns:

the file name.

Error Index getPath

```
public String getPath()
```

Gets the path of the file.

Returns:

the file path.

Error Index getAbsolutePath

```
public String getAbsolutePath()
```

Gets the absolute path of the file.

Returns:

the absolute file path.

Error Index getParent

```
public String getParent()
```

Gets the name of the parent directory.

Returns:
the parent directory, or null if one is not found.

Error Index exists

```
public boolean exists()
```

Returns a boolean indicating whether or not a file exists.

Error Index canWrite

```
public boolean canWrite()
```

Returns a boolean indicating whether or not a writable file exists.

Error Index canRead

```
public boolean canRead()
```

Returns a boolean indicating whether or not a readable file exists.

Error Index isFile

```
public boolean isFile()
```

Returns a boolean indicating whether or not a normal file exists.

Error Index isDirectory

```
public boolean isDirectory()
```

Returns a boolean indicating whether or not a directory file exists.

Error Index

isAbsolute

```
public boolean isAbsolute()
```

Returns a boolean indicating whether the file name is absolute.

Error Index

lastModified

```
public long lastModified()
```

Returns the last modification time. The return value should only be used to compare modification dates. It is meaningless as an absolute time.

Error Index

length

```
public long length()
```

Returns the length of the file.

Error Index

mkdir

```
public boolean mkdir()
```

Creates a directory and returns a boolean indicating the success of the creation.

Error Index

renameTo

```
public boolean renameTo(File dest)
```

Renames a file and returns a boolean indicating whether or not this method was successful.

Parameters:

dest - the new file name

Error Index

makedirs

```
public boolean mkdirs()
```

Creates all directories in this path. This method returns true if all directories in this path are created.

Error Index list

```
public String[] list()
```

Lists the files in a directory. Works only on directories.

Returns:

an array of file names. This list will include all files in the directory except the equivalent of "." and ".." .

Error Index list

```
public String[] list(FilenameFilter filter)
```

Uses the specified filter to list files in a directory.

Parameters:

filter - the filter used to select file names

Returns:

the filter selected files in this directory.

See Also:

FilenameFilter

Error Index delete

```
public boolean delete()
```

Deletes the specified file. Returns true if the file could be deleted.

Error Index hashCode

```
public int hashCode()
```

Computes a hashCode for the file.

Overrides:

hashCode in class Object

Error Index equals

public boolean equals(Object obj)

Compares this object against the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index toString

public String toString()

Returns a String object representing this file's path.

Overrides:

toString in class Object

Class java.io.FileDescriptor

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileDescriptor

[java.lang.Object](#)

|

+----[java.io.FileDescriptor](#)

public final class **FileDescriptor**
extends [Object](#)

Error Index

Error Index

Handle to standard error.

err

Error Index

Handle to standard input.

in

Error Index

Handle to standard output.

out

Error Index

Error Index

FileDescriptor()

Error Index

Error Index

valid()

Determines whether the file descriptor object is valid.

Error Index

Error Index_{in}

public final static FileDescriptor in

Handle to standard input.

Error Index_{out}

public final static FileDescriptor out

Handle to standard output.

Error Index_{err}

public final static FileDescriptor err

Handle to standard error.

Error Index

Error Index_{FileDescriptor}

public FileDescriptor()

Error Index

Error Index_{valid}

public boolean valid()

Determines whether the file descriptor object is valid.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileInputStream

```
java.lang.Object
|
+----java.io.InputStream
|
+----java.io.FileInputStream
```

```
public class FileInputStream
    extends InputStream
```

File input stream, can be constructed from a file descriptor or a file name.

See Also:

[FileOutputStream](#), [File](#)

Error Index

Error Index

[FileInputStream](#)(String)

Creates an input file with the specified system dependent file name.

Error Index

[FileInputStream](#)(File)

Creates an input file from the specified File object.

Error Index

[FileInputStream](#)(FileDescriptor)

Error Index

Error Index

[available](#)()

Returns the number of bytes that can be read without blocking.

Error Index

[close](#)()

Closes the input stream.

Error Index

finalize()

Closes the stream when garbage is collected.

Error Index

getFD()

Returns the opaque file descriptor object associated with this stream.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[])

Reads data into an array of bytes.

Error Index

read(byte[], int, int)

Reads data into an array of bytes.

Error Index

skip(long)

Skips n bytes of input.

Error Index

Error Index

FileInputStream

public FileInputStream(String name) throws FileNotFoundException

Creates an input file with the specified system dependent file name.

Parameters:

name - the system dependent file name

Throws: IOException

If the file is not found.

Error Index

FileInputStream

public FileInputStream(File file) throws FileNotFoundException

Creates an input file from the specified File object.

Parameters:

file - the file to be opened for reading

Throws: IOException

If the file is not found.

Error Index

FileInputStream

```
public FileInputStream(FileDescriptor fdObj)
```

Error Index

Error Index

read

```
public int read() throws IOException
```

Reads a byte of data. This method will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index

read

```
public int read(byte b[]) throws IOException
```

Reads data into an array of bytes. This method blocks until some input is available.

Parameters:

b - the buffer into which the data is read

Returns:

the actual number of bytes read. -1 is returned if the end of stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index

read

```
public int read(byte b[],  
    int off,  
    int len) throws IOException
```

Reads data into an array of bytes. This method blocks until some input is available.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read. -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index skip

public long skip(long n) throws IOException

Skips n bytes of input.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Throws: IOException

If an I/O error has occurred.

Overrides:

skip in class InputStream

Error Index available

public int available() throws IOException

Returns the number of bytes that can be read without blocking.

Returns:

the number of available bytes, which is initially equal to the file size.

Overrides:

available in class InputStream

Error Index close

public void close() throws IOException

Closes the input stream. This method must be called to release any resources associated with the stream.

Throws: [IOException](#)
If an I/O error has occurred.
Overrides:
[close](#) in class [InputStream](#)

Error Index **getFD**

public final [FileDescriptor](#) getFD() throws [IOException](#)

Returns the opaque file descriptor object associated with this stream.
Returns:
the file descriptor.

Error Index **finalize**

protected void finalize() throws [IOException](#)

Closes the stream when garbage is collected.
Overrides:
[finalize](#) in class [Object](#)

Interface java.io.FilenameFilter

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.io.FilenameFilter

public interface **FilenameFilter**
extends [Object](#)

A filter interface for file names.

See Also:

[File](#)

Error Index

Error Index

accept(File, String)

Determines whether a name should be included in a file list.

Error Index

Error Index

accept

public abstract boolean **accept**([File](#) dir,
[String](#) name)

Determines whether a name should be included in a file list.

Parameters:

dir - the directory in which the file was found
name - the name of the file

Returns:

true if name should be included in file list; false otherwise.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileNotFoundException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileNotFoundException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
                  |
                  +----java.io.FileNotFoundException
```

```
public class FileNotFoundException
    extends IOException
```

Signals that a file was not found.

Error Index

Error Index

[**FileNotFoundException\(\)**](#)

Constructs a FileNotFoundException with no detail message.

Error Index

[**FileNotFoundException\(String\)**](#)

Constructs a FileNotFoundException with the specified detail message.

Error Index

Error Index

FileNotFoundException

```
public FileNotFoundException()
```

Constructs a FileNotFoundException with no detail message. A detail message is a String that describes this particular exception.

Error Index

FileNotFoundException

public FileNotFoundException(String s)

Constructs a FileNotFoundException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileOutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FileOutputStream

[java.lang.Object](#)

```
|
+----java.io.OutputStream
      |
      +----java.io.FileOutputStream
```

```
public class FileOutputStream
    extends OutputStream
```

File output stream, can be constructed from a file descriptor or a file name.

See Also:

[FileInputStream](#), [File](#)

Error Index

Error Index

FileOutputStream(String)

Creates an output file with the specified system dependent file name.

Error Index

FileOutputStream(File)

Creates an output file with the specified File object.

Error Index

FileOutputStream(FileDescriptor)

Error Index

Error Index

close()

Closes the stream.

Error Index

finalize()

Closes the stream when garbage is collected.

Error Index

getFD()

Returns the file descriptor associated with this stream.

Error Index

write(int)

Writes a byte of data.

Error Index

write(byte[])

Writes an array of bytes.

Error Index

write(byte[], int, int)

Writes a sub array of bytes.

Error Index

Error Index

FileOutputStream

public FileOutputStream(String name) throws IOException

Creates an output file with the specified system dependent file name.

Parameters:

name - the system dependent file name

Throws: IOException

If the file is not found.

Error Index

FileOutputStream

public FileOutputStream(File file) throws IOException

Creates an output file with the specified File object.

Parameters:

file - the file to be opened for reading

Throws: IOException

If the file is not found.

Error Index

FileOutputStream

public FileOutputStream(FileDescriptor fdObj)

Error Index

Error Index write

public void write(int b) throws IOException

Writes a byte of data. This method will block until the byte is actually written.

Parameters:

b - the byte to be written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index write

public void write(byte b[]) throws IOException

Writes an array of bytes. Will block until the bytes are actually written.

Parameters:

b - the data to be written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index write

public void write(byte b[],
int off,
int len) throws IOException

Writes a sub array of bytes.

Parameters:

b - the data to be written

off - the start offset in the data

len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index close

public void close() throws [IOException](#)

Closes the stream. This method must be called to release any resources associated with the stream.

Throws: [IOException](#)

If an I/O error has occurred.

Overrides:

[close](#) in class [OutputStream](#)

Error Index getFD

public final [FileDescriptor](#) getFD() throws [IOException](#)

Returns the file descriptor associated with this stream.

Returns:

the file descriptor.

Error Index finalize

protected void finalize() throws [IOException](#)

Closes the stream when garbage is collected.

Overrides:

[finalize](#) in class [Object](#)

Class java.io.FilterInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FilterInputStream

[java.lang.Object](#)

```
|
+----java.io.InputStream
      |
      +----java.io.FilterInputStream
```

public class **FilterInputStream**
extends [InputStream](#)

Abstract class representing a filtered input stream of bytes. This class is the basis for enhancing input stream functionality. It allows multiple input stream filters to be chained together, each providing additional functionality.

Error Index

Error Index

in

The actual input stream.

Error Index

Error Index

FilterInputStream(InputStream)

Creates an input stream filter.

Error Index

Error Index

available()

Returns the number of bytes that can be read without blocking.

Error Index

close()

Closes the input stream.

Error Index

mark(int)

Marks the current position in the input stream.

Error Index

markSupported()

Returns true if this stream type supports mark/reset.

Error Index

read()

Reads a byte.

Error Index

read(byte[])

Reads into an array of bytes.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

reset()

Repositions the stream to the last marked position.

Error Index

skip(long)

Skips bytes of input.

Error Index

Error Index

protected InputStream in

The actual input stream.

Error Index

Error Index

protected FilterInputStream(InputStream in)

Creates an input stream filter.

Parameters:

in - the input stream

Error Index

Error Index read

public int read() throws IOException

Reads a byte. Will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index read

public int read(byte b[]) throws IOException

Reads into an array of bytes. Blocks until some input is available.

Parameters:

b - the buffer into which the data is read

Returns:

the actual number of bytes read. Returns -1 when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index read

public int read(byte b[],
int off,
int len) throws IOException

Reads into an array of bytes. Blocks until some input is available. This method should be overridden in a subclass for efficiency (the default implementation reads 1 byte at a time).

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read. Returns -1 when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index skip

public long skip(long n) throws IOException

Skips bytes of input.

Parameters:

n - bytes to be skipped

Returns:

actual number of bytes skipped

Throws: IOException

If an I/O error has occurred.

Overrides:

skip in class InputStream

Error Index available

public int available() throws IOException

Returns the number of bytes that can be read without blocking.

Returns:

the number of available bytes

Overrides:

available in class InputStream

Error Index close

public void close() throws IOException

Closes the input stream. Must be called to release any resources associated with the stream.

Throws: IOException

If an I/O error has occurred.

Overrides:

close in class InputStream

Error Index mark

public synchronized void mark(int readlimit)

Marks the current position in the input stream. A subsequent call to reset() will reposition the stream at the last marked position so that subsequent reads will re-read the same bytes. The stream promises to allow readlimit bytes to be read before the mark position gets invalidated.

Parameters:

readlimit - the maximum limit of bytes allowed to be read before the mark position becomes invalid.

Overrides:

mark in class InputStream

Error Index reset

public synchronized void reset() throws IOException

Repositions the stream to the last marked position. If the stream has not been marked, or if the mark has been invalidated, an IOException is thrown. Stream marks are intended to be used in situations where you need to read ahead a little to see what's in the stream. Often this is most easily done by invoking some general parser. If the stream is of the type handled by the parser, it just chugs along happily. If the stream is not of that type, the parser should toss an exception when it fails. If this happens within readlimit bytes, it allows the outer code to reset the stream and try another parser.

Overrides:

reset in class InputStream

Error Index markSupported

public boolean markSupported()

Returns true if this stream type supports mark/reset.

Overrides:

markSupported in class InputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FilterOutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.FilterOutputStream

```
java.lang.Object
|
+----java.io.OutputStream
      |
      +----java.io.FilterOutputStream
```

```
public class FilterOutputStream
    extends OutputStream
```

Abstract class representing a filtered output stream of bytes. This class is the basis for enhancing output stream functionality. It allows multiple output stream filters to be chained together, each providing additional functionality.

Error Index

Error Index

The actual output stream. [out](#)

Error Index

Error Index

[FilterOutputStream](#)(OutputStream)
Creates an output stream filter.

Error Index

Error Index

[close\(\)](#)
Closes the stream.

Error Index

Flushes the stream.

flush()

Error Index

Writes a byte.

write(int)

Error Index

Writes an array of bytes.

write(byte[])

Error Index

Writes a subarray of bytes.

write(byte[], int, int)

Error Index

Error Index **out**

protected OutputStream out

The actual output stream.

Error Index

Error Index **FilterOutputStream**

public FilterOutputStream(OutputStream out)

Creates an output stream filter.

Parameters:

out - the output stream

Error Index

Error Index **write**

public void write(int b) throws IOException

Writes a byte. Will block until the byte is actually written.

Parameters:

b - the byte

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index write

public void write(byte b[]) throws IOException

Writes an array of bytes. Will block until the bytes are actually written.

Parameters:

b - the data to be written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index write

public void write(byte b[],
int off,
int len) throws IOException

Writes a subarray of bytes. To be efficient it should be overridden in a subclass.

Parameters:

b - the data to be written

off - the start offset in the data

len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index flush

public void flush() throws IOException

Flushes the stream. This will write any buffered output bytes.

Throws: [IOException](#)

If an I/O error has occurred.

Overrides:

[flush](#) in class [OutputStream](#)

Error Index [close](#)

public void close() throws [IOException](#)

Closes the stream. This method must be called to release any resources associated with the stream.

Throws: [IOException](#)

If an I/O error has occurred.

Overrides:

[close](#) in class [OutputStream](#)

Class java.io.InputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.InputStream

[java.lang.Object](#)

|

+----java.io.InputStream

```
public class InputStream
    extends Object
```

An abstract class representing an input stream of bytes. All InputStreams are based on this class.

See Also:

[OutputStream](#), [FilterInputStream](#), [BufferedInputStream](#), [DataInputStream](#) , [ByteArrayInputStream](#),
[PushbackInputStream](#)

Error Index

Error Index

[InputStream\(\)](#)

Error Index

Error Index

[available\(\)](#)

Returns the number of bytes that can be read without blocking.

Error Index

[close\(\)](#)

Closes the input stream.

Error Index

[mark\(int\)](#)

Marks the current position in the input stream.

Error Index

[markSupported\(\)](#)

Returns a boolean indicating whether or not this stream type supports mark/reset.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[])

Reads into an array of bytes.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

reset()

Repositions the stream to the last marked position.

Error Index

skip(long)

Skips n bytes of input.

Error Index

Error Index

InputStream

public InputStream()

Error Index

Error Index

read

public abstract int read() throws IOException

Reads a byte of data. This method will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Error Index

read

public int read(byte b[]) throws IOException

Reads into an array of bytes. This method will block until some input is available.

Parameters:

b - the buffer into which the data is read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Error Index read

```
public int read(byte b[],  
    int off,  
    int len) throws IOException
```

Reads into an array of bytes. This method will block until some input is available.

Parameters:

b - the buffer into which the data is read

off - the start offset of the data

len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Error Index skip

```
public long skip(long n) throws IOException
```

Skips n bytes of input.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Throws: IOException

If an I/O error has occurred.

Error Index available

```
public int available() throws IOException
```

Returns the number of bytes that can be read without blocking.

Returns:

the number of available bytes.

Error Index close

public void close() throws IOException

Closes the input stream. Must be called to release any resources associated with the stream.

Throws: IOException

If an I/O error has occurred.

Error Index mark

public synchronized void mark(int readlimit)

Marks the current position in the input stream. A subsequent call to reset() will reposition the stream at the last marked position so that subsequent reads will re-read the same bytes. The stream promises to allow readlimit bytes to be read before the mark position gets invalidated.

Parameters:

readlimit - the maximum limit of bytes allowed to be read before the mark position becomes invalid.

Error Index reset

public synchronized void reset() throws IOException

Repositions the stream to the last marked position. If the stream has not been marked, or if the mark has been invalidated, an IOException is thrown. Stream marks are intended to be used in situations where you need to read ahead a little to see what's in the stream. Often this is most easily done by invoking some general parser. If the stream is of the type handled by the parser, it just chugs along happily. If the stream is not of that type, the parser should toss an exception when it fails, which, if it happens within readlimit bytes, allows the outer code to reset the stream and try another parser.

Throws: IOException

If the stream has not been marked or if the mark has been invalidated.

Error Index markSupported

public boolean markSupported()

Returns a boolean indicating whether or not this stream type supports mark/reset.

Returns:

true if this stream type supports mark/reset; false otherwise.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.InterruptedIOException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.InterruptedIOException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
                  |
                  +----java.io.InterruptedIOException
```

```
public class InterruptedIOException
    extends IOException
```

Signals that an I/O operation has been interrupted.

See Also:

[InputStream](#), [OutputStream](#)

Error Index

Error Index

bytesTransferred

Reports how many bytes had been transferred as part of the IO operation before it was interrupted.

Error Index

Error Index

InterruptedIOException()

Constructs an IOException with no detail message.

Error Index

InterruptedIOException(String)

Constructs an IOException with the specified detail message.

Error Index

Error Index

bytesTransferred

```
public int bytesTransferred
```

Reports how many bytes had been transferred as part of the IO operation before it was interrupted.

Error Index

Error Index

InterruptedException

```
public InterruptedException()
```

Constructs an IOException with no detail message. A detail message is a String that describes this particular exception.

Error Index

InterruptedException

```
public InterruptedException(String s)
```

Constructs an IOException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

Class java.io.IOException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.IOException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
```

```
public class IOException
extends Exception
```

Signals that an I/O exception has occurred.

See Also:

[InputStream](#), [OutputStream](#)

Error Index

Error Index

[IOException\(\)](#)

Constructs an IOException with no detail message.

Error Index

[IOException\(String\)](#)

Constructs an IOException with the specified detail message.

Error Index

Error Index

[IOException](#)

```
public IOException()
```

Constructs an IOException with no detail message. A detail message is a String that describes this

particular exception.

Error Index IOException

public IOException(String s)

Constructs an IOException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.LineNumberInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.LineNumberInputStream

```
java.lang.Object
|
+----java.io.InputStream
      |
      +----java.io.FilterInputStream
            |
            +----java.io.LineNumberInputStream
```

```
public class LineNumberInputStream
    extends FilterInputStream
```

An input stream that keeps track of line numbers.

Error Index

Error Index

[**LineNumberInputStream**](#)(InputStream)

Constructs a new LineNumberInputStream initialized with the specified input stream.

Error Index

Error Index

[**available**](#)()

Returns the number of bytes that can be read without blocking.

Error Index

[**getLineNumber**](#)()

Returns the current line number.

Error Index

[**mark**](#)(int)

Marks the current position in the input stream.

Error Index

[**read**](#)()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

reset()

Repositions the stream to the last marked position.

Error Index

setLineNumber(int)

Sets the current line number.

Error Index

skip(long)

Skips n bytes of input.

Error Index

Error Index

LineNumberInputStream

public LineNumberInputStream(InputStream in)

Constructs a new LineNumberInputStream initialized with the specified input stream.

Parameters:

in - the input stream

Error Index

Error Index

read

public int read() throws IOException

Reads a byte of data. The method will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index **read**

```
public int read(byte b[],  
               int off,  
               int len) throws IOException
```

Reads into an array of bytes. This method will blocks until some input is available.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index **setLineNumber**

```
public void setLineNumber(int lineNumber)
```

Sets the current line number.

Parameters:

lineNumber - the line number to be set

Error Index **getLineNumber**

```
public int getLineNumber()
```

Returns the current line number.

Error Index **skip**

```
public long skip(long n) throws IOException
```

Skips n bytes of input.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Throws: IOException

If an I/O error has occurred.

Overrides:

skip in class FilterInputStream

Error Index available

public int available() throws IOException

Returns the number of bytes that can be read without blocking.

Returns:

the number of available bytes

Overrides:

available in class FilterInputStream

Error Index mark

public void mark(int readlimit)

Marks the current position in the input stream. A subsequent call to `reset()` will reposition the stream at the last marked position so that subsequent reads will re-read the same bytes. The stream promises to allow `readlimit` bytes to be read before the mark position gets invalidated.

Parameters:

`readlimit` - the maximum limit of bytes allowed to be read before the mark position becomes invalid.

Overrides:

mark in class FilterInputStream

Error Index reset

public void reset() throws IOException

Repositions the stream to the last marked position. If the stream has not been marked, or if the mark has been invalidated, an `IOException` is thrown. Stream marks are intended to be used in situations where you need to read ahead a little to see what's in the stream. Often this is most easily done by invoking some general parser. If the stream is of the type handled by the parser, it just chugs along happily. If the stream is not of that type, the parser should toss an exception when it fails, which, if it happens within `readlimit` bytes, allows the outer code to reset the stream and try another parser.

Overrides:

reset in class FilterInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.OutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.OutputStream

[java.lang.Object](#)

|

+----java.io.OutputStream

public class **OutputStream**

extends [Object](#)

Abstract class representing an output stream of bytes. All OutputStreams are based on this class.

See Also:

[InputStream](#), [FilterOutputStream](#), [BufferedOutputStream](#), [DataOutputStream](#),
[ByteArrayOutputStream](#)

Error Index

Error Index

OutputStream()

Error Index

Error Index

close()

Closes the stream.

Error Index

flush()

Flushes the stream.

Error Index

write(int)

Writes a byte.

Error Index

write(byte[])

Writes an array of bytes.

Error Index

write(byte[], int, int)

Writes a sub array of bytes.

Error Index

Error Index

OutputStream

public OutputStream()

Error Index

Error Index

write

public abstract void write(int b) throws IOException

Writes a byte. This method will block until the byte is actually written.

Parameters:

b - the byte

Throws: IOException

If an I/O error has occurred.

Error Index

write

public void write(byte b[]) throws IOException

Writes an array of bytes. This method will block until the bytes are actually written.

Parameters:

b - the data to be written

Throws: IOException

If an I/O error has occurred.

Error Index

write

public void write(byte b[],
int off,
int len) throws IOException

Writes a sub array of bytes.

Parameters:

b - the data to be written

off - the start offset in the data

len - the number of bytes that are written

Throws: [IOException](#)

If an I/O error has occurred.

Error Index flush

public void flush() throws [IOException](#)

Flushes the stream. This will write any buffered output bytes.

Throws: [IOException](#)

If an I/O error has occurred.

Error Index close

public void close() throws [IOException](#)

Closes the stream. This method must be called to release any resources associated with the stream.

Throws: [IOException](#)

If an I/O error has occurred.

Class java.io.PipedInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.PipedInputStream

[java.lang.Object](#)

```
|
+----java.io.InputStream
      |
      +----java.io.PipedInputStream
```

public class **PipedInputStream**
extends [InputStream](#)

PipedInputStream must be connected to a PipedOutputStream to be useful. A thread reading from a PipedInputStream receives data from a thread writing to the PipedOutputStream it is connected to.

See Also:

[PipedOutputStream](#)

Error Index

Error Index

[PipedInputStream\(PipedOutputStream\)](#)

Creates an input file from the specified PipedOutputStream.

Error Index

[PipedInputStream\(\)](#)

Creates an input file that isn't connected to anything (yet).

Error Index

Error Index

[close\(\)](#)

Closes the input stream.

Error Index

[connect\(PipedOutputStream\)](#)

Connects this input stream to a sender.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

Error Index

PipedInputStream

public PipedInputStream(PipedOutputStream src) throws IOException

Creates an input file from the specified PipedOutputStream.

Parameters:

src - the stream to connect to.

Error Index

PipedInputStream

public PipedInputStream()

Creates an input file that isn't connected to anything (yet). It must be connected to a PipedOutputStream before being used.

Error Index

Error Index

connect

public void connect(PipedOutputStream src) throws IOException

Connects this input stream to a sender.

Parameters:

src - The OutputStream to connect to.

Error Index

read

public synchronized int read() throws IOException

Reads a byte of data. This method will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If the pipe is broken.

Overrides:

read in class InputStream

Error Index read

public synchronized int read(byte b[],
int off,
int len) throws IOException

Reads into an array of bytes. Blocks until some input is available.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index close

public void close() throws IOException

Closes the input stream. Must be called to release any resources associated with the stream.

Throws: IOException

If an I/O error has occurred.

Overrides:

close in class InputStream

Class java.io.PipedOutputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.PipedOutputStream

[java.lang.Object](#)

```
|
+----java.io.OutputStream
      |
      +----java.io.PipedOutputStream
```

public class **PipedOutputStream**
extends [OutputStream](#)

Piped output stream, must be connected to a PipedInputStream. A thread reading from a PipedInputStream receives data from a thread writing to the PipedOutputStream it is connected to.

See Also:

[PipedInputStream](#)

Error Index

Error Index

[PipedOutputStream](#)(PipedInputStream)

Creates an output file connected to the specified PipedInputStream.

Error Index

[PipedOutputStream](#)()

Creates an output file that isn't connected to anything (yet).

Error Index

Error Index

[close](#)()

Closes the stream.

Error Index

[connect](#)(PipedInputStream)

Connect this output stream to a receiver.

Error Index

write(int)

Write a byte.

Error Index

write(byte[], int, int)

Writes a sub array of bytes.

Error Index

Error Index

PipedOutputStream

public PipedOutputStream(PipedInputStream snk) throws IOException

Creates an output file connected to the specified PipedInputStream.

Parameters:

snk - The InputStream to connect to.

Error Index

PipedOutputStream

public PipedOutputStream()

Creates an output file that isn't connected to anything (yet). It must be connected before being used.

Error Index

Error Index

connect

public void connect(PipedInputStream snk) throws IOException

Connect this output stream to a receiver.

Parameters:

snk - The InputStream to connect to.

Error Index

write

public void write(int b) throws IOException

Write a byte. This method will block until the byte is actually written.

Parameters:

b - the byte to be written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index **write**

```
public void write(byte b[],  
    int off,  
    int len) throws IOException
```

Writes a sub array of bytes.

Parameters:

b - the data to be written

off - the start offset in the data

len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class OutputStream

Error Index **close**

```
public void close() throws IOException
```

Closes the stream. This method must be called to release any resources associated with the stream.

Throws: IOException

If an I/O error has occurred.

Overrides:

close in class OutputStream

Class java.io.PrintStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.PrintStream

```
java.lang.Object
|
+----java.io.OutputStream
      |
      +----java.io.FilterOutputStream
            |
            +----java.io.PrintStream
```

```
public class PrintStream
    extends FilterOutputStream
```

This class implements an output stream that has additional methods for printing. You can specify that the stream should be flushed every time a newline character is written.

The top byte of 16 bit characters is discarded.

Example:

```
System.out.println("Hello world!");
System.out.print("x = ");
System.out.println(x);
System.out.println("y = " + y);
```

Error Index

Error Index

PrintStream(OutputStream)

Creates a new PrintStream.

Error Index

PrintStream(OutputStream, boolean)

Creates a new PrintStream, with auto flushing.

Error Index

Error Index

checkError()

Flushes the print stream and returns whether or not there was an error on the output stream.

Error Index

close()

Closes the stream.

Error Index

flush()

Flushes the stream.

Error Index

print(Object)

Prints an object.

Error Index

print(String)

Prints a String.

Error Index

print(char[])

Prints an array of characters.

Error Index

print(char)

Prints an character.

Error Index

print(int)

Prints an integer.

Error Index

print(long)

Prints a long.

Error Index

print(float)

Prints a float.

Error Index

print(double)

Prints a double.

Error Index

print(boolean)

Prints a boolean.

Error Index

println()

Prints a newline.

Error Index

println(Object)

Prints an object followed by a newline.

Error Index

println(String)

Prints a string followed by a newline.

Error Index

println(char[])

Prints an array of characters followed by a newline.

Error Index

println(char)

Prints a character followed by a newline.

Error Index

println(int)

Prints an integer followed by a newline.

Error Index

println(long)

Prints a long followed by a newline.

Error Index

println(float)

Prints a float followed by a newline.

Error Index

println(double)

Prints a double followed by a newline.

Error Index

println(boolean)

Prints a boolean followed by a newline.

Error Index

write(int)

Writes a byte.

Error Index

write(byte[], int, int)

Writes a sub array of bytes.

Error Index

Error Index

PrintStream

public PrintStream(OutputStream out)

Creates a new PrintStream.

Parameters:

out - the output stream

Error Index

PrintStream

public PrintStream(OutputStream out,

boolean autoflush)

Creates a new PrintStream, with auto flushing.

Parameters:

out - the output stream

autoflush - if true the stream automatically flushes
its output when a newline character is
printed

Error Index

Error Index write

public void write(int b)

Writes a byte. This method will block until the byte is actually written.

Parameters:

b - the byte

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class FilterOutputStream

Error Index write

public void write(byte b[],
int off,
int len)

Writes a sub array of bytes.

Parameters:

b - the data to be written

off - the start offset in the data

len - the number of bytes that are written

Throws: IOException

If an I/O error has occurred.

Overrides:

write in class FilterOutputStream

Error Index flush

public void flush()

Flushes the stream. This will write any buffered output bytes.

Overrides:

flush in class FilterOutputStream

Error Index close

public void close()

Closes the stream.

Overrides:

close in class FilterOutputStream

Error Index checkError

public boolean checkError()

Flushes the print stream and returns whether or not there was an error on the output stream. Errors are cumulative; once the print stream encounters an error this routine will continue to return true on all successive calls.

Returns:

true if the print stream has ever encountered an error on the output stream.

Error Index print

public void print(Object obj)

Prints an object.

Parameters:

obj - the object to be printed

Error Index print

public synchronized void print(String s)

Prints a String.

Parameters:

s - the String to be printed

Error Index print

```
public synchronized void print(char s[])
```

Prints an array of characters.

Parameters:

s - the array of chars to be printed

Error Index print

```
public void print(char c)
```

Prints an character.

Parameters:

c - the character to be printed

Error Index print

```
public void print(int i)
```

Prints an integer.

Parameters:

i - the integer to be printed

Error Index print

```
public void print(long l)
```

Prints a long.

Parameters:

l - the long to be printed.

Error Index print

```
public void print(float f)
```

Prints a float.

Parameters:

f - the float to be printed

Error Index print

```
public void print(double d)
```

Prints a double.

Parameters:

d - the double to be printed

Error Index print

```
public void print(boolean b)
```

Prints a boolean.

Parameters:

b - the boolean to be printed

Error Index println

```
public void println()
```

Prints a newline.

Error Index println

```
public synchronized void println(Object obj)
```

Prints an object followed by a newline.

Parameters:

obj - the object to be printed

Error Index println

public synchronized void println(String s)

Prints a string followed by a newline.

Parameters:

s - the String to be printed

Error Index println

public synchronized void println(char s[])

Prints an array of characters followed by a newline.

Parameters:

s - the array of characters to be printed

Error Index println

public synchronized void println(char c)

Prints a character followed by a newline.

Parameters:

c - the character to be printed

Error Index println

public synchronized void println(int i)

Prints an integer followed by a newline.

Parameters:

i - the integer to be printed

Error Indexprintln

public synchronized void println(long l)

Prints a long followed by a newline.

Parameters:

l - the long to be printed

Error Indexprintln

public synchronized void println(float f)

Prints a float followed by a newline.

Parameters:

f - the float to be printed

Error Indexprintln

public synchronized void println(double d)

Prints a double followed by a newline.

Parameters:

d - the double to be printed

Error Indexprintln

public synchronized void println(boolean b)

Prints a boolean followed by a newline.

Parameters:

b - the boolean to be printed

Class java.io.PushbackInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.PushbackInputStream

```
java.lang.Object
|
+----java.io.InputStream
      |
      +----java.io.FilterInputStream
            |
            +----java.io.PushbackInputStream
```

```
public class PushbackInputStream
    extends FilterInputStream
```

An input stream that has a 1 byte push back buffer.

Error Index

Error Index

pushBack

Push back character.

Error Index

Error Index

PushbackInputStream(InputStream)

Creates a PushbackInputStream.

Error Index

Error Index

available()

Returns the number of bytes that can be read.

Error Index

markSupported()

Returns true if this stream type supports mark/reset.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

unread(int)

Pushes back a character.

Error Index

Error Index

pushBack

protected int pushBack

Push back character.

Error Index

Error Index

PushbackInputStream

public PushbackInputStream(InputStream in)

Creates a PushbackInputStream.

Parameters:

in - the input stream

Error Index

Error Index

read

public int read() throws IOException

Reads a byte of data. This method will block if no input is available.

Returns:

the byte read, or -1 if the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index read

```
public int read(byte bytes[],  
               int offset,  
               int length) throws IOException
```

Reads into an array of bytes. This method blocks until some input is available.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class FilterInputStream

Error Index unread

```
public void unread(int ch) throws IOException
```

Pushes back a character.

Parameters:

ch - the character to push back.

Throws: IOException

If an attempt to push back more than one character is made.

Error Index available

```
public int available() throws IOException
```

Returns the number of bytes that can be read. without blocking.

Overrides:

available in class FilterInputStream

Error Index **markSupported**

public boolean markSupported()

Returns true if this stream type supports mark/reset.

Overrides:

markSupported in class FilterInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.RandomAccessFile

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.RandomAccessFile

[java.lang.Object](#)

|

+----java.io.RandomAccessFile

```
public class RandomAccessFile
    extends Object
    implements DataOutput, DataInput
```

Random access files can be constructed from file descriptors, file names, or file objects. This class provides a sense of security by offering methods that allow specified mode accesses of read-only or read-write to files.

Error Index

Error Index

[RandomAccessFile](#)(String, String)

Creates a RandomAccessFile with the specified system dependent file name and the specified mode.

Error Index

[RandomAccessFile](#)(File, String)

Creates a RandomAccessFile from a specified File object and mode ("r" or "rw").

Error Index

Error Index

[close](#)()

Closes the file.

Error Index

[getFD](#)()

Returns the opaque file descriptor object.

Error Index

[getFilePointer](#)()

Returns the current location of the file pointer.

Error Index

length()

Returns the length of the file.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads a sub array as a sequence of bytes.

Error Index

read(byte[])

Reads data into an array of bytes.

Error Index

readBoolean()

Reads a boolean.

Error Index

readByte()

Reads a byte.

Error Index

readChar()

Reads a 16 bit char.

Error Index

readDouble()

Reads a 64 bit double.

Error Index

readFloat()

Reads a 32 bit float.

Error Index

readFully(byte[])

Reads bytes, blocking until all bytes are read.

Error Index

readFully(byte[], int, int)

Reads bytes, blocking until all bytes are read.

Error Index

readInt()

Reads a 32 bit int.

Error Index

readLine()

Reads a line terminated by a '\n' or EOF.

Error Index

readLong()

Reads a 64 bit long.

Error Index

readShort()

Reads 16 bit short.

Error Index

readUTF()

Reads a UTF formatted String.

Error Index

readUnsignedByte()

Reads an unsigned 8 bit byte.

Error Index

readUnsignedShort()

Reads 16 bit short.

Error Index

seek(long)

Sets the file pointer to the specified absolute position.

Error Index

skipBytes(int)

Error Index

write(int)

Writes a byte of data.

Error Index

write(byte[])

Writes an array of bytes.

Error Index

write(byte[], int, int)

Writes a sub array of bytes.

Error Index

writeBoolean(boolean)

Writes a boolean.

Error Index

writeByte(int)

Writes a byte.

Error Index

writeBytes(String)

Writes a String as a sequence of bytes.

Error Index

writeChar(int)

Writes a character.

Error Index

writeChars(String)

Writes a String as a sequence of chars.

Error Index

writeDouble(double)

Error Index

writeFloat(float)

Error Index

writeInt(int)

Writes an integer.

Error Index

Writes a long.

writeLong(long)

Error Index

Writes a short.

writeShort(int)

Error Index

Writes a String in UTF format.

writeUTF(String)

Error Index

Error Index

RandomAccessFile

```
public RandomAccessFile(String name,  
                        String mode) throws IOException
```

Creates a RandomAccessFile with the specified system dependent file name and the specified mode. Mode "r" is for read-only and mode "rw" is for read+write.

Parameters:

name - the system dependent file name

mode - the access mode

Throws: IOException

If an I/O error has occurred.

Error Index

RandomAccessFile

```
public RandomAccessFile(File file,  
                        String mode) throws IOException
```

Creates a RandomAccessFile from a specified File object and mode ("r" or "rw").

Parameters:

file - the file object

mode - the access mode

Error Index

Error Index

getFD

public final FileDescriptor getFD() throws IOException

Returns the opaque file descriptor object.

Returns:
the file descriptor.

Error Index read

public int read() throws IOException

Reads a byte of data. This method will block if no input is available.

Returns:
the byte read, or -1 if the end of the stream is reached.

Throws: IOException
If an I/O error has occurred.

Error Index read

public int read(byte b[],
int off,
int len) throws IOException

Reads a sub array as a sequence of bytes.

Parameters:
b - the data to be written
off - the start offset in the data
len - the number of bytes that are written

Throws: IOException
If an I/O error has occurred.

Error Index read

public int read(byte b[]) throws IOException

Reads data into an array of bytes. This method blocks until some input is available.

Returns:
the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException
If an I/O error has occurred.

Error Index

readFully

public final void readFully(byte b[]) throws IOException

Reads bytes, blocking until all bytes are read.

Parameters:

b - the buffer into which the data is read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Error Index

readFully

public final void readFully(byte b[],
int off,
int len) throws IOException

Reads bytes, blocking until all bytes are read.

Parameters:

b - the buffer into which the data is read

off - the start offset of the data

len - the maximum number of bytes read

Returns:

the actual number of bytes read, -1 is returned when the end of the stream is reached.

Throws: IOException

If an I/O error has occurred.

Error Index

skipBytes

public int skipBytes(int n) throws IOException

Error Index

write

public void write(int b) throws IOException

Writes a byte of data. This method will block until the byte is actually written.

Parameters:

b - the byte to be written

Throws: IOException
If an I/O error has occurred.

Error Index write

public void write(byte b[]) throws IOException

Writes an array of bytes. Will block until the bytes are actually written.

Parameters:

b - the data to be written

Throws: IOException
If an I/O error has occurred.

Error Index write

public void write(byte b[],
int off,
int len) throws IOException

Writes a sub array of bytes.

Parameters:

b - the data to be written

off - the start offset in the data

len - the number of bytes that are written

Throws: IOException
If an I/O error has occurred.

Error Index getFilePointer

public long getFilePointer() throws IOException

Returns the current location of the file pointer.

Error Index seek

public void seek(long pos) throws IOException

Sets the file pointer to the specified absolute position.

Parameters:

pos - the absolute position

Error Indexlength

public long length() throws IOException

Returns the length of the file.

Error Indexclose

public void close() throws IOException

Closes the file.

Throws: IOException
If an I/O error has occurred.

Error IndexreadBoolean

public final boolean readBoolean() throws IOException

Reads a boolean.

Error IndexreadByte

public final byte readByte() throws IOException

Reads a byte.

Error IndexreadUnsignedByte

public final int readUnsignedByte() throws IOException

Reads an unsigned 8 bit byte.

Returns:
the 8 bit byte read.

Error Index

readShort

public final short readShort() throws IOException

Reads 16 bit short.

Returns:

the read 16 bit short.

Error Index

readUnsignedShort

public final int readUnsignedShort() throws IOException

Reads 16 bit short.

Returns:

the read 16 bit short.

Error Index

readChar

public final char readChar() throws IOException

Reads a 16 bit char.

Returns:

the read 16 bit char.

Error Index

readInt

public final int readInt() throws IOException

Reads a 32 bit int.

Returns:

the read 32 bit integer.

Error Index

readLong

public final long readLong() throws IOException

Reads a 64 bit long.

Returns:

the read 64 bit long.

Error Index readFloat

public final float readFloat() throws IOException

Reads a 32 bit float.

Returns:

the read 32 bit float.

Error Index readDouble

public final double readDouble() throws IOException

Reads a 64 bit double.

Returns:

the read 64 bit double.

Error Index readLine

public final String readLine() throws IOException

Reads a line terminated by a '\n' or EOF.

Error Index readUTF

public final String readUTF() throws IOException

Reads a UTF formatted String.

Error Index writeBoolean

public final void writeBoolean(boolean v) throws IOException

Writes a boolean.

Parameters:

v - the boolean value

Error Index writeByte

public final void writeByte(int v) throws IOException

Writes a byte.

Parameters:

v - the byte

Error Index writeShort

public final void writeShort(int v) throws IOException

Writes a short.

Parameters:

v - the short

Error Index writeChar

public final void writeChar(int v) throws IOException

Writes a character.

Parameters:

v - the char

Error Index writeInt

public final void writeInt(int v) throws IOException

Writes an integer.

Parameters:

v - the integer

Error Index writeLong

public final void writeLong(long v) throws IOException

Writes a long.

Parameters:

v - the long

Error Index writeFloat

public final void writeFloat(float v) throws IOException

Error Index writeDouble

public final void writeDouble(double v) throws IOException

Error Index writeBytes

public final void writeBytes(String s) throws IOException

Writes a String as a sequence of bytes.

Parameters:

s - the String

Error Index writeChars

public final void writeChars(String s) throws IOException

Writes a String as a sequence of chars.

Parameters:

s - the String

Error Index writeUTF

public final void writeUTF(String str) throws IOException

Writes a String in UTF format.

Parameters:

str - the String

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.SequenceInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.SequenceInputStream

```
java.lang.Object
|
+----java.io.InputStream
      |
      +----java.io.SequenceInputStream
```

```
public class SequenceInputStream
    extends InputStream
```

Converts a sequence of input streams into an InputStream.

Error Index

Error Index

[SequenceInputStream](#)(Enumeration)

Constructs a new SequenceInputStream initialized to the specified list.

Error Index

[SequenceInputStream](#)(InputStream, InputStream)

Constructs a new SequenceInputStream initialized to the two specified input streams.

Error Index

Error Index

[close\(\)](#)

Closes the input stream; flipping to the next stream, if an EOF is reached.

Error Index

[read\(\)](#)

Reads a stream, and upon reaching an EOF, flips to the next stream.

Error Index

[read](#)(byte[], int, int)

Reads data into an array of bytes, and upon reaching an EOF, flips to the next stream.

Error Index

Error Index

SequenceInputStream

```
public SequenceInputStream(Enumeration e)
```

Constructs a new SequenceInputStream initialized to the specified list.

Parameters:

e - the list

Error Index

SequenceInputStream

```
public SequenceInputStream(InputStream s1,  
                           InputStream s2)
```

Constructs a new SequenceInputStream initialized to the two specified input streams.

Parameters:

s1 - the first input stream

s2 - the second input stream

Error Index

Error Index

read

```
public int read() throws IOException
```

Reads a stream, and upon reaching an EOF, flips to the next stream.

Overrides:

read in class InputStream

Error Index

read

```
public int read(byte buf[],  
                int pos,  
                int len) throws IOException
```

Reads data into an array of bytes, and upon reaching an EOF, flips to the next stream.

Parameters:

buf - the buffer into which the data is read

pos - the start position of the data

len - the maximum number of bytes read

Throws: IOException

If an I/O error has occurred.

Overrides:

read in class InputStream

Error Index close

public void close() throws IOException

Closes the input stream; flipping to the next stream, if an EOF is reached. This method must be called to release any resources associated with the stream.

Throws: IOException

If an I/O error has occurred.

Overrides:

close in class InputStream

Class java.io.StreamTokenizer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.StreamTokenizer

[java.lang.Object](#)

|
+----java.io.StreamTokenizer

```
public class StreamTokenizer  
extends Object
```

A class to turn an input stream into a stream of tokens. There are a number of methods that define the lexical syntax of tokens.

Error Index

Error Index

The End-of-file token.

TT_EOF

Error Index

The End-of-line token.

TT_EOL

Error Index

The number token.

TT_NUMBER

Error Index

The word token.

TT_WORD

Error Index

The number value.

nval

Error Index

The Stream value.

sval

Error Index

The type of the last token returned.

ttype

Error Index

Error Index

StreamTokenizer(InputStream)

Creates a stream tokenizer that parses the specified input stream.

Error Index

Error Index

commentChar(int)

Specifies that this character starts a single line comment.

Error Index

eollsSignificant(boolean)

If the flag is true, end-of-lines are significant (TT_EOL will be returned by nexttoken).

Error Index

lineno()

Return the current line number.

Error Index

lowerCaseMode(boolean)

Examines a boolean to decide whether TT_WORD tokens are forced to be lower case.

Error Index

nextToken()

Parses a token from the input stream.

Error Index

ordinaryChar(int)

Specifies that this character is 'ordinary': it removes any significance as a word, comment, string, whitespace or number character.

Error Index

ordinaryChars(int, int)

Specifies that characters in this range are 'ordinary'.

Error Index

parseNumbers()

Specifies that numbers should be parsed.

Error Index

pushBack()

Pushes back a stream token.

Error Index

quoteChar(int)

Specifies that matching pairs of this character delimit String constants.

Error Index

resetSyntax()

Resets the syntax table so that all characters are special.

Error Index

slashSlashComments(boolean)

If the flag is true, recognize C++ style(//) comments.

Error Index

slashStarComments(boolean)

If the flag is true, recognize C style(/*) comments.

Error Index

toString()

Returns the String representation of the stream token.

Error Index

whitespaceChars(int, int)

Specifies that characters in this range are whitespace characters.

Error Index

wordChars(int, int)

Specifies that characters in this range are word characters.

Error Index

Error Index

ttype

public int ttype

The type of the last token returned. It's value will either be one of the following TT_* constants, or a single character. For example, if '+' is encountered and is not a valid word character, ttype will be '+'

Error Index

TT_EOF

public final static int TT_EOF

The End-of-file token.

Error Index

TT_EOL

public final static int TT_EOL

The End-of-line token.

Error Index

TT_NUMBER

public final static int TT_NUMBER

The number token. This value is in nval.

Error Index TT_WORD

```
public final static int TT_WORD
```

The word token. This value is in sval.

Error Index sval

```
public String sval
```

The Stream value.

Error Index nval

```
public double nval
```

The number value.

Error Index
Error Index StreamTokenizer

```
public StreamTokenizer(InputStream I)
```

Creates a stream tokenizer that parses the specified input stream. By default, it recognizes numbers, Strings quoted with single and double quotes, and all the alphabetics.

Parameters:

I - the input stream

Error Index
Error Index resetSyntax


```
public void resetSyntax()
```

Resets the syntax table so that all characters are special.

Error Index wordChars

```
public void wordChars(int low,  
                      int hi)
```

Specifies that characters in this range are word characters.

Parameters:

low - the low end of the range
hi - the high end of the range

Error Index whitespaceChars

```
public void whitespaceChars(int low,  
                           int hi)
```

Specifies that characters in this range are whitespace characters.

Parameters:

low - the low end of the range
hi - the high end of the range

Error Index ordinaryChars

```
public void ordinaryChars(int low,  
                          int hi)
```

Specifies that characters in this range are 'ordinary'. Ordinary characters mean that any significance as words, comments, strings, whitespaces or number characters are removed. When these characters are encountered by the parser, they return a ttype equal to the character.

Parameters:

low - the low end of the range
hi - the high end of the range

Error Index ordinaryChar

```
public void ordinaryChar(int ch)
```

Specifies that this character is 'ordinary': it removes any significance as a word, comment, string, whitespace or number character. When encountered by the parser, it returns a ttype equal to the character.

Parameters:

ch - the character

Error Index commentChar

```
public void commentChar(int ch)
```

Specifies that this character starts a single line comment.

Parameters:

ch - the character

Error Index quoteChar

```
public void quoteChar(int ch)
```

Specifies that matching pairs of this character delimit String constants. When a String constant is recognized, ttype will be the character that delimits the String, and sval will have the body of the String.

Parameters:

ch - the character

Error Index parseNumbers

```
public void parseNumbers()
```

Specifies that numbers should be parsed. This method accepts double precision floating point numbers and returns a ttype of TT_NUMBER with the value in nval.

Error Index eolIsSignificant

```
public void eolIsSignificant(boolean flag)
```

If the flag is true, end-of-lines are significant (TT_EOL will be returned by nexttoken). If false, they will

be treated as whitespace.

Error Index slashStarComments

```
public void slashStarComments(boolean flag)
```

If the flag is true, recognize C style(/*) comments.

Error Index slashSlashComments

```
public void slashSlashComments(boolean flag)
```

If the flag is true, recognize C++ style(//) comments.

Error Index lowerCaseMode

```
public void lowerCaseMode(boolean fl)
```

Examines a boolean to decide whether TT_WORD tokens are forced to be lower case.

Parameters:

fl - the boolean flag

Error Index nextToken

```
public int nextToken() throws IOException
```

Parses a token from the input stream. The return value is the same as the value of ttype. Typical clients of this class first set up the syntax tables and then sit in a loop calling nextToken to parse successive tokens until TT_EOF is returned.

Error Index pushBack

```
public void pushBack()
```

Pushes back a stream token.

Error Index lineno

```
public int lineno()
```

Return the current line number.

Error Index toString

```
public String toString()
```

Returns the String representation of the stream token.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.StringBufferInputStream

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.StringBufferInputStream

[java.lang.Object](#)

```
|
+----java.io.InputStream
      |
      +----java.io.StringBufferInputStream
```

```
public class StringBufferInputStream
    extends InputStream
```

This class implements a String buffer that can be used as an InputStream.

Error Index

Error Index

buffer

The buffer where data is stored.

Error Index

count

The number of characters to use in the buffer.

Error Index

pos

The position in the buffer.

Error Index

Error Index

StringBufferInputStream(String)

Creates an StringBufferInputStream from the specified array of bytes.

Error Index

Error Index

available()

Returns the number of available bytes in the buffer.

Error Index

read()

Reads a byte of data.

Error Index

read(byte[], int, int)

Reads into an array of bytes.

Error Index

reset()

Resets the buffer to the beginning.

Error Index

skip(long)

Skips n bytes of input.

Error Index

Error Index

buffer

protected String buffer

The buffer where data is stored.

Error Index

pos

protected int pos

The position in the buffer.

Error Index

count

protected int count

The number of characters to use in the buffer.

Error Index

Error Index

StringBufferInputStream

```
public StringBufferInputStream(String s)
```

Creates an StringBufferInputStream from the specified array of bytes.

Parameters:

s - the input buffer (not copied)

Error Index

Error Index

read

```
public synchronized int read()
```

Reads a byte of data.

Returns:

the byte read, or -1 if the end of the stream is reached.

Overrides:

read in class InputStream

Error Index

read

```
public synchronized int read(byte b[],  
                             int off,  
                             int len)
```

Reads into an array of bytes.

Parameters:

b - the buffer into which the data is read
off - the start offset of the data
len - the maximum number of bytes read

Returns:

the actual number of bytes read; -1 is returned when the end of the stream is reached.

Overrides:

read in class InputStream

Error Index

skip

public synchronized long skip(long n)

Skips n bytes of input.

Parameters:

n - the number of bytes to be skipped

Returns:

the actual number of bytes skipped.

Overrides:

skip in class InputStream

Error Index available

public synchronized int available()

Returns the number of available bytes in the buffer.

Overrides:

available in class InputStream

Error Index reset

public synchronized void reset()

Resets the buffer to the beginning.

Overrides:

reset in class InputStream

Class java.io.UTFDataFormatException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.io.UTFDataFormatException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.io.IOException
|
+----java.io.UTFDataFormatException
```

```
public class UTFDataFormatException
extends IOException
```

Signals that a malformed UTF-8 string has been read in a DataInput stream.

See Also:

[IOException](#), [DataInput](#)

Error Index

Error Index

UTFDataFormatException()

Constructs an UTFDataFormatException with no detail message.

Error Index

UTFDataFormatException(String)

Constructs an UTFDataFormatException with the specified detail message.

Error Index

Error Index

UTFDataFormatException

```
public UTFDataFormatException()
```

Constructs an UTFDataFormatException with no detail message. A detail message is a String that describes this particular exception.

Error Index

UTFDataFormatException

```
public UTFDataFormatException(String s)
```

Constructs an UTFDataFormatException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.AbstractMethodError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.AbstractMethodError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
                  |
                  +----java.lang.IncompatibleClassChangeError
                        |
                        +----java.lang.AbstractMethodError
```

```
public class AbstractMethodError
    extends IncompatibleClassChangeError
```

Signals an attempt to call an abstract method.

Error Index

Error Index

AbstractMethodError()

Constructs an AbstractMethodError with no detail message.

Error Index

AbstractMethodError(String)

Constructs an AbstractMethodError with the specified detail message.

Error Index

Error Index

AbstractMethodError

```
public AbstractMethodError()
```

Constructs an AbstractMethodError with no detail message. A detail message is a String that describes this particular exception.

Error Index

AbstractMethodError

```
public AbstractMethodError(String s)
```

Constructs an AbstractMethodError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String that contains the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ArithmeticException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ArithmeticException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.RuntimeException
|
+----java.lang.ArithmeticException
```

```
public class ArithmeticException
extends RuntimeException
```

Signals that an exceptional arithmetic condition has occurred. For example, dividing by zero would invoke this class.

Error Index

Error Index

[**ArithmeticException\(\)**](#)

Constructs an ArithmeticException with no detail message.

Error Index

[**ArithmeticException\(String\)**](#)

Constructs an ArithmeticException with the specified detail message.

Error Index

Error Index

ArithmeticException

```
public ArithmeticException()
```

Constructs an ArithmeticException with no detail message. A detail message is a String that

describes this particular exception.

Error Index **ArithmeticException**

```
public ArithmeticException(String s)
```

Constructs an ArithmeticException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String that contains a detailed message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.ArrayIndexOutOfBoundsException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.ArrayIndexOutOfBoundsException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IndexOutOfBoundsException
                        |
                        +----java.lang.ArrayIndexOutOfBoundsException
```

```
public class ArrayIndexOutOfBoundsException
    extends IndexOutOfBoundsException
```

Signals that an invalid array index has been used.

Error Index

Error Index

[ArrayIndexOutOfBoundsException\(\)](#)

Constructs an ArrayIndexOutOfBoundsException with no detail message.

Error Index

[ArrayIndexOutOfBoundsException\(int\)](#)

Constructs a new ArrayIndexOutOfBoundsException class initialized to the specific index.

Error Index

[ArrayIndexOutOfBoundsException\(String\)](#)

Constructs an ArrayIndexOutOfBoundsException class with the specified detail message.

Error Index

Error Index

ArrayIndexOutOfBoundsException

```
public ArrayIndexOutOfBoundsException()
```

Constructs an ArrayIndexOutOfBoundsException with no detail message. A detail message is a String that describes this particular exception.

Error Index

ArrayIndexOutOfBoundsException

```
public ArrayIndexOutOfBoundsException(int index)
```

Constructs a new ArrayIndexOutOfBoundsException class initialized to the specific index.

Parameters:

index - the index where the error occurred

Error Index

ArrayIndexOutOfBoundsException

```
public ArrayIndexOutOfBoundsException(String s)
```

Constructs an ArrayIndexOutOfBoundsException class with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String containing a detail message

Class java.lang.ArrayStoreException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ArrayStoreException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.RuntimeException
|
+----java.lang.ArrayStoreException
```

```
public class ArrayStoreException
extends RuntimeException
```

An attempt has been made to store the wrong type of Object to an array.

Error Index

Error Index

[**ArrayStoreException\(\)**](#)

Constructs a ArrayStoreException with no detail message.

Error Index

[**ArrayStoreException\(String\)**](#)

Constructs a ArrayStoreException with the specified detail message.

Error Index

Error Index

ArrayStoreException

```
public ArrayStoreException()
```

Constructs a ArrayStoreException with no detail message. A detail message is a String that describes this particular exception.

Error Index

ArrayStoreException

public ArrayStoreException(String s)

Constructs a ArrayStoreException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String containing a detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Boolean

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Boolean

[java.lang.Object](#)

|
+----java.lang.Boolean

public final class **Boolean**
extends [Object](#)

The Boolean class provides an object wrapper for Boolean data values, and serves as a place for boolean-oriented operations. A wrapper is useful because most of Java's utility classes require the use of objects. Since booleans are not objects in Java, they need to be "wrapped" in a Boolean instance.

Error Index

Error Index

FALSE

Assigns this Boolean to be false.

Error Index

MAX_VALUE

The maximum value a Character can have.

Error Index

MIN_VALUE

The minimum value a Character can have.

Error Index

TRUE

Assigns this Boolean to be true.

Error Index

Error Index

Boolean(boolean)

Constructs a Boolean object initialized to the specified boolean value.

Error Index

Boolean(String)

Constructs a Boolean object initialized to the value specified by the String parameter.

Error Index

Error Index

booleanValue()

Returns the value of this Boolean object as a boolean.

Error Index

equals(Object)

Compares this object against the specified object.

Error Index

getBoolean(String)

Gets a Boolean from the properties.

Error Index

hashCode()

Returns a hashCode for this Boolean.

Error Index

toString()

Returns a new String object representing this Boolean's value.

Error Index

valueOf(String)

Returns the boolean value represented by the specified String.

Error Index

Error Index

TRUE

public final static **Boolean** TRUE

Assigns this Boolean to be true.

Error Index

FALSE

public final static **Boolean** FALSE

Assigns this Boolean to be false.

Error Index MIN_VALUE

```
public final static char MIN_VALUE
```

The minimum value a Character can have. The lowest minimum value an Integer can have is .

Error Index MAX_VALUE

```
public final static char MAX_VALUE
```

The maximum value a Character can have. The greatest maximum value an Integer can have is .

Error Index

Error Index Boolean

```
public Boolean(boolean value)
```

Constructs a Boolean object initialized to the specified boolean value.

Parameters:

value - the value of the boolean

Error Index Boolean

```
public Boolean(String s)
```

Constructs a Boolean object initialized to the value specified by the String parameter.

Parameters:

s - the String to be converted to a Boolean

Error Index

Error Index booleanValue

```
public boolean booleanValue()
```

Returns the value of this Boolean object as a boolean.

Error Index **valueOf**

```
public static Boolean valueOf(String s)
```

Returns the boolean value represented by the specified String.

Parameters:

s - the String to be parsed

Error Index **toString**

```
public String toString()
```

Returns a new String object representing this Boolean's value.

Overrides:

toString in class Object

Error Index **hashCode**

```
public int hashCode()
```

Returns a hashcode for this Boolean.

Overrides:

hashCode in class Object

Error Index **equals**

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

Compares this object against the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index

getBoolean

public static boolean getBoolean(String name)

Gets a Boolean from the properties.

Parameters:

name - the property name.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Character

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Character

[java.lang.Object](#)

|

+----java.lang.Character

public final class **Character**
extends [Object](#)

The Character class provides an object wrapper for Character data values and serves as a place for character-oriented operations. A wrapper is useful because most of Java's utility classes require the use of objects. Since characters are not objects in Java, they need to be "wrapped" in a Character instance.

Error Index

Error Index

[MAX_RADIX](#)

The maximum radix available for conversion to and from Strings.

Error Index

[MIN_RADIX](#)

The minimum radix available for conversion to and from Strings.

Error Index

Error Index

[Character\(char\)](#)

Constructs a Character object with the specified value.

Error Index

Error Index

[charValue\(\)](#)

Returns the value of this Character object.

Error Index

digit(char, int)

Returns the numeric value of the character digit using the specified radix.

Error Index

equals(Object)

Compares this object against the specified object.

Error Index

forDigit(int, int)

Returns the character value for the specified digit in the specified radix.

Error Index

hashCode()

Returns a hashcode for this Character.

Error Index

isDigit(char)

Determines if the specified character is a ISO-LATIN-1 digit.

Error Index

isLowerCase(char)

Determines if the specified character is ISO-LATIN-1 lower case.

Error Index

isSpace(char)

Determines if the specified character is ISO-LATIN-1 white space according to Java.

Error Index

isUpperCase(char)

Determines if the specified character is ISO-LATIN-1 upper case.

Error Index

toLowerCase(char)

Returns the lower case character value of the specified ISO-LATIN-1 character.

Error Index

toString()

Returns a String object representing this character's value.

Error Index

toUpperCase(char)

Returns the upper case character value of the specified ISO-LATIN-1 character.

Error Index

Error Index

MIN_RADIX

public final static int MIN_RADIX

The minimum radix available for conversion to and from Strings. The lowest minimum value that a radix can be is 2.

See Also:

toString

Error Index

MAX_RADIX

```
public final static int MAX_RADIX
```

The maximum radix available for conversion to and from Strings. The largest maximum value that a radix can have is 36.

See Also:

toString

Error Index

Error Index

Character

```
public Character(char value)
```

Constructs a Character object with the specified value.

Parameters:

value - value of this Character object

Error Index

Error Index

isLowerCase

```
public static boolean isLowerCase(char ch)
```

Determines if the specified character is ISO-LATIN-1 lower case.

Parameters:

ch - the character to be tested

Returns:

true if the character is lower case; false otherwise.

Error Index isUpperCase

```
public static boolean isUpperCase(char ch)
```

Determines if the specified character is ISO-LATIN-1 upper case.

Parameters:

ch - the character to be tested

Returns:

true if the character is upper case; false otherwise.

Error Index isDigit

```
public static boolean isDigit(char ch)
```

Determines if the specified character is a ISO-LATIN-1 digit.

Parameters:

ch - the character to be tested

Returns:

true if this character is a digit; false otherwise.

Error Index isSpace

```
public static boolean isSpace(char ch)
```

Determines if the specified character is ISO-LATIN-1 white space according to Java.

Parameters:

ch - the character to be tested

Returns:

true if the character is white space; false otherwise.

Error Index toLowerCase

```
public static char toLowerCase(char ch)
```

Returns the lower case character value of the specified ISO-LATIN-1 character. Characters that are not upper case letters are returned unmodified.

Parameters:

ch - the character to be converted

Error Index toUpperCase

```
public static char toUpperCase(char ch)
```

Returns the upper case character value of the specified ISO-LATIN-1 character. Characters that are not lower case letters are returned unmodified. Note that German ess-zed and latin small letter y diaeresis have no corresponding upper case letters, even though they are lower case. There is a capital y diaeresis, but not in ISO-LATIN-1...

Parameters:

ch - the character to be converted

Error Index digit

```
public static int digit(char ch,  
                        int radix)
```

Returns the numeric value of the character digit using the specified radix. If the character is not a valid digit, it returns -1.

Parameters:

ch - the character to be converted

radix - the radix

Error Index forDigit

```
public static char forDigit(int digit,  
                           int radix)
```

Returns the character value for the specified digit in the specified radix. If the digit is not valid in the radix, the 0 character is returned.

Parameters:

digit - the digit chosen by the character value

radix - the radix containing the digit

Error Index charValue

```
public char charValue()
```

Returns the value of this Character object.

Error Index hashCode

```
public int hashCode()
```

Returns a hashcode for this Character.

Overrides:

hashCode in class Object

Error Index equals

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

Compares this object against the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index toString

```
public String toString()
```

Returns a String object representing this character's value.

Overrides:

toString in class Object

Class java.lang.Class

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Class

java.lang.Object
|
+----java.lang.Class

public final class **Class**
extends Object

Class objects contain runtime representations of classes. Every object in the system is an instance of some Class, and for each Class there is one of these descriptor objects. A Class descriptor is not modifiable at runtime.

The following example uses a Class object to print the Class name of an object:

```
void printClassName(Object obj) {  
    System.out.println("The class of " + obj +  
        " is " + obj.getClass().getName());  
}
```

Error Index

Error Index

forName(String)

Returns the runtime Class descriptor for the specified Class.

Error Index

getClassLoader()

Returns the Class loader of this Class.

Error Index

getInterfaces()

Returns the interfaces of this Class.

Error Index

getName()

Returns the name of this Class.

Error Index

getSuperclass()

Returns the superclass of this Class.

Error Index

isInterface()

Returns a boolean indicating whether or not this Class is an interface.

Error Index

newInstance()

Creates a new instance of this Class.

Error Index

toString()

Returns the name of this class or interface.

Error Index

Error Index

forName

public static Class forName(String className) throws ClassNotFoundException

Returns the runtime Class descriptor for the specified Class. For example, the following code fragment returns the runtime Class descriptor for the Class named java.lang.Thread:

```
Class t = Class.forName("java.lang.Thread")
```

Parameters:

className - the fully qualified name of the desired Class

Throws: ClassNotFoundException

If the Class could not be found.

Error Index

newInstance

public Object newInstance() throws InstantiationException, IllegalAccessException

Creates a new instance of this Class.

Returns:

the new instance of this Class.

Throws: InstantiationException

If you try to instantiate an abstract class or an interface, or if the instantiation fails for some other reason.

Throws: IllegalAccessException

If the class or initializer is not accessible.

Error Index **getName**

```
public String getName()
```

Returns the name of this Class.

Error Index **getSuperclass**

```
public Class getSuperclass()
```

Returns the superclass of this Class.

Error Index **getInterfaces**

```
public Class[] getInterfaces()
```

Returns the interfaces of this Class. An array of length 0 is returned if this Class implements no interfaces.

Error Index **getClassLoader**

```
public ClassLoader getClassLoader()
```

Returns the Class loader of this Class. Returns null if this Class does not have a Class loader.

See Also:

ClassLoader

Error Index **isInterface**

```
public boolean isInterface()
```

Returns a boolean indicating whether or not this Class is an interface.

Error Index **toString**

```
public String toString()
```

Returns the name of this class or interface. The word "class" is prepended if it is a Class; the word

"interface" is prepended if it is an interface.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassCastException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassCastException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.RuntimeException
|
+----java.lang.ClassCastException
```

```
public class ClassCastException
extends RuntimeException
```

Signals that an invalid cast has occurred.

Error Index

Error Index

ClassCastException()

Constructs a ClassCastException with no detail message.

Error Index

ClassCastException(String)

Constructs a ClassCastException with the specified detail message.

Error Index

Error Index

ClassCastException

```
public ClassCastException()
```

Constructs a ClassCastException with no detail message. A detail message is a String that describes this particular exception.

Error Index

ClassCastException

public ClassCastException(String s)

Constructs a ClassCastException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String containing a detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassCircularityError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassCircularityError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
                  |
                  +----java.lang.ClassCircularityError
```

```
public class ClassCircularityError
    extends LinkageError
```

Signals that a circularity has been detected when initializing a class.

Error Index

Error Index

ClassCircularityError()

Constructs a ClassCircularityError with no detail message.

Error Index

ClassCircularityError(String)

Constructs a ClassCircularityError with the specified detail message.

Error Index

Error Index

ClassCircularityError

```
public ClassCircularityError()
```

Constructs a ClassCircularityError with no detail message. A detail message is a String that describes this particular exception.

Error Index

ClassCircularityError

public ClassCircularityError(String s)

Constructs a ClassCircularityError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassFormatError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassFormatError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.LinkageError
|
+----java.lang.ClassFormatError
```

```
public class ClassFormatError
extends LinkageError
```

Signals an invalid file format has occurred.

Error Index

Error Index

[**ClassFormatError\(\)**](#)

Constructs a ClassFormatError with no detail message.

Error Index

[**ClassFormatError\(String\)**](#)

Constructs a ClassFormatError with the specified detail message.

Error Index

Error Index

ClassFormatError

```
public ClassFormatError()
```

Constructs a ClassFormatError with no detail message. A detail message is a String that describes this particular exception.

Error Index

ClassFormatError

public ClassFormatError(String s)

Constructs a ClassFormatError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String containing the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassLoader

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassLoader

[java.lang.Object](#)

|
+----java.lang.ClassLoader

public class **ClassLoader**
extends [Object](#)

ClassLoader is an abstract Class that can be used to define a policy for loading Java classes into the runtime environment. By default, the runtime system loads classes that originate as files by reading them from the directory defined by the CLASSPATH environment variable (this is platform dependent). The default mechanism does not involve a Class loader.

However, some classes may not originate from a file; they could be loaded from some other source, e.g., the network. Classes loaded from the network are an array of bytes. A ClassLoader can be used to tell the runtime system to convert an array of bytes into an instance of class Class. This conversion information is passed to the runtime using the defineClass() method.

Classes that are created through the defineClass() mechanism can reference other classes by name. To resolve those names, the runtime system calls the ClassLoader that originally created the Class. The runtime system calls the abstract method loadClass() to load the referenced classes.

```
ClassLoader loader = new NetworkClassLoader(host, port);
Object main = loader.loadClass("Main").newInstance();
....
```

The NetworkClassLoader subclass must define the method loadClass() to load a Class from the network. Once it has downloaded the bytes that make up the Class it should use the method defineClass() to create a Class instance. A sample implementation could be:

```
class NetworkClassLoader {
    String host;
    int port;
    Hashtable cache = new Hashtable();
    private byte loadClassData(String name)[] {
        // load the class data from the connection
        ...
    }
    public synchronized Class loadClass(String name) {
        Class c = cache.get(name);
```



```

    if (c == null) {
        byte data[] = loadClassData(name);
        cache.put(name, defineClass(data, 0, data.length));
    }
    return c;
}
}

```

See Also:

[Class](#)

Error Index

Error Index

[ClassLoader\(\)](#)

Constructs a new Class loader and initializes it.

Error Index

Error Index

[defineClass\(byte\[\], int, int\)](#)

Converts an array of bytes to an instance of class Class.

Error Index

[findSystemClass\(String\)](#)

Loads a system Class.

Error Index

[loadClass\(String, boolean\)](#)

Resolves the specified name to a Class.

Error Index

[resolveClass\(Class\)](#)

Resolves classes referenced by this Class.

Error Index

Error Index

ClassLoader

protected ClassLoader()

Constructs a new Class loader and initializes it.

Error Index

Error Index

loadClass

protected abstract Class loadClass(String name,
boolean resolve) throws ClassNotFoundException

Resolves the specified name to a Class. The method loadClass() is called by the virtual machine. As an abstract method, loadClass() must be defined in a subclass of ClassLoader. By using a Hashtable, you can avoid loading the same Class more than once.

Parameters:

name - the name of the desired Class
resolve - true if the Class needs to be resolved

Returns:

the resulting Class, or null if it was not found.

Throws: ClassNotFoundException

Cannot find a definition for the class

See Also:

Hashtable

Error Index

defineClass

protected final Class defineClass(byte data[],
int offset,
int length)

Converts an array of bytes to an instance of class Class. Before the Class can be used it must be resolved.

Parameters:

data - the bytes that make up the Class
offset - the start offset of the Class data
length - the length of the Class data

Returns:

the Class object which was created from the data.

Throws: ClassFormatError

If the data does not contain a valid Class.

See Also:

loadClass, resolveClass

Error Index

resolveClass

protected final void resolveClass(Class c)

Resolves classes referenced by this Class. This must be done before the Class can be used. Class names referenced by the resulting Class are resolved by calling loadClass().

Parameters:

c - the Class to be resolved

See Also:

[defineClass](#)

Error Index

findSystemClass

protected final [Class](#) findSystemClass([String](#) name) throws [ClassNotFoundException](#)

Loads a system Class. A system Class is a class with the primordial Class loader (which is null).

Parameters:

name - the name of the system Class

Throws: [NoClassDefFoundError](#)

If the Class is not found.

Throws: [ClassNotFoundException](#)

Cannot find a definition for the class

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassNotFoundException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ClassNotFoundException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.ClassNotFoundException
```

```
public class ClassNotFoundException
    extends Exception
```

Signals that a class could not be found.

Error Index

Error Index

[ClassNotFoundException\(\)](#)

Constructs a ClassNotFoundException with no detail message.

Error Index

[ClassNotFoundException\(String\)](#)

Constructs a ClassNotFoundException with the specified detail message.

Error Index

Error Index

ClassNotFoundException

```
public ClassNotFoundException()
```

Constructs a ClassNotFoundException with no detail message. A detail message is a String that describes this particular exception.

Error Index

ClassNotFoundException

public ClassNotFoundException(String s)

Constructs a ClassNotFoundException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.lang.Cloneable

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.lang.Cloneable

public interface **Cloneable**
extends [Object](#)

An interface indicating that this object may be copied or cloned.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.CloneNotSupportedException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.CloneNotSupportedException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.CloneNotSupportedException
```

```
public class CloneNotSupportedException
    extends Exception
```

Signals that an attempt has been made to clone an object that does not want to be cloned.

Error Index

Error Index

CloneNotSupportedException()

Constructs an CloneNotSupportedException with no detail message.

Error Index

CloneNotSupportedException(String)

Constructs an CloneNotSupportedException with the specified detail message.

Error Index

Error Index

CloneNotSupportedException

```
public CloneNotSupportedException()
```

Constructs an CloneNotSupportedException with no detail message. A detail message is a String

that describes this particular exception.

Error Index

CloneNotSupportedException

```
public CloneNotSupportedException(String s)
```

Constructs an CloneNotSupportedException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Compiler

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Compiler

[java.lang.Object](#)

|

+----[java.lang.Compiler](#)

public final class **Compiler**
extends [Object](#)

Error Index

Error Index

[command](#)(Object)

Error Index

[compileClass](#)(Class)

Error Index

[compileClasses](#)(String)

Error Index

[disable](#)()

Error Index

[enable](#)()

Error Index

Error Index

[compileClass](#)

public static boolean [compileClass](#)([Class](#) clazz)

Error Index compileClasses

public static boolean compileClasses(String string)

Error Index command

public static Object command(Object any)

Error Index enable

public static void enable()

Error Index disable

public static void disable()

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Double

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Double

[java.lang.Object](#)

```
|
+----java.lang.Number
      |
      +----java.lang.Double
```

```
public final class Double
    extends Number
```

The Double class provides an object wrapper for Double data values and serves as a place for double-oriented operations. A wrapper is useful because most of Java's utility classes require the use of objects. Since doubles are not objects in Java, they need to be "wrapped" in a Double instance.

Error Index

Error Index

MAX_VALUE

The maximum value a double can have.

Error Index

MIN_VALUE

The minimum value a double can have.

Error Index

NEGATIVE_INFINITY

Negative infinity.

Error Index

NaN

Not-a-Number.

Error Index

POSITIVE_INFINITY

Positive infinity.

Error Index

Error Index

Double(double)

Constructs a Double wrapper for the specified double value.

Error Index

Double(String)

Constructs a Double object initialized to the value specified by the String parameter.

Error Index

Error Index

doubleToLongBits(double)

Returns the bit representation of a double-float value

Error Index

doubleValue()

Returns the double value of this Double.

Error Index

equals(Object)

Compares this object against the specified object.

Error Index

floatValue()

Returns the float value of this Double.

Error Index

hashCode()

Returns a hashcode for this Double.

Error Index

intValue()

Returns the integer value of this Double (by casting to an int).

Error Index

isInfinite(double)

Returns true if the specified number is infinitely large in magnitude.

Error Index

isInfinite()

Returns true if this Double value is infinitely large in magnitude.

Error Index

isNaN(double)

Returns true if the specified number is the special Not-a-Number (NaN) value.

Error Index

isNaN()

Returns true if this Double value is the special Not-a-Number (NaN) value.

Error Index

longBitsToDouble(long)

Returns the double-float corresponding to a given bit representation.

Error Index

longValue()

Returns the long value of this Double (by casting to a long).

Error Index

toString(double)

Returns a String representation for the specified double value.

Error Index

toString()

Returns a String representation of this Double object.

Error Index

valueOf(String)

Returns a new Double value initialized to the value represented by the specified String.

Error Index

Error Index

POSITIVE_INFINITY

public final static double POSITIVE_INFINITY

Positive infinity.

Error Index

NEGATIVE_INFINITY

public final static double NEGATIVE_INFINITY

Negative infinity.

Error Index

NaN

public final static double NaN

Not-a-Number. *Note: is not equal to anything, including itself*

Error Index

MAX_VALUE

public final static double MAX_VALUE

The maximum value a double can have. The greatest maximum value that a double can have is 1.79769313486231570e+308d.

Error Index

MIN_VALUE

public final static double MIN_VALUE

The minimum value a double can have. The lowest minimum value that a double can have is 4.94065645841246544e-324d.

Error Index

Error Index Double

public Double(double value)

Constructs a Double wrapper for the specified double value.

Parameters:

value - the initial value of the double

Error Index Double

public Double(String s) throws NumberFormatException

Constructs a Double object initialized to the value specified by the String parameter.

Parameters:

s - the String to be converted to a Double

Throws: NumberFormatException

If the String does not contain a parsable number.

Error Index

Error Index toString

public static String toString(double d)

Returns a String representation for the specified double value.

Parameters:

d - the double to be converted

Error Index **valueOf**

public static Double valueOf(String s) throws NumberFormatException

Returns a new Double value initialized to the value represented by the specified String.

Parameters:

s - the String to be parsed

Throws: NumberFormatException

If the String cannot be parsed.

Error Index **isNaN**

public static boolean isNaN(double v)

Returns true if the specified number is the special Not-a-Number (NaN) value.

Parameters:

v - the value to be tested

Error Index **isInfinite**

public static boolean isInfinite(double v)

Returns true if the specified number is infinitely large in magnitude.

Parameters:

v - the value to be tested

Error Index **isNaN**

public boolean isNaN()

Returns true if this Double value is the special Not-a-Number (NaN) value.

Error Index **isInfinite**

public boolean isInfinite()

Returns true if this Double value is infinitely large in magnitude.

Error Index toString

```
public String toString()
```

Returns a String representation of this Double object.

Overrides:

toString in class Object

Error Index intValue

```
public int intValue()
```

Returns the integer value of this Double (by casting to an int).

Overrides:

intValue in class Number

Error Index longValue

```
public long longValue()
```

Returns the long value of this Double (by casting to a long).

Overrides:

longValue in class Number

Error Index floatValue

```
public float floatValue()
```

Returns the float value of this Double.

Overrides:

floatValue in class Number

Error Index doubleValue

```
public double doubleValue()
```


Returns the double value of this Double.

Overrides:

doubleValue in class Number

Error Index hashCode

public int hashCode()

Returns a hashCode for this Double.

Overrides:

hashCode in class Object

Error Index equals

public boolean equals(Object obj)

Compares this object against the specified object.

Note: To be useful in hashtables this method considers two NaN double values to be equal. This is not according to IEEE specification

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index doubleToLongBits

public static long doubleToLongBits(double value)

Returns the bit representation of a double-float value

Error Index longBitsToDouble

public static double longBitsToDouble(long bits)

Returns the double-float corresponding to a given bit representation.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Error

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Error

[java.lang.Object](#)
|
+----[java.lang.Throwable](#)
|
+----java.lang.Error

```
public class Error  
extends Throwable
```

Error is a subtype of Throwable for abnormal events that should not occur. Do not try to catch Error's unless you really know what you're doing.

Error Index

Error Index

Error()

Constructs an Error with no specified detail message.

Error Index

Error(String)

Constructs an Error with the specified detail message.

Error Index

Error Index

Error

```
public Error()
```

Constructs an Error with no specified detail message. A detail message is a String that describes this particular error.

Error Index

Error

public Error(String s)

Constructs an Error with the specified detail message. A detail message is a String that describes this particular error

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Exception

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Exception

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
```

```
public class Exception
    extends Throwable
```

Exception are a form of Throwable that normal programs may wish to try and catch.

Error Index

Error Index

Exception()

Constructs an Exception with no specified detail message.

Error Index

Exception(String)

Constructs a Exception with the specified detail message.

Error Index

Error Index

Exception

```
public Exception()
```

Constructs an Exception with no specified detail message. A detail message is a String that describes this particular exception.

Error Index

Exception

public Exception(String s)

Constructs a Exception with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Float

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Float

```
java.lang.Object
|
+----java.lang.Number
      |
      +----java.lang.Float
```

```
public final class Float
    extends Number
```

The Float class provides an object wrapper for Float data values, and serves as a place for float-oriented operations. A wrapper is useful because most of Java's utility classes require the use of objects. Since floats are not objects in Java, they need to be "wrapped" in a Float instance.

Error Index

Error Index

MAX_VALUE

The maximum value a float can have.

Error Index

MIN_VALUE

The minimum value a float can have.

Error Index

NEGATIVE_INFINITY

Negative infinity.

Error Index

NaN

Not-a-Number.

Error Index

POSITIVE_INFINITY

Positive infinity.

Error Index

Error Index

Float(float)

Constructs a Float wrapper for the specified float value.

Error Index

Float(double)

Constructs a Float wrapper for the specified double value.

Error Index

Float(String)

Constructs a Float object initialized to the value specified by the String parameter.

Error Index

Error Index

doubleValue()

Returns the double value of this Float.

Error Index

equals(Object)

Compares this object against some other object.

Error Index

floatToIntBits(float)

Returns the bit representation of a single-float value

Error Index

floatValue()

Returns the float value of this Float object.

Error Index

hashCode()

Returns a hashcode for this Float.

Error Index

intBitsToFloat(int)

Returns the single-float corresponding to a given bit representation.

Error Index

intValue()

Returns the integer value of this Float (by casting to an int).

Error Index

isInfinite(float)

Returns true if the specified number is infinitely large in magnitude.

Error Index

isInfinite()

Returns true if this Float value is infinitely large in magnitude.

Error Index

isNaN(float)

Returns true if the specified number is the special Not-a-Number (NaN) value.

Error Index

isNaN()

Returns true if this Float value is Not-a-Number (NaN).

Error Index

longValue()

Returns the long value of this Float (by casting to a long).

Error Index

toString(float)

Returns a String representation for the specified float value.

Error Index

toString()

Returns a String representation of this Float object.

Error Index

valueOf(String)

Returns the floating point value represented by the specified String.

Error Index

Error Index

POSITIVE_INFINITY

public final static float POSITIVE_INFINITY

Positive infinity.

Error Index

NEGATIVE_INFINITY

public final static float NEGATIVE_INFINITY

Negative infinity.

Error Index

NaN

public final static float NaN

Not-a-Number. *Note: is not equal to anything, including itself*

Error Index

MAX_VALUE

public final static float MAX_VALUE

The maximum value a float can have. The largest maximum value possible is 3.40282346638528860e+38.

Error Index MIN_VALUE

public final static float MIN_VALUE

The minimum value a float can have. The lowest minimum value possible is 1.40129846432481707e-45.

Error Index

Error Index Float

public Float(float value)

Constructs a Float wrapper for the specified float value.

Parameters:

value - the value of the Float

Error Index Float

public Float(double value)

Constructs a Float wrapper for the specified double value.

Parameters:

value - the value of the Float

Error Index Float

public Float(String s) throws NumberFormatException

Constructs a Float object initialized to the value specified by the String parameter.

Parameters:

s - the String to be converted to a Float

Throws: NumberFormatException

If the String does not contain a parsable number.

Error Index

Error Index toString

public static String toString(float f)

Returns a String representation for the specified float value.

Parameters:

f - the float to be converted

Error Index valueOf

public static Float valueOf(String s) throws NumberFormatException

Returns the floating point value represented by the specified String.

Parameters:

s - the String to be parsed

Throws: NumberFormatException

If the String does not contain a parsable Float.

Error Index isNaN

public static boolean isNaN(float v)

Returns true if the specified number is the special Not-a-Number (NaN) value.

Parameters:

v - the value to be tested

Error Index isInfinite

public static boolean isInfinite(float v)

Returns true if the specified number is infinitely large in magnitude.

Parameters:

v - the value to be tested

Error Index isNaN

```
public boolean isNaN()
```

Returns true if this Float value is Not-a-Number (NaN).

Error Index isInfinite

```
public boolean isInfinite()
```

Returns true if this Float value is infinitely large in magnitude.

Error Index toString

```
public String toString()
```

Returns a String representation of this Float object.

Overrides:

toString in class Object

Error Index intValue

```
public int intValue()
```

Returns the integer value of this Float (by casting to an int).

Overrides:

intValue in class Number

Error Index longValue

```
public long longValue()
```

Returns the long value of this Float (by casting to a long).

Overrides:

longValue in class Number

Error Index floatValue

public float floatValue()

Returns the float value of this Float object.

Overrides:

floatValue in class Number

Error Index doubleValue

public double doubleValue()

Returns the double value of this Float.

Overrides:

doubleValue in class Number

Error Index hashCode

public int hashCode()

Returns a hashcode for this Float.

Overrides:

hashCode in class Object

Error Index equals

public boolean equals(Object obj)

Compares this object against some other object.

Note: To be useful in hashtables this method considers two Nan floating point values to be equal. This is not according to IEEE specification

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index floatToIntBits

```
public static int floatToIntBits(float value)
```

Returns the bit representation of a single-float value

Error Index intBitsToFloat

```
public static float intBitsToFloat(int bits)
```

Returns the single-float corresponding to a given bit representation.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalAccessException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalAccessException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.LinkageError
|
+----java.lang.IncompatibleClassChangeError
|
+----java.lang.IllegalAccessException
```

```
public class IllegalAccessException
    extends IncompatibleClassChangeError
```

Signals that an illegal access exception has occurred.

Error Index

Error Index

IllegalAccessException()

Constructs an IllegalAccessException with no detail message.

Error Index

IllegalAccessException(String)

Constructs an IllegalAccessException with the specified detail message.

Error Index

Error Index

IllegalAccessException

```
public IllegalAccessException()
```

Constructs an `IllegalAccessError` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index `IllegalAccessError`

```
public IllegalAccessError(String s)
```

Constructs an `IllegalAccessError` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalAccessException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalAccessException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.IllegalAccessException
```

```
public class IllegalAccessException
    extends Exception
```

Signals that a particular method could not be found.

Error Index

Error Index

[IllegalAccessException\(\)](#)

Constructs a `IllegalAccessException` without a detail message.

Error Index

[IllegalAccessException\(String\)](#)

Constructs a `IllegalAccessException` with a detail message.

Error Index

Error Index

`IllegalAccessException`

```
public IllegalAccessException()
```

Constructs a `IllegalAccessException` without a detail message. A detail message is a `String` that describes this particular exception.

Error Index

IllegalAccessException

public IllegalAccessException(String s)

Constructs a IllegalAccessException with a detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalArgumentException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalArgumentException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IllegalArgumentException
```

```
public class IllegalArgumentException
    extends RuntimeException
```

Signals that an illegal argument exception has occurred.

See Also:

[setPriority](#)

Error Index

Error Index

[**IllegalArgumentException\(\)**](#)

Constructs an IllegalArgumentException with no detail message.

Error Index

[**IllegalArgumentException\(String\)**](#)

Constructs an IllegalArgumentException with the specified detail message.

Error Index

Error Index

IllegalArgumentException

```
public IllegalArgumentException()
```

Constructs an `IllegalArgumentException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index

`IllegalArgumentException`

```
public IllegalArgumentException(String s)
```

Constructs an `IllegalArgumentException` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.IllegalMonitorStateException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalMonitorStateException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IllegalMonitorStateException
```

```
public class IllegalMonitorStateException
    extends RuntimeException
```

Signals that a monitor operation has been attempted when the monitor is in an invalid state. For example, trying to notify a monitor that you do not own would invoke this class.

Error Index

Error Index

IllegalMonitorStateException()

Constructs an `IllegalMonitorStateException` with no detail message.

Error Index

IllegalMonitorStateException(String)

Constructs an `IllegalMonitorStateException` with the specified detail message.

Error Index

Error Index

IllegalMonitorStateException

```
public IllegalMonitorStateException()
```

Constructs an `IllegalMonitorStateException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index `IllegalMonitorStateException`

```
public IllegalMonitorStateException(String s)
```

Constructs an `IllegalMonitorStateException` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the `String` that contains a detailed message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.IllegalThreadStateException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IllegalThreadStateException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IllegalArgumentException
                        |
                        +----java.lang.IllegalThreadStateException
```

```
public class IllegalThreadStateException
    extends IllegalArgumentException
```

Exception indicating that a thread is not in the proper state for the requested operation.

See Also:

[suspend](#), [resume](#)

Error Index

Error Index

[IllegalThreadStateException\(\)](#)

Constructs an `IllegalThreadStateException` with no detail message.

Error Index

[IllegalThreadStateException\(String\)](#)

Constructs an `IllegalThreadStateException` with the specified detail message.

Error Index

Error Index

IllegalThreadStateException

```
public IllegalThreadStateException()
```

Constructs an `IllegalThreadStateException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index

IllegalThreadStateException

```
public IllegalThreadStateException(String s)
```

Constructs an `IllegalThreadStateException` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.IncompatibleClassChangeError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IncompatibleClassChangeError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
                  |
                  +----java.lang.IncompatibleClassChangeError
```

```
public class IncompatibleClassChangeError
    extends LinkageError
```

Signals that an incompatible class change has occurred.

Error Index

Error Index

[**IncompatibleClassChangeError\(\)**](#)

Constructs an IncompatibleClassChangeError with no detail message.

Error Index

[**IncompatibleClassChangeError\(String\)**](#)

Constructs an IncompatibleClassChangeError with the specified detail message.

Error Index

Error Index

[**IncompatibleClassChangeError**](#)

```
public IncompatibleClassChangeError()
```

Constructs an IncompatibleClassChangeError with no detail message. A detail message is a String that describes this particular exception.

Error Index

IncompatibleClassChangeError

```
public IncompatibleClassChangeError(String s)
```

Constructs an IncompatibleClassChangeError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.IndexOutOfBoundsException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.IndexOutOfBoundsException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IndexOutOfBoundsException
```

```
public class IndexOutOfBoundsException
    extends RuntimeException
```

Signals that an index of some sort is out of bounds.

Error Index

Error Index

[**IndexOutOfBoundsException\(\)**](#)

Constructs an IndexOutOfBoundsException with no detail message.

Error Index

[**IndexOutOfBoundsException\(String\)**](#)

Constructs a IndexOutOfBoundsException with the specified detail message.

Error Index

Error Index

[**IndexOutOfBoundsException**](#)

```
public IndexOutOfBoundsException()
```

Constructs an IndexOutOfBoundsException with no detail message. A detail message is a String that describes this particular exception.

Error Index IndexOutOfBoundsException

```
public IndexOutOfBoundsException(String s)
```

Constructs a IndexOutOfBoundsException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.InstantiationError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.InstantiationError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.LinkageError
|
+----java.lang.IncompatibleClassChangeError
|
+----java.lang.InstantiationError
```

```
public class InstantiationError
    extends IncompatibleClassChangeError
```

Signals that the interpreter has tried to instantiate an abstract class or an interface.

Error Index

Error Index

InstantiationError()

Constructs an InstantiationError with no detail message.

Error Index

InstantiationError(String)

Constructs an InstantiationError with the specified detail message.

Error Index

Error Index

InstantiationError

```
public InstantiationError()
```

Constructs an `InstantiationError` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index `InstantiationError`

```
public InstantiationError(String s)
```

Constructs an `InstantiationError` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the `String` that contains the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.langInstantiationException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.langInstantiationException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.langInstantiationException
```

```
public class InstantiationException
    extends Exception
```

Signals that an attempt has been made to instantiate an abstract class or an interface.

Error Index

Error Index

InstantiationException()

Constructs an `InstantiationException` with no detail message.

Error Index

InstantiationException(String)

Constructs an `InstantiationException` with the specified detail message.

Error Index

Error Index

`InstantiationException`

```
public InstantiationException()
```

Constructs an `InstantiationException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index

InstantiationException

public InstantiationException(String s)

Constructs an InstantiationException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the String containing a detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Integer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Integer

[java.lang.Object](#)

|

+----[java.lang.Number](#)

|

+----java.lang.Integer

public final class **Integer**
extends [Number](#)

The Integer class is a wrapper for integer values. In Java, integers are not objects and most of the Java utility classes require the use of objects. Thus, if you needed to store an integer in a hashtable, you would have to "wrap" an Integer instance around it.

Error Index

Error Index

MAX_VALUE

The maximum value an Integer can have.

Error Index

MIN_VALUE

The minimum value an Integer can have.

Error Index

Error Index

Integer(int)

Constructs an Integer object initialized to the specified int value.

Error Index

Integer(String)

Constructs an Integer object initialized to the value specified by the String parameter.

Error Index

Error Index

doubleValue()

Returns the value of this Integer as a double.

Error Index

equals()(Object)

Compares this object to the specified object.

Error Index

floatValue()

Returns the value of this Integer as a float.

Error Index

getInteger()(String)

Gets an Integer property.

Error Index

getInteger()(String, int)

Gets an Integer property.

Error Index

getInteger()(String, Integer)

Gets an Integer property.

Error Index

hashCode()

Returns a hashCode for this Integer.

Error Index

intValue()

Returns the value of this Integer as an int.

Error Index

longValue()

Returns the value of this Integer as a long.

Error Index

parseInt()(String, int)

Assuming the specified String represents an integer, returns that integer's value.

Error Index

parseInt()(String)

Assuming the specified String represents an integer, returns that integer's value.

Error Index

toString()(int, int)

Returns a new String object representing the specified integer in the specified radix.

Error Index

toString()(int)

Returns a new String object representing the specified integer.

Error Index

toString()

Returns a String object representing this Integer's value.

Error Index

valueOf(String, int)

Assuming the specified String represents an integer, returns a new Integer object initialized to that value.

Error Index

valueOf(String)

Assuming the specified String represents an integer, returns a new Integer object initialized to that value.

Error Index

Error Index

MIN_VALUE

public final static int MIN_VALUE

The minimum value an Integer can have. The lowest minimum value an Integer can have is 0x80000000.

Error Index

MAX_VALUE

public final static int MAX_VALUE

The maximum value an Integer can have. The greatest maximum value an Integer can have is 0x7fffffff.

Error Index

Error Index

Integer

public Integer(int value)

Constructs an Integer object initialized to the specified int value.

Parameters:

value - the initial value of the Integer

Error Index

Integer

public Integer(String s) throws NumberFormatException

Constructs an Integer object initialized to the value specified by the String parameter. The radix is assumed to be 10.

Parameters:

s - the String to be converted to an Integer

Throws: NumberFormatException

If the String does not contain a parsable integer.

Error Index

Error Index toString

```
public static String toString(int i,  
                               int radix)
```

Returns a new String object representing the specified integer in the specified radix.

Parameters:

i - the integer to be converted

radix - the radix

See Also:

MIN_RADIX, MAX_RADIX

Error Index toString

```
public static String toString(int i)
```

Returns a new String object representing the specified integer. The radix is assumed to be 10.

Parameters:

i - the integer to be converted

Error Index parseInt

```
public static int parseInt(String s,  
                           int radix) throws NumberFormatException
```

Assuming the specified String represents an integer, returns that integer's value. Throws an exception if the String cannot be parsed as an int.

Parameters:

s - the String containing the integer

radix - the radix to be used

Throws: NumberFormatException

If the String does not contain a parsable integer.

Error Index parseInt

public static int parseInt(String s) throws NumberFormatException

Assuming the specified String represents an integer, returns that integer's value. Throws an exception if the String cannot be parsed as an int. The radix is assumed to be 10.

Parameters:

s - the String containing the integer

Throws: NumberFormatException

If the string does not contain a parsable integer.

Error Index valueOf

public static Integer valueOf(String s,
int radix) throws NumberFormatException

Assuming the specified String represents an integer, returns a new Integer object initialized to that value. Throws an exception if the String cannot be parsed as an int.

Parameters:

s - the String containing the integer

radix - the radix to be used

Throws: NumberFormatException

If the String does not contain a parsable integer.

Error Index valueOf

public static Integer valueOf(String s) throws NumberFormatException

Assuming the specified String represents an integer, returns a new Integer object initialized to that value. Throws an exception if the String cannot be parsed as an int. The radix is assumed to be 10.

Parameters:

s - the String containing the integer

Throws: NumberFormatException

If the String does not contain a parsable integer.

Error Index intValue

```
public int intValue()
```

Returns the value of this Integer as an int.

Overrides:

intValue in class Number

Error Index longValue

```
public long longValue()
```

Returns the value of this Integer as a long.

Overrides:

longValue in class Number

Error Index floatValue

```
public float floatValue()
```

Returns the value of this Integer as a float.

Overrides:

floatValue in class Number

Error Index doubleValue

```
public double doubleValue()
```

Returns the value of this Integer as a double.

Overrides:

doubleValue in class Number

Error Index toString

```
public String toString()
```

Returns a String object representing this Integer's value.

Overrides:

toString in class Object

Error Index hashCode

```
public int hashCode()
```

Returns a hashcode for this Integer.

Overrides:

hashCode in class Object

Error Index equals

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

Compares this object to the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index getInteger

```
public static Integer getInteger(String nm)
```

Gets an Integer property. If the property does not exist, it will return 0.

Parameters:

nm - the property name

Error Index getInteger

```
public static Integer getInteger(String nm,  
int val)
```

Gets an Integer property. If the property does not exist, it will return val. Deals with Hexadecimal and octal numbers.

Parameters:

nm - the String name

val - the Integer value

Error Index `getInteger`

```
public static Integer getInteger(String nm,  
                                Integer val)
```

Gets an Integer property. If the property does not exist, it will return val. Deals with Hexadecimal and octal numbers.

Parameters:

nm - the property name

val - the integer value

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.InternalError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.InternalError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.VirtualMachineError
|
+----java.lang.InternalError
```

```
public class InternalError
extends VirtualMachineError
```

Signals that an internal error has occurred.

Error Index

Error Index

[InternalError\(\)](#)

Constructs an InternalError with no detail message.

Error Index

[InternalError\(String\)](#)

Constructs an InternalError with the specified detail message.

Error Index

Error Index

InternalError

```
public InternalError()
```

Constructs an InternalError with no detail message. A detail message is a String that describes this particular exception.

Error Index

InternalError

public InternalError(String s)

Constructs an InternalError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.InterruptedIOException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.InterruptedIOException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.InterruptedIOException
```

```
public class InterruptedIOException
    extends Exception
```

An exception indicated that some thread has interrupted this thread.

See Also:

[interrupt](#), [interrupted](#)

Error Index

Error Index

[InterruptedIOException\(\)](#)

Constructs an InterruptedIOException with no detail message.

Error Index

[InterruptedIOException\(String\)](#)

Constructs an InterruptedIOException with the specified detail message.

Error Index

Error Index

[InterruptedIOException](#)

```
public InterruptedIOException()
```

Constructs an InterruptedException with no detail message. A detail message is a String that describes this particular exception.

Error Index InterruptedException

```
public InterruptedException(String s)
```

Constructs an InterruptedException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.LinkageError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.LinkageError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
```

```
public class LinkageError
    extends Error
```

LinkageError and its subclasses indicate that a class has some dependency on another class; however the latter class has incompatibly changed after the compilation of the former class.

Error Index

Error Index

[LinkageError\(\)](#)

Constructs a LinkageError with no detail message.

Error Index

[LinkageError\(String\)](#)

Constructs a LinkageError with the specified detail message.

Error Index

Error Index

LinkageError

```
public LinkageError()
```

Constructs a LinkageError with no detail message. A detail message is a String that describes this particular exception.

Error Index

LinkageError

public LinkageError(String s)

Constructs a LinkageError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Long

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Long

```
java.lang.Object
|
+----java.lang.Number
      |
      +----java.lang.Long
```

```
public final class Long
extends Number
```

The Long class provides an object wrapper for Long data values and serves as a place for long-oriented operations. A wrapper is useful because most of Java's utility classes require the use of objects. Since longs are not objects in Java, they need to be "wrapped" in a Long instance.

Error Index

Error Index

MAX_VALUE

The maximum value a Long can have.

Error Index

MIN_VALUE

The minimum value a Long can have.

Error Index

Error Index

Long(long)

Constructs a Long object initialized to the specified value.

Error Index

Long(String)

Constructs a Long object initialized to the value specified by the String parameter.

Error Index

Error Index

doubleValue()

Returns the value of this Long as a double.

Error Index

equals()(Object)

Compares this object against the specified object.

Error Index

floatValue()

Returns the value of this Long as a float.

Error Index

getLong()(String)

Get a Long property.

Error Index

getLong()(String, long)

Get a Long property.

Error Index

getLong()(String, Long)

Get a Long property.

Error Index

hashCode()

Computes a hashCode for this Long.

Error Index

intValue()

Returns the value of this Long as an int.

Error Index

longValue()

Returns the value of this Long as a long.

Error Index

parseLong()(String, int)

Assuming the specified String represents a long, returns that long's value.

Error Index

parseLong()(String)

Assuming the specified String represents a long, return that long's value.

Error Index

toString()(long, int)

Returns a new String object representing the specified long in the specified radix.

Error Index

toString()(long)

Returns a new String object representing the specified integer.

Error Index

toString()

Returns a String object representing this Long's value.

Error Index

valueOf(String, int)

Assuming the specified String represents a long, returns a new Long object initialized to that value.

Error Index

valueOf(String)

Assuming the specified String represents a long, returns a new Long object initialized to that value.

Error Index

Error Index

MIN_VALUE

public final static long MIN_VALUE

The minimum value a Long can have. The lowest minimum value that a Long can have is 0x8000000000000000.

Error Index

MAX_VALUE

public final static long MAX_VALUE

The maximum value a Long can have. The largest maximum value that a Long can have is 0x7fffffffffffffff.

Error Index

Error Index

Long

public Long(long value)

Constructs a Long object initialized to the specified value.

Parameters:

value - the initial value of the Long

Error Index

Long

public Long(String s) throws NumberFormatException

Constructs a Long object initialized to the value specified by the String parameter. The radix is assumed to be 10.

Parameters:

s - the String to be converted to a Long

Throws: NumberFormatException

If the String does not contain a parsable long.

Error Index

Error Index toString

```
public static String toString(long i,  
                               int radix)
```

Returns a new String object representing the specified long in the specified radix.

Parameters:

i - the long to be converted

radix - the radix

See Also:

MIN_RADIX, MAX_RADIX

Error Index toString

```
public static String toString(long i)
```

Returns a new String object representing the specified integer. The radix is assumed to be 10.

Parameters:

i - the long to be converted

Error Index parseLong

```
public static long parseLong(String s,  
                             int radix) throws NumberFormatException
```

Assuming the specified String represents a long, returns that long's value. Throws an exception if the String cannot be parsed as a long.

Parameters:

s - the String containing the integer

radix - the radix to be used

Throws: NumberFormatException

If the String does not contain a parsable integer.

Error Index `parseLong`

public static long parseLong(String s) throws NumberFormatException

Assuming the specified String represents a long, return that long's value. Throws an exception if the String cannot be parsed as a long. The radix is assumed to be 10.

Parameters:

s - the String containing the long

Throws: NumberFormatException

If the string does not contain a parsable long.

Error Index `valueOf`

public static Long valueOf(String s,
int radix) throws NumberFormatException

Assuming the specified String represents a long, returns a new Long object initialized to that value. Throws an exception if the String cannot be parsed as a long.

Parameters:

s - the String containing the long.

radix - the radix to be used

Throws: NumberFormatException

If the String does not contain a parsable long.

Error Index `valueOf`

public static Long valueOf(String s) throws NumberFormatException

Assuming the specified String represents a long, returns a new Long object initialized to that value. Throws an exception if the String cannot be parsed as a long. The radix is assumed to be 10.

Parameters:

s - the String containing the long

Throws: NumberFormatException

If the String does not contain a parsable long.

Error Index intValue

```
public int intValue()
```

Returns the value of this Long as an int.

Overrides:

intValue in class Number

Error Index longValue

```
public long longValue()
```

Returns the value of this Long as a long.

Overrides:

longValue in class Number

Error Index floatValue

```
public float floatValue()
```

Returns the value of this Long as a float.

Overrides:

floatValue in class Number

Error Index doubleValue

```
public double doubleValue()
```

Returns the value of this Long as a double.

Overrides:

doubleValue in class Number

Error Index toString

```
public String toString()
```

Returns a String object representing this Long's value.

Overrides:

toString in class Object

Error Index hashCode

```
public int hashCode()
```

Computes a hashCode for this Long.

Overrides:

hashCode in class Object

Error Index equals

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

Compares this object against the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index getLong

```
public static Long getLong(String nm)
```

Get a Long property. If the property does not exist, it will return 0.

Parameters:

nm - the property name

Error Index getLong

```
public static Long getLong(String nm,  
                             long val)
```

Get a Long property. If the property does not exist, it will return val. Deals with Hexadecimal and octal numbers.

Parameters:

nm - the String name

val - the Long value

Error Index `getLong`

```
public static Long getLong(String nm,  
                           Long val)
```

Get a Long property. If the property does not exist, it will return val. Deals with Hexadecimal and octal numbers.

Parameters:

nm - the property name

val - the Long value

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Math

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Math

[java.lang.Object](#)

|

+----java.lang.Math

public final class **Math**
extends [Object](#)

The standard Math library. For the methods in this Class, error handling for out-of-range or immeasurable results are platform dependent. This class cannot be subclassed or instantiated because all methods and variables are static.

Error Index

Error Index

E

The float representation of the value E.

Error Index

PI

The float representation of the value Pi.

Error Index

Error Index

IEEEremainder(double, double)

Returns the remainder of f1 divided by f2 as defined by IEEE 754.

Error Index

abs(int)

Returns the absolute integer value of a.

Error Index

abs(long)

Returns the absolute long value of a.

Error Index

abs(float)

Returns the absolute float value of a.

Error Index

abs(double)

Returns the absolute double value of a.

Error Index

acos(double)

Returns the arc cosine of a, in the range of 0.0 through Pi.

Error Index

asin(double)

Returns the arc sine of a, in the range of -Pi/2 through Pi/2.

Error Index

atan(double)

Returns the arc tangent of a, in the range of -Pi/2 through Pi/2.

Error Index

atan2(double, double)

Converts rectangular coordinates (a, b) to polar (r, theta).

Error Index

ceil(double)

Returns the "ceiling" or smallest whole number greater than or equal to a.

Error Index

cos(double)

Returns the trigonometric cosine of an angle.

Error Index

exp(double)

Returns the exponential number e(2.718...) raised to the power of a.

Error Index

floor(double)

Returns the "floor" or largest whole number less than or equal to a.

Error Index

log(double)

Returns the natural logarithm (base e) of a.

Error Index

max(int, int)

Takes two int values, a and b, and returns the greater number of the two.

Error Index

max(long, long)

Takes two long values, a and b, and returns the greater number of the two.

Error Index

max(float, float)

Takes two float values, a and b, and returns the greater number of the two.

Error Index

max(double, double)

Takes two double values, a and b, and returns the greater number of the two.

Error Index

min(int, int)

Takes two integer values, a and b, and returns the smallest number of the two.

Error Index

min(long, long)

Takes two long values, a and b, and returns the smallest number of the two.

Error Index

min(float, float)

Takes two float values, a and b, and returns the smallest number of the two.

Error Index

min(double, double)

Takes two double values, a and b, and returns the smallest number of the two.

Error Index

pow(double, double)

Returns the number a raised to the power of b.

Error Index

random()

Generates a random number between 0.0 and 1.0.

Error Index

rint(double)

Converts a double value into an integral value in double format.

Error Index

round(float)

Rounds off a float value by first adding 0.5 to it and then returning the largest integer that is less than or equal to this new value.

Error Index

round(double)

Rounds off a double value by first adding 0.5 to it and then returning the largest integer that is less than or equal to this new value.

Error Index

sin(double)

Returns the trigonometric sine of an angle.

Error Index

sqrt(double)

Returns the square root of a.

Error Index

tan(double)

Returns the trigonometric tangent of an angle.

Error Index

Error Index_E

public final static double E

The float representation of the value E. E is equivalent to 2.7182818284590452354f in Java.

Error Index_{PI}

```
public final static double PI
```

The float representation of the value Pi. Pi is equivalent to 3.14159265358979323846f in Java.

Error Index

Error Index_{sin}

```
public static double sin(double a)
```

Returns the trigonometric sine of an angle.

Parameters:

a - an assigned angle that is measured in radians

Error Index_{cos}

```
public static double cos(double a)
```

Returns the trigonometric cosine of an angle.

Parameters:

a - an assigned angle that is measured in radians

Error Index_{tan}

```
public static double tan(double a)
```

Returns the trigonometric tangent of an angle.

Parameters:

a - an assigned angle that is measured in radians

Error Index asin

public static double asin(double a)

Returns the arc sine of a, in the range of $-\pi/2$ through $\pi/2$.

Parameters:

a - (-1.0)

Error Index acos

public static double acos(double a)

Returns the arc cosine of a, in the range of 0.0 through π .

Parameters:

a - (-1.0)

Error Index atan

public static double atan(double a)

Returns the arc tangent of a, in the range of $-\pi/2$ through $\pi/2$.

Parameters:

a - an assigned value

Returns:

the arc tangent of a.

Error Index exp

public static double exp(double a)

Returns the exponential number $e(2.718\dots)$ raised to the power of a.

Parameters:

a - an assigned value

Error Index log

public static double log(double a) throws ArithmeticException

Returns the natural logarithm (base e) of a.

Parameters:

a - a is a number greater than 0.0

Throws: ArithmeticException

If a is less than 0.0 .

Error Index sqrt

public static double sqrt(double a) throws ArithmeticException

Returns the square root of a.

Parameters:

a - a is a number greater than or equal to 0.0

Throws: ArithmeticException

If a is a value less than 0.0 .

Error Index IEEERemainder

public static double IEEERemainder(double f1,
double f2)

Returns the remainder of f1 divided by f2 as defined by IEEE 754.

Parameters:

f1 - the dividend

f2 - the divisor

Error Index ceil

public static double ceil(double a)

Returns the "ceiling" or smallest whole number greater than or equal to a.

Parameters:

a - an assigned value

Error Index floor

public static double floor(double a)

Returns the "floor" or largest whole number less than or equal to a.

Parameters:

a - an assigned value

Error Index rint

```
public static double rint(double a)
```

Converts a double value into an integral value in double format.

Parameters:

a - an assigned double value

Error Index atan2

```
public static double atan2(double a,  
double b)
```

Converts rectangular coordinates (a, b) to polar (r, theta). This method computes the phase theta by computing an arc tangent of b/a in the range of -Pi to Pi.

Parameters:

a - an assigned value

b - an assigned value

Returns:

the polar coordinates (r, theta).

Error Index pow

```
public static double pow(double a,  
double b) throws ArithmeticException
```

Returns the number a raised to the power of b. If (a == 0.0), then b must be greater than 0.0; otherwise you will throw an exception. An exception will also occur if (a

Parameters:

a - an assigned value with the exceptions: (a == 0.0) -> (b > 0.0) & (a (b == a whole number)

b - an assigned value with the exceptions: (a == 0.0) -> (b > 0.0) & (a (b == a whole number)

Throws: ArithmeticException

If (a == 0.0) and (b **Throws:** ArithmeticException

If (a

Error Index round

```
public static int round(float a)
```

Rounds off a float value by first adding 0.5 to it and then returning the largest integer that is less than or equal to this new value.

Parameters:

a - the value to be rounded off

Error Index round

```
public static long round(double a)
```

Rounds off a double value by first adding 0.5 to it and then returning the largest integer that is less than or equal to this new value.

Parameters:

a - the value to be rounded off

Error Index random

```
public static synchronized double random()
```

Generates a random number between 0.0 and 1.0.

Random number generators are often referred to as pseudorandom number generators because the numbers produced tend to repeat themselves after a period of time.

Returns:

a pseudorandom double between 0.0 and 1.0.

Error Index abs

```
public static int abs(int a)
```

Returns the absolute integer value of a.

Parameters:

a - an assigned integer value

Error Index abs

```
public static long abs(long a)
```

Returns the absolute long value of a.

Parameters:

a - an assigned long value.

Error Index_{abs}

```
public static float abs(float a)
```

Returns the absolute float value of a.

Parameters:

a - an assigned float value

Error Index_{abs}

```
public static double abs(double a)
```

Returns the absolute double value of a.

Parameters:

a - an assigned double value

Error Index_{max}

```
public static int max(int a,  
int b)
```

Takes two int values, a and b, and returns the greater number of the two.

Parameters:

a - an integer value to be compared

b - an integer value to be compared

Error Index_{max}

```
public static long max(long a,  
long b)
```

Takes two long values, a and b, and returns the greater number of the two.

Parameters:

a - a long value to be compared

b - a long value to be compared

Error Index _{max}

```
public static float max(float a,  
                        float b)
```

Takes two float values, a and b, and returns the greater number of the two.

Parameters:

a - a float value to be compared

b - a float value to be compared

Error Index _{max}

```
public static double max(double a,  
                        double b)
```

Takes two double values, a and b, and returns the greater number of the two.

Parameters:

a - a double value to be compared

b - a double value to be compared

Error Index _{min}

```
public static int min(int a,  
                     int b)
```

Takes two integer values, a and b, and returns the smallest number of the two.

Parameters:

a - an integer value to be compared

b - an integer value to be compared

Error Index _{min}


```
public static long min(long a,  
                       long b)
```

Takes two long values, a and b, and returns the smallest number of the two.

Parameters:

- a - a long value to be compared
- b - a long value to be compared

Error Index_{min}

```
public static float min(float a,  
                       float b)
```

Takes two float values, a and b, and returns the smallest number of the two.

Parameters:

- a - a float value to be compared
- b - a float value to be compared

Error Index_{min}

```
public static double min(double a,  
                        double b)
```

Takes two double values, a and b, and returns the smallest number of the two.

Parameters:

- a - a double value to be compared
- b - a double value to be compared

Class

java.lang.NegativeArraySizeException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NegativeArraySizeException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.NegativeArraySizeException
```

```
public class NegativeArraySizeException
    extends RuntimeException
```

Signals that an attempt has been made to create an array with negative size.

Error Index

Error Index

[**NegativeArraySizeException\(\)**](#)

Constructs a NegativeArraySizeException with no detail message.

Error Index

[**NegativeArraySizeException\(String\)**](#)

Constructs a NegativeArraySizeException with the specified detail message.

Error Index

Error Index

[**NegativeArraySizeException**](#)

```
public NegativeArraySizeException()
```

Constructs a `NegativeArraySizeException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index

NegativeArraySizeException

```
public NegativeArraySizeException(String s)
```

Constructs a `NegativeArraySizeException` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoClassDefFoundError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoClassDefFoundError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
                  |
                  +----java.lang.NoClassDefFoundError
```

```
public class NoClassDefFoundError
    extends LinkageError
```

Signals that a class could not be found.

Error Index

Error Index

[NoClassDefFoundError\(\)](#)

Constructs a NoClassDefFoundError with no detail message.

Error Index

[NoClassDefFoundError\(String\)](#)

Constructs a NoClassDefFoundError with the specified detail message.

Error Index

Error Index

NoClassDefFoundError

```
public NoClassDefFoundError()
```

Constructs a NoClassDefFoundError with no detail message. A detail message is a String that describes this particular exception.

Error Index

NoClassDefFoundError

public NoClassDefFoundError(String s)

Constructs a NoClassDefFoundError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoSuchFieldError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoSuchFieldError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.LinkageError
|
+----java.lang.IncompatibleClassChangeError
|
+----java.lang.NoSuchFieldError
```

```
public class NoSuchFieldError
extends IncompatibleClassChangeError
```

Signals that a particular field could not be found.

Error Index

Error Index

[NoSuchFieldError\(\)](#)

Constructs a NoSuchFieldException without a detail message.

Error Index

[NoSuchFieldError\(String\)](#)

Constructs a NoSuchFieldException with a detail message.

Error Index

Error Index

NoSuchFieldError

```
public NoSuchFieldError()
```

Constructs a NoSuchFieldException without a detail message. A detail message is a String that describes this particular exception.

Error Index

NoSuchFieldError

```
public NoSuchFieldError(String s)
```

Constructs a NoSuchFieldException with a detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoSuchMethodError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoSuchMethodError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
                  |
                  +----java.lang.IncompatibleClassChangeError
                        |
                        +----java.lang.NoSuchMethodError
```

```
public class NoSuchMethodError
    extends IncompatibleClassChangeError
```

Signals that a particular method could not be found.

Error Index

Error Index

Error Index

NoSuchMethodError()

NoSuchMethodError(String)

Constructs a NoSuchMethodException with a detail message.

Error Index

Error Index

NoSuchMethodError

```
public NoSuchMethodError()
```


Error Index

NoSuchMethodError

public NoSuchMethodError(String s)

Constructs a NoSuchMethodException with a detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoSuchMethodException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NoSuchMethodException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.NoSuchMethodException
```

```
public class NoSuchMethodException
    extends Exception
```

Signals that a particular method could not be found.

Error Index

Error Index

NoSuchMethodException()

Constructs a NoSuchMethodException without a detail message.

Error Index

NoSuchMethodException(String)

Constructs a NoSuchMethodException with a detail message.

Error Index

Error Index

NoSuchMethodException

```
public NoSuchMethodException()
```

Constructs a NoSuchMethodException without a detail message. A detail message is a String that describes this particular exception.

Error Index

NoSuchMethodException

public NoSuchMethodException(String s)

Constructs a NoSuchMethodException with a detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NullPointerException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NullPointerException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.RuntimeException
|
+----java.lang.NullPointerException
```

```
public class NullPointerException
extends RuntimeException
```

Signals the illegal use of a null pointer.

Error Index

Error Index

[**NullPointerException\(\)**](#)

Constructs a NullPointerException with no detail message.

Error Index

[**NullPointerException\(String\)**](#)

Constructs a NullPointerException with the specified detail message.

Error Index

Error Index

NullPointerException

```
public NullPointerException()
```

Constructs a NullPointerException with no detail message. A detail message is a String that describes this particular exception.

Error Index

NullPointerException

```
public NullPointerException(String s)
```

Constructs a NullPointerException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Number

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Number

[java.lang.Object](#)

|

+----java.lang.Number

```
public class Number
    extends Object
```

Number is an abstract superclass for numeric scalar types. Integer, Long, Float and Double are subclasses of Number that bind to a particular numeric representation.

See Also:

[Integer](#), [Long](#), [Float](#), [Double](#)

Error Index

Error Index

[Number\(\)](#)

Error Index

Error Index

[doubleValue\(\)](#)

Returns the value of the number as a double.

Error Index

[floatValue\(\)](#)

Returns the value of the number as a float.

Error Index

[intValue\(\)](#)

Returns the value of the number as an int.

Error Index

[longValue\(\)](#)

Returns the value of the number as a long.

Error Index

Error Index **Number**

```
public Number()
```

Error Index

Error Index **intValue**

```
public abstract int intValue()
```

Returns the value of the number as an int. This may involve rounding if the number is not already an integer.

Error Index **longValue**

```
public abstract long longValue()
```

Returns the value of the number as a long. This may involve rounding if the number is not already a long.

Error Index **floatValue**

```
public abstract float floatValue()
```

Returns the value of the number as a float. This may involve rounding if the number is not already a float.

Error Index **doubleValue**

```
public abstract double doubleValue()
```

Returns the value of the number as a double. This may involve rounding if the number is not already a double.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NumberFormatException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.NumberFormatException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IllegalArgumentException
                        |
                        +----java.lang.NumberFormatException
```

```
public class NumberFormatException
    extends IllegalArgumentException
```

Signals that an invalid number format has occurred.

See Also:

[toString](#)

Error Index

Error Index

NumberFormatException()

Constructs a NumberFormatException with no detail message.

Error Index

NumberFormatException(String)

Constructs a NumberFormatException with the specified detail message.

Error Index

Error Index

NumberFormatException

public NumberFormatException()

Constructs a NumberFormatException with no detail message. A detail message is a String that describes this particular exception.

Error Index

NumberFormatException

public NumberFormatException(String s)

Constructs a NumberFormatException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Object

java.lang.Object

public class **Object**

The root of the Class hierarchy. Every Class in the system has Object as its ultimate parent. Every variable and method defined here is available in every Object.

See Also:

[Class](#)

Error Index

Error Index

Object()

Error Index

Error Index

clone()

Creates a clone of the object.

Error Index

equals()(Object)

Compares two Objects for equality.

Error Index

finalize()

Code to perform when this object is garbage collected.

Error Index

getClass()

Returns the Class of this Object.

Error Index

hashCode()

Returns a hashcode for this Object.

Error Index

notify()

Notifies a single waiting thread on a change in condition of another thread.

Error Index

notifyAll()

Notifies all of the threads waiting for a condition to change.

Error Index

toString()

Returns a String that represents the value of this Object.

Error Index

wait(long)

Causes a thread to wait until it is notified or the specified timeout expires.

Error Index

wait(long, int)

More accurate wait.

Error Index

wait()

Causes a thread to wait forever until it is notified.

Error Index

Error Index

Object

public Object()

Error Index

Error Index

getClass

public final Class getClass()

Returns the Class of this Object. Java has a runtime representation for classes- a descriptor of type Class- which the method getClass() returns for any Object.

Error Index

hashCode

public int hashCode()

Returns a hashcode for this Object. Each Object in the Java system has a hashcode. The hashcode

is a number that is usually different for different Objects. It is used when storing Objects in hashtables. Note: hashcodes can be negative as well as positive.

See Also:

Hashtable

Error Index equals

public boolean equals(Object obj)

Compares two Objects for equality. Returns a boolean that indicates whether this Object is equivalent to the specified Object. This method is used when an Object is stored in a hashtable.

Parameters:

obj - the Object to compare with

Returns:

true if these Objects are equal; false otherwise.

See Also:

Hashtable

Error Index clone

protected Object clone() throws CloneNotSupportedException

Creates a clone of the object. A new instance is allocated and a bitwise clone of the current object is place in the new object.

Returns:

a clone of this Object.

Throws: OutOfMemoryError

If there is not enough memory.

Throws: CloneNotSupportedException

Object explicitly does not want to be cloned, or it does not support the Cloneable interface.

Error Index toString

public String toString()

Returns a String that represents the value of this Object. It is recommended that all subclasses override this method.

Error Index notify

public final void notify()

Notifies a single waiting thread on a change in condition of another thread. The thread effecting the change notifies the waiting thread using notify(). Threads that want to wait for a condition to change before proceeding can call wait().

The method notify() can only be called from within a synchronized method.

Throws: IllegalMonitorStateException

If the current thread is not the owner of the Object's monitor.

See Also:

wait, notifyAll

Error Index notifyAll

public final void notifyAll()

Notifies all of the threads waiting for a condition to change. Threads that are waiting are generally waiting for another thread to change some condition. Thus, the thread effecting a change that more than one thread is waiting for notifies all the waiting threads using the method notifyAll(). Threads that want to wait for a condition to change before proceeding can call wait().

The method notifyAll() can only be called from within a synchronized method.

Throws: IllegalMonitorStateException

If the current thread is not the owner of the Object's monitor.

See Also:

wait, notify

Error Index wait

public final void wait(long timeout) throws InterruptedException

Causes a thread to wait until it is notified or the specified timeout expires.

The method wait() can only be called from within a synchronized method.

Parameters:

timeout - the maximum time to wait in milliseconds

Throws: IllegalMonitorStateException

If the current thread is not the owner of the Object's monitor.

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index wait

public final void wait(long timeout,
int nanos) throws InterruptedException

More accurate wait. *The method wait() can only be called from within a synchronized method.*

Parameters:

timeout - the maximum time to wait in milliseconds

nano - additional time, in nanoseconds range 0-999999

Throws: IllegalMonitorStateException

If the current thread is not the owner of the Object's monitor.

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index wait

public final void wait() throws InterruptedException

Causes a thread to wait forever until it is notified.

The method wait() can only be called from within a synchronized method

Throws: IllegalMonitorStateException

If the current thread is not the owner of the Object's monitor.

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index finalize

protected void finalize() throws Throwable

Code to perform when this object is garbage collected. The default is that nothing needs to be performed. Any exception thrown by a finalize method causes the finalization to halt. But otherwise, it is ignored.

Class java.lang.OutOfMemoryError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.OutOfMemoryError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.VirtualMachineError
|
+----java.lang.OutOfMemoryError
```

```
public class OutOfMemoryError
extends VirtualMachineError
```

Signals that you are out of memory.

Error Index

Error Index

[**OutOfMemoryError\(\)**](#)

Constructs an OutOfMemoryError with no detail message.

Error Index

[**OutOfMemoryError\(String\)**](#)

Constructs an OutOfMemoryError with the specified detail message.

Error Index

Error Index

OutOfMemoryError

```
public OutOfMemoryError()
```

Constructs an OutOfMemoryError with no detail message. A detail message is a String that describes this particular exception.

Error Index

OutOfMemoryError

public OutOfMemoryError(String s)

Constructs an OutOfMemoryError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Process

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Process

[java.lang.Object](#)

|
+----java.lang.Process

public class **Process**
extends [Object](#)

An instance of class Process is returned by variants of the exec () method in class System. From the Process instance, it is possible to: get the stdin and/or stdout of the subprocess, kill the subprocess, wait for it to terminate, and to retrieve the final exit value of the process.

Dropping the last reference to a Process instance does not kill the subprocess. There is no requirement that the subprocess execute asynchronously with the existing Java process.

Error Index

Error Index

[Process\(\)](#)

Error Index

Error Index

[destroy\(\)](#)

Kills the subprocess.

Error Index

[exitValue\(\)](#)

Returns the exit value for the subprocess.

Error Index

[getErrorStream\(\)](#)

Returns the an InputStream connected to the error stream of the child process.

Error Index

[getInputStream\(\)](#)

Returns a Stream connected to the output of the child process.

Error Index

getOutputStream()

Returns a Stream connected to the input of the child process.

Error Index

waitFor()

Waits for the subprocess to complete.

Error Index

Error Index

Process

public Process()

Error Index

Error Index

getOutputStream

public abstract OutputStream getOutputStream()

Returns a Stream connected to the input of the child process. This stream is traditionally buffered.

Error Index

getInputStream

public abstract InputStream getInputStream()

Returns a Stream connected to the output of the child process. This stream is traditionally buffered.

Error Index

getErrorStream

public abstract InputStream getErrorStream()

Returns the an InputStream connected to the error stream of the child process. This stream is traditionally unbuffered.

Error Index

waitFor

public abstract int waitFor() throws InterruptedException

Waits for the subprocess to complete. If the subprocess has already terminated, the exit value is simply returned. If the subprocess has not yet terminated the calling thread will be blocked until the subprocess exits.

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index `exitValue`

public abstract int exitValue()

Returns the exit value for the subprocess.

Throws: IllegalThreadStateException

If the subprocess has not yet terminated.

Error Index `destroy`

public abstract void destroy()

Kills the subprocess.

Interface java.lang.Runnable

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.lang.Runnable

public interface **Runnable**
extends [Object](#)

This interface is designed to provide a common protocol for Objects that wish to execute code while they are active. For example, Runnable is implemented by class Thread. Being active simply means that a thread has been started and has not yet been stopped.

In addition, Runnable provides the means for a class to be active while not subclassing Thread. A class that implements Runnable can run without subclassing Thread by instantiating a Thread instance and passing itself in as the target. In most cases, the Runnable interface should be used if you are only planning to override the run() method and no other Thread methods. This is important because classes should not be subclassed unless the programmer intends on modifying or enhancing the fundamental behavior of the class.

See Also:

[Thread](#)

Error Index

Error Index

[run\(\)](#)

The method that is executed when a Runnable object is activated.

Error Index

Error Index

[run](#)

public abstract void run()

The method that is executed when a Runnable object is activated. The run() method is the "soul" of a Thread. It is in this method that all of the action of a Thread takes place.

See Also:

[run](#)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Runtime

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Runtime

[java.lang.Object](#)

|

+----java.lang.Runtime

```
public class Runtime
    extends Object
```

Error Index

Error Index

exec(String)

Executes the system command specified in the parameter.

Error Index

exec(String, String[])

Executes the system command specified in the parameter.

Error Index

exec(String[])

Executes the system command specified by cmdarray[0] with arguments specified by the strings in the rest of the array.

Error Index

exec(String[], String[])

Executes the system command specified by cmdarray[0] with arguments specified by the strings in the rest of the array.

Error Index

exit(int)

Exits the virtual machine with an exit code.

Error Index

freeMemory()

Returns the number of free bytes in system memory.

Error Index

gc()

Runs the garbage collector.

Error Index

Localize an input stream.

getLocalizedInputStream(InputStream)

Error Index

Localize an output stream.

getLocalizedOutputStream(OutputStream)

Error Index

Returns the runtime.

getRuntime()

Error Index

Loads a dynamic library, given a complete path name.

load(String)

Error Index

Loads a dynamic library with the specified library name.

loadLibrary(String)

Error Index

Runs the finalization methods of any objects pending finalization.

runFinalization()

Error Index

Returns the total number of bytes in system memory.

totalMemory()

Error Index

Enables/Disables tracing of instructions.

traceInstructions(boolean)

Error Index

Enables/Disables tracing of method calls.

traceMethodCalls(boolean)

Error Index

Error Index

getRuntime

public static Runtime getRuntime()

Returns the runtime.

Error Index

exit

public void exit(int status)

Exits the virtual machine with an exit code. This method does not return, use with caution.

Parameters:

status - exit status, 0 if successful, other values indicate various error types.

Error Index exec

public Process exec(String command) throws IOException

Executes the system command specified in the parameter. Returns a Process which has methods for obtaining the stdin, stdout, and stderr of the subprocess. This method fails if executed by untrusted code.

Parameters:

command - a specified system command

Returns:

an instance of class Process

Error Index exec

public Process exec(String command,
String envp[]) throws IOException

Executes the system command specified in the parameter. Returns a Process which has methods for obtaining the stdin, stdout, and stderr of the subprocess. This method fails if executed by untrusted code.

Parameters:

command - a specified system command

Returns:

an instance of class Process

Error Index exec

public Process exec(String cmdarray[]) throws IOException

Executes the system command specified by cmdarray[0] with arguments specified by the strings in the rest of the array. Returns a Process which has methods for obtaining the stdin, stdout, and stderr of the subprocess. This method fails if executed by untrusted code.

Parameters:

an - array containing the command to call and its arguments

envp - array containing environment in format name=value

Returns:

an instance of class Process

Error Index **exec**

```
public Process exec(String cmdarray[],  
    String envp[]) throws IOException
```

Executes the system command specified by cmdarray[0] with arguments specified by the strings in the rest of the array. Returns a Process which has methods for obtaining the stdin, stdout, and stderr of the subprocess. This method fails if executed by untrusted code.

Parameters:

an - array containing the command to call and its arguments

envp - array containing environment in format name=value

Returns:

an instance of class Process

Error Index **freeMemory**

```
public long freeMemory()
```

Returns the number of free bytes in system memory. This number is not always accurate because it is just an estimation of the available memory. More memory may be freed by calling System.gc() .

Error Index **totalMemory**

```
public long totalMemory()
```

Returns the total number of bytes in system memory.

Error Index **gc**

```
public void gc()
```

Runs the garbage collector.

Error Index **runFinalization**

```
public void runFinalization()
```

Runs the finalization methods of any objects pending finalization. Usually you will not need to call this method since finalization methods will be called asynchronously by the finalization thread. However, under some circumstances (like running out of a finalized resource) it can be useful to run finalization

methods synchronously.

Error Index `traceInstructions`

`public void traceInstructions(boolean on)`

Enables/Disables tracing of instructions.

Parameters:

on - start tracing if true

Error Index `traceMethodCalls`

`public void traceMethodCalls(boolean on)`

Enables/Disables tracing of method calls.

Parameters:

on - start tracing if true

Error Index `load`

`public synchronized void load(String filename)`

Loads a dynamic library, given a complete path name. If you use this from `java_g` it will automatically insert `"_g"` before the `".so"`. Example: `Runtime.getRuntime().load("/home/avh/lib/libX11.so");`

Parameters:

filename - the file to load

Throws: `UnsatisfiedLinkError`

If the file does not exist.

See Also:

`getRuntime`

Error Index `loadLibrary`

`public synchronized void loadLibrary(String libname)`

Loads a dynamic library with the specified library name. The call to `LoadLibrary()` should be made in the static initializer of the first class that is loaded. Linking in the same library more than once is ignored.

Parameters:

libname - the name of the library

Throws: [UnsatisfiedLinkError](#)
If the library does not exist.

Error Index

getLocalizedInputStream

```
public InputStream getLocalizedInputStream(InputStream in)
```

Localize an input stream. A localized input stream will automatically translate the input from the local format to UNICODE.

Error Index

getLocalizedOutputStream

```
public OutputStream getLocalizedOutputStream(OutputStream out)
```

Localize an output stream. A localized output stream will automatically translate the output from UNICODE to the local format.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.RuntimeException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.RuntimeException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.lang.RuntimeException
```

```
public class RuntimeException
extends Exception
```

An exception that can reasonably occur during the execution of a Java program by the Virtual machine.

Error Index

Error Index

RuntimeException()

Constructs a RuntimeException with no detail message.

Error Index

RuntimeException(String)

Constructs a RuntimeException with the specified detail message.

Error Index

Error Index

RuntimeException

```
public RuntimeException()
```

Constructs a RuntimeException with no detail message. A detail message is a String that describes this particular exception.

Error Index

RuntimeException

public RuntimeException(String s)

Constructs a RuntimeException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.SecurityException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.SecurityException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.SecurityException
```

```
public class SecurityException
    extends RuntimeException
```

Signals that a security exception has occurred.

Error Index

Error Index

[SecurityException\(\)](#)

Constructs a SecurityException with no detail message.

Error Index

[SecurityException\(String\)](#)

Constructs a SecurityException with the specified detail message.

Error Index

Error Index

SecurityException

```
public SecurityException()
```

Constructs a SecurityException with no detail message. A detail message is a String that describes this particular exception.

Error Index **SecurityException**

public SecurityException(String s)

Constructs a SecurityException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.SecurityManager

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.SecurityManager

[java.lang.Object](#)

|

+----java.lang.SecurityManager

public class **SecurityManager**
extends [Object](#)

An abstract class that can be subclassed to implement a security policy. It allows the inspection of the classloaders on the execution stack.

Error Index

Error Index

[inCheck](#)

Error Index

Error Index

[SecurityManager\(\)](#)

Constructs a new SecurityManager.

Error Index

Error Index

[checkAccept](#)(String, int)

Checks to see if a socket connection to the specified port on the specified host has been accepted.

Error Index

[checkAccess](#)(Thread)

Checks to see if the specified Thread is allowed to modify the Thread group.

Error Index

checkAccess(ThreadGroup)

Checks to see if the specified Thread group is allowed to modify this group.

Error Index

checkConnect(String, int)

Checks to see if a socket has connected to the specified port on the the specified host.

Error Index

checkConnect(String, int, Object)

Checks to see if the current execution context and the indicated execution context are both allowed to connect to the indicated host and port.

Error Index

checkCreateClassLoader()

Checks to see if the ClassLoader has been created.

Error Index

checkDelete(String)

Checks to see if a file with the specified system dependent file name can be deleted.

Error Index

checkExec(String)

Checks to see if the system command is executed by trusted code.

Error Index

checkExit(int)

Checks to see if the system has exited the virtual machine with an exit code.

Error Index

checkLink(String)

Checks to see if the specified linked library exists.

Error Index

checkListen(int)

Checks to see if a server socket is listening to the specified local port that it is bounded to.

Error Index

checkPackageAccess(String)

Checks to see if an applet can access a package.

Error Index

checkPackageDefinition(String)

Checks to see if an applet can define classes in a package.

Error Index

checkPropertiesAccess()

Checks to see who has access to the System properties.

Error Index

checkPropertyAccess(String)

Checks to see who has access to the System property named by *key*.

Error Index

checkPropertyAccess(String, String)

Checks to see who has access to the System property named by *key* and *def*.

Error Index

checkRead(FileDescriptor)

Checks to see if an input file with the specified file descriptor object gets created.

Error Index

checkRead(String)

Checks to see if an input file with the specified system dependent file name gets created.

Error Index

checkRead(String, Object)

Checks to see if the current context or the indicated context are both allowed to read the given file name.

Error Index

checkSetFactory()

Checks to see if an applet can set a networking-related object factory.

Error Index

checkTopLevelWindow(Object)

Checks to see if top-level windows can be created by the caller.

Error Index

checkWrite(FileDescriptor)

Checks to see if an output file with the specified file descriptor object gets created.

Error Index

checkWrite(String)

Checks to see if an output file with the specified system dependent file name gets created.

Error Index

classDepth(String)

Return the position of the stack frame containing the first occurrence of the named class.

Error Index

classLoaderDepth()

Error Index

currentClassLoader()

The current ClassLoader on the execution stack.

Error Index

getClassContext()

Gets the context of this Class.

Error Index

getInCheck()

Returns whether there is a security check in progress.

Error Index

getSecurityContext()

Returns an implementation-dependent Object which encapsulates enough information about the current execution environment to perform some of the security checks later.

Error Index

inClass(String)

Returns true if the specified String is in this Class.

Error Index

inClassLoader()

Returns a boolean indicating whether or not the current ClassLoader is equal to null.

Error Index

Error Index **inCheck**

protected boolean inCheck

Error Index

Error Index **SecurityManager**

protected SecurityManager()

Constructs a new SecurityManager.

Throws: SecurityException

If the security manager cannot be created.

Error Index

Error Index **getInCheck**

public boolean getInCheck()

Returns whether there is a security check in progress.

Error Index **getClassContext**

protected Class[] getClassContext()

Gets the context of this Class.

Error Index **currentClassLoader**

protected ClassLoader currentClassLoader()

The current ClassLoader on the execution stack.

Error Index

classDepth

protected int classDepth(String name)

Return the position of the stack frame containing the first occurrence of the named class.

Parameters:

name - classname of the class to search for

Error Index

classLoaderDepth

protected int classLoaderDepth()

Error Index

inClass

protected boolean inClass(String name)

Returns true if the specified String is in this Class.

Parameters:

name - the name of the class

Error Index

inClassLoader

protected boolean inClassLoader()

Returns a boolean indicating whether or not the current ClassLoader is equal to null.

Error Index

getSecurityContext

public Object getSecurityContext()

Returns an implementation-dependent Object which encapsulates enough information about the current execution environment to perform some of the security checks later.

Error Index

checkCreateClassLoader

```
public void checkCreateClassLoader()
```

Checks to see if the ClassLoader has been created.

Throws: SecurityException

If a security error has occurred.

Error Index checkAccess

```
public void checkAccess(Thread g)
```

Checks to see if the specified Thread is allowed to modify the Thread group.

Parameters:

g - the Thread to be checked

Throws: SecurityException

If the current Thread is not allowed to access this Thread group.

Error Index checkAccess

```
public void checkAccess(ThreadGroup g)
```

Checks to see if the specified Thread group is allowed to modify this group.

Parameters:

g - the Thread group to be checked

Throws: SecurityException

If the current Thread group is not allowed to access this Thread group.

Error Index checkExit

```
public void checkExit(int status)
```

Checks to see if the system has exited the virtual machine with an exit code.

Parameters:

status - exit status, 0 if successful, other values indicate various error types.

Throws: SecurityException

If a security error has occurred.

Error Index checkExec

```
public void checkExec(String cmd)
```

Checks to see if the system command is executed by trusted code.

Parameters:

cmd - the specified system command

Throws: SecurityException

If a security error has occurred.

Error Index checkLink

```
public void checkLink(String lib)
```

Checks to see if the specified linked library exists.

Parameters:

lib - the name of the library

Throws: SecurityException

If the library does not exist.

Error Index checkRead

```
public void checkRead(FileDescriptor fd)
```

Checks to see if an input file with the specified file descriptor object gets created.

Parameters:

fd - the system dependent file descriptor

Throws: SecurityException

If a security error has occurred.

Error Index checkRead

```
public void checkRead(String file)
```

Checks to see if an input file with the specified system dependent file name gets created.

Parameters:

file - the system dependent file name

Throws: SecurityException

If the file is not found.

Error Index

checkRead

```
public void checkRead(String file,  
                     Object context)
```

Checks to see if the current context or the indicated context are both allowed to read the given file name.

Parameters:

file - the system dependent file name

context - the alternate execution context which must also be checked

Throws: SecurityException

If the file is not found.

Error Index

checkWrite

```
public void checkWrite(FileDescriptor fd)
```

Checks to see if an output file with the specified file descriptor object gets created.

Parameters:

fd - the system dependent file descriptor

Throws: SecurityException

If a security error has occurred.

Error Index

checkWrite

```
public void checkWrite(String file)
```

Checks to see if an output file with the specified system dependent file name gets created.

Parameters:

file - the system dependent file name

Throws: SecurityException

If the file is not found.

Error Index

checkDelete

```
public void checkDelete(String file)
```


Checks to see if a file with the specified system dependent file name can be deleted.

Parameters:

file - the system dependent file name

Throws: SecurityException

If the file is not found.

Error Index checkConnect

```
public void checkConnect(String host,  
                        int port)
```

Checks to see if a socket has connected to the specified port on the the specified host.

Parameters:

host - the host name port to connect to

port - the protocol port to connect to

Throws: SecurityException

If a security error has occurred.

Error Index checkConnect

```
public void checkConnect(String host,  
                        int port,  
                        Object context)
```

Checks to see if the current execution context and the indicated execution context are both allowed to connect to the indicated host and port.

Error Index checkListen

```
public void checkListen(int port)
```

Checks to see if a server socket is listening to the specified local port that it is bounded to.

Parameters:

port - the protocol port to connect to

Throws: SecurityException

If a security error has occurred.

Error Index checkAccept

```
public void checkAccept(String host,  
                        int port)
```

Checks to see if a socket connection to the specified port on the specified host has been accepted.

Parameters:

host - the host name to connect to

port - the protocol port to connect to

Throws: SecurityException

If a security error has occurred.

Error Index

checkPropertiesAccess

```
public void checkPropertiesAccess()
```

Checks to see who has access to the System properties.

Throws: SecurityException

If a security error has occurred.

Error Index

checkPropertyAccess

```
public void checkPropertyAccess(String key)
```

Checks to see who has access to the System property named by *key*.

Parameters:

key - the System property that the caller wants to examine

Throws: SecurityException

If a security error has occurred.

Error Index

checkPropertyAccess

```
public void checkPropertyAccess(String key,  
                                String def)
```

Checks to see who has access to the System property named by *key* and *def*.

Parameters:

key - the System property that the caller wants to examine

def - default value to return if this property is not defined

Throws: SecurityException

If a security error has occurred.

Error Index

checkTopLevelWindow

```
public boolean checkTopLevelWindow(Object window)
```

Checks to see if top-level windows can be created by the caller. A return of false means that the window creation is allowed but the window should indicate some sort of visual warning. Returning true means the creation is allowed with no special restrictions. To disallow the creation entirely, this method should throw a SecurityException.

Parameters:

window - the new window that's being created.

Error Index

checkPackageAccess

```
public void checkPackageAccess(String pkg)
```

Checks to see if an applet can access a package.

Error Index

checkPackageDefinition

```
public void checkPackageDefinition(String pkg)
```

Checks to see if an applet can define classes in a package.

Error Index

checkSetFactory

```
public void checkSetFactory()
```

Checks to see if an applet can set a networking-related object factory.

Class java.lang.StackOverflowError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.StackOverflowError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.VirtualMachineError
|
+----java.lang.StackOverflowError
```

```
public class StackOverflowError
    extends VirtualMachineError
```

Signals that a stack overflow has occurred.

Error Index

Error Index

StackOverflowError()

Constructs a StackOverflowError with no detail message.

Error Index

StackOverflowError(String)

Constructs a StackOverflowError with the specified detail message.

Error Index

Error Index

StackOverflowError

```
public StackOverflowError()
```

Constructs a StackOverflowError with no detail message. A detail message is a String that describes this particular exception.

Error Index

StackOverflowError

public StackOverflowError(String s)

Constructs a StackOverflowError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.String

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.String

[java.lang.Object](#)

|
+----java.lang.String

public final class **String**
extends [Object](#)

A general class of objects to represent character Strings. Strings are constant, their values cannot be changed after creation. The compiler makes sure that each String constant actually results in a String object. Because String objects are immutable they can be shared. For example:

```
String str = "abc";
```

is equivalent to:

```
char data[] = {'a', 'b', 'c'};  
String str = new String(data);
```

Here are some more examples of how strings can be used:

```
System.out.println("abc");  
String cde = "cde";  
System.out.println("abc" + cde);  
String c = "abc".substring(2,3);  
String d = cde.substring(1, 2);
```

See Also:

[StringBuffer](#)

Error Index

Error Index

[String\(\)](#)

Constructs a new empty String.

Error Index

String(String)

Constructs a new String that is a copy of the specified String.

Error Index

String(char[])

Constructs a new String whose initial value is the specified array of characters.

Error Index

String(char[], int, int)

Constructs a new String whose initial value is the specified sub array of characters.

Error Index

String(byte[], int, int, int)

Constructs a new String whose initial value is the specified sub array of bytes.

Error Index

String(byte[], int)

Constructs a new String whose value is the specified array of bytes.

Error Index

String(StringBuffer)

Construct a new string whose value is the current contents of the given string buffer

Error Index

Error Index

charAt(int)

Returns the character at the specified index.

Error Index

compareTo(String)

Compares this String to another specified String.

Error Index

concat(String)

Concatenates the specified string to the end of this String.

Error Index

copyValueOf(char[], int, int)

Returns a String that is equivalent to the specified character array.

Error Index

copyValueOf(char[])

Returns a String that is equivalent to the specified character array.

Error Index

endsWith(String)

Determines whether the String ends with some suffix.

Error Index

equals(Object)

Compares this String to the specified object.

Error Index

equalsIgnoreCase(String)

Compares this String to another object.

Error Index

getBytes(int, int, byte[], int)

Copies characters from this String into the specified byte array.

Error Index

getChars(int, int, char[], int)

Copies characters from this String into the specified character array.

Error Index

hashCode()

Returns a hashcode for this String.

Error Index

indexOf(int)

Returns the index within this String of the first occurrence of the specified character.

Error Index

indexOf(int, int)

Returns the index within this String of the first occurrence of the specified character, starting the search at fromIndex.

Error Index

indexOf(String)

Returns the index within this String of the first occurrence of the specified substring.

Error Index

indexOf(String, int)

Returns the index within this String of the first occurrence of the specified substring.

Error Index

intern()

Returns a String that is equal to this String but which is guaranteed to be from the unique String pool.

Error Index

lastIndexOf(int)

Returns the index within this String of the last occurrence of the specified character.

Error Index

lastIndexOf(int, int)

Returns the index within this String of the last occurrence of the specified character.

Error Index

lastIndexOf(String)

Returns the index within this String of the last occurrence of the specified substring.

Error Index

lastIndexOf(String, int)

Returns the index within this String of the last occurrence of the specified substring.

Error Index

length()

Returns the length of the String.

Error Index

regionMatches(int, String, int, int)

Determines whether a region of this String matches the specified region of the specified String.

Error Index

regionMatches(boolean, int, String, int, int)

Determines whether a region of this String matches the specified region of the specified String.

Error Index

replace(char, char)

Converts this String by replacing all occurrences of oldChar with newChar.

Error Index

startsWith(String, int)

Determines whether this String starts with some prefix.

Error Index

startsWith(String)

Determines whether this String starts with some prefix.

Error Index

substring(int)

Returns the substring of this String.

Error Index

substring(int, int)

Returns the substring of a String.

Error Index

toCharArray()

Converts this String to a character array.

Error Index

toLowerCase()

Converts all of the characters in this String to lower case.

Error Index

toString()

Converts this String to a String.

Error Index

toUpperCase()

Converts all of the characters in this String to upper case.

Error Index

trim()

Trims leading and trailing whitespace from this String.

Error Index

valueOf(Object)

Returns a String that represents the String value of the object.

Error Index

valueOf(char[])

Returns a String that is equivalent to the specified character array.

Error Index

valueOf(char[], int, int)

Returns a String that is equivalent to the specified character array.

Error Index

valueOf(boolean)

Returns a String object that represents the state of the specified boolean.

Error Index

valueOf(char)

Returns a String object that contains a single character

Error Index

valueOf(int)

Returns a String object that represents the value of the specified integer.

Error Index

valueOf(long)

Returns a String object that represents the value of the specified long.

Error Index

valueOf(float)

Returns a String object that represents the value of the specified float.

Error Index

valueOf(double)

Returns a String object that represents the value of the specified double.

Error Index

Error Index

String

public String()

Constructs a new empty String.

Error Index

String

public String(String value)

Constructs a new String that is a copy of the specified String.

Parameters:

value - the initial value of the String

Error Index

String

public String(char value[])

Constructs a new String whose initial value is the specified array of characters.

Parameters:

value - the initial value of the String

Error Index String

```
public String(char value[],  
              int offset,  
              int count)
```

Constructs a new String whose initial value is the specified sub array of characters. The length of the new string will be count characters starting at offset within the specified character array.

Parameters:

value - the initial value of the String, an array of characters

offset - the offset into the value of the String

count - the length of the value of the String

Throws: StringIndexOutOfBoundsException

If the offset and count arguments are invalid.

Error Index String

```
public String(byte ascii[],  
              int hibyte,  
              int offset,  
              int count)
```

Constructs a new String whose initial value is the specified sub array of bytes. The high-byte of each character can be specified, it should usually be 0. The length of the new String will be count characters starting at offset within the specified character array.

Parameters:

ascii - the bytes that will be converted to characters

hibyte - the high byte of each Unicode character

offset - the offset into the ascii array

count - the length of the String

Throws: StringIndexOutOfBoundsException

If the offset and count arguments are invalid.

Error Index String

```
public String(byte ascii[],  
              int hibyte)
```

Constructs a new String whose value is the specified array of bytes. The byte array transformed into Unicode chars using hibyte as the upper byte of each character.

Parameters:

ascii - the byte that will be converted to characters

hibyte - the top 8 bits of each 16 bit Unicode character

Error Index String

```
public String(StringBuffer buffer)
```

Construct a new string whose value is the current contents of the given string buffer

Parameters:

buffer - the stringbuffer to be converted

Error Index

Error Index length

```
public int length()
```

Returns the length of the String. The length of the String is equal to the number of 16 bit Unicode characters in the String.

Error Index charAt

```
public char charAt(int index)
```

Returns the character at the specified index. An index ranges from 0 to length() - 1.

Parameters:

index - the index of the desired character

Throws: StringIndexOutOfBoundsException

If the index is not in the range 0 to length()-1.

Error Index getChars

```
public void getChars(int srcBegin,  
    int srcEnd,  
    char dst[],  
    int dstBegin)
```

Copies characters from this String into the specified character array. The characters of the specified substring (determined by srcBegin and srcEnd) are copied into the character array, starting at the

array's dstBegin location.

Parameters:

srcBegin - index of the first character in the string

srcEnd - end of the characters that are copied

dst - the destination array

dstBegin - the start offset in the destination array

Error IndexgetBytes

```
public void getBytes(int srcBegin,  
                    int srcEnd,  
                    byte dst[],  
                    int dstBegin)
```

Copies characters from this String into the specified byte array. Copies the characters of the specified substring (determined by srcBegin and srcEnd) into the byte array, starting at the array's dstBegin location.

Parameters:

srcBegin - index of the first character in the String

srcEnd - end of the characters that are copied

dst - the destination array

dstBegin - the start offset in the destination array

Error Indexequals

```
public boolean equals(Object anObject)
```

Compares this String to the specified object. Returns true if the object is equal to this String; that is, has the same length and the same characters in the same sequence.

Parameters:

anObject - the object to compare this String against

Returns:

true if the Strings are equal; false otherwise.

Overrides:

equals in class Object

Error IndexequalsIgnoreCase

```
public boolean equalsIgnoreCase(String anotherString)
```

Compares this String to another object. Returns true if the object is equal to this String; that is, has the same length and the same characters in the same sequence. Upper case characters are folded to

lower case before they are compared.

Parameters:

anotherString - the String to compare this String against

Returns:

true if the Strings are equal, ignoring case; false otherwise.

Error Index compareTo

```
public int compareTo(String anotherString)
```

Compares this String to another specified String. Returns an integer that is less than, equal to, or greater than zero. The integer's value depends on whether this String is less than, equal to, or greater than anotherString.

Parameters:

anotherString - the String to be compared

Error Index regionMatches

```
public boolean regionMatches(int toffset,  
                             String other,  
                             int ooffset,  
                             int len)
```

Determines whether a region of this String matches the specified region of the specified String.

Parameters:

toffset - where to start looking in this String

other - the other String

ooffset - where to start looking in the other String

len - the number of characters to compare

Returns:

true if the region matches with the other; false otherwise.

Error Index regionMatches

```
public boolean regionMatches(boolean ignoreCase,  
                             int toffset,  
                             String other,  
                             int ooffset,  
                             int len)
```

Determines whether a region of this String matches the specified region of the specified String. If the boolean ignoreCase is true, upper case characters are considered equivalent to lower case letters.

Parameters:

ignoreCase - if true, case is ignored
toffset - where to start looking in this String
other - the other String
ooffset - where to start looking in the other String
len - the number of characters to compare

Returns:

true if the region matches with the other; false otherwise.

Error Index **startsWith**

```
public boolean startsWith(String prefix,  
                        int toffset)
```

Determines whether this String starts with some prefix.

Parameters:

prefix - the prefix
toffset - where to begin looking in the the String

Returns:

true if the String starts with the specified prefix; false otherwise.

Error Index **startsWith**

```
public boolean startsWith(String prefix)
```

Determines whether this String starts with some prefix.

Parameters:

prefix - the prefix

Returns:

true if the String starts with the specified prefix; false otherwise.

Error Index **endsWith**

```
public boolean endsWith(String suffix)
```

Determines whether the String ends with some suffix.

Parameters:

suffix - the suffix

Returns:

true if the String ends with the specified suffix; false otherwise.

Error Index hashCode

```
public int hashCode()
```

Returns a hashcode for this String. This is a large number composed of the character values in the String.

Overrides:

hashCode in class Object

Error Index indexOf

```
public int indexOf(int ch)
```

Returns the index within this String of the first occurrence of the specified character. This method returns -1 if the index is not found.

Parameters:

ch - the character to search for

Error Index indexOf

```
public int indexOf(int ch,  
                  int fromIndex)
```

Returns the index within this String of the first occurrence of the specified character, starting the search at fromIndex. This method returns -1 if the index is not found.

Parameters:

ch - the character to search for

fromIndex - the index to start the search from

Error Index lastIndexOf

```
public int lastIndexOf(int ch)
```

Returns the index within this String of the last occurrence of the specified character. The String is searched backwards starting at the last character. This method returns -1 if the index is not found.

Parameters:

ch - the character to search for

Error Index

lastIndexOf

```
public int lastIndexOf(int ch,  
                      int fromIndex)
```

Returns the index within this String of the last occurrence of the specified character. The String is searched backwards starting at fromIndex. This method returns -1 if the index is not found.

Parameters:

ch - the character to search for
fromIndex - the index to start the search from

Error Index

indexOf

```
public int indexOf(String str)
```

Returns the index within this String of the first occurrence of the specified substring. This method returns -1 if the index is not found.

Parameters:

str - the substring to search for

Error Index

indexOf

```
public int indexOf(String str,  
                  int fromIndex)
```

Returns the index within this String of the first occurrence of the specified substring. The search is started at fromIndex. This method returns -1 if the index is not found.

Parameters:

str - the substring to search for
fromIndex - the index to start the search from

Error Index

lastIndexOf

```
public int lastIndexOf(String str)
```

Returns the index within this String of the last occurrence of the specified substring. The String is searched backwards. This method returns -1 if the index is not found.

Parameters:

str - the substring to search for

Error Index **lastIndexOf**

```
public int lastIndexOf(String str,  
int fromIndex)
```

Returns the index within this String of the last occurrence of the specified substring. The String is searched backwards starting at fromIndex. This method returns -1 if the index is not found.

Parameters:

str - the substring to search for

fromIndex - the index to start the search from

Error Index **substring**

```
public String substring(int beginIndex)
```

Returns the substring of this String. The substring is specified by a beginIndex (inclusive) and the end of the string.

Parameters:

beginIndex - the beginning index, inclusive

Error Index **substring**

```
public String substring(int beginIndex,  
int endIndex)
```

Returns the substring of a String. The substring is specified by a beginIndex (inclusive) and an endIndex (exclusive).

Parameters:

beginIndex - the beginning index, inclusive

endIndex - the ending index, exclusive

Throws: StringIndexOutOfBoundsException

If the beginIndex or the endIndex is out of range.

Error Index **concat**

```
public String concat(String str)
```

Concatenates the specified string to the end of this String.

Parameters:

str - the String which is concatenated to the end of this String

Error Index **replace**

```
public String replace(char oldChar,  
                      char newChar)
```

Converts this String by replacing all occurrences of oldChar with newChar.

Parameters:

oldChar - the old character

newChar - the new character

Error Index **toLowerCase**

```
public String toLowerCase()
```

Converts all of the characters in this String to lower case.

Returns:

the String, converted to lowercase.

See Also:

[toLowerCase](#), [toUpperCase](#)

Error Index **toUpperCase**

```
public String toUpperCase()
```

Converts all of the characters in this String to upper case.

Returns:

the String, converted to uppercase.

See Also:

[toUpperCase](#), [toLowerCase](#)

Error Index **trim**

```
public String trim()
```

Trims leading and trailing whitespace from this String.

Returns:

the String, with whitespace removed.

Error Index toString

```
public String toString()
```

Converts this String to a String.

Returns:

the String itself.

Overrides:

toString in class Object

Error Index toCharArray

```
public char[] toCharArray()
```

Converts this String to a character array. This creates a new array.

Returns:

an array of characters.

Error Index valueOf

```
public static String valueOf(Object obj)
```

Returns a String that represents the String value of the object. The object may choose how to represent itself by implementing the toString() method.

Parameters:

obj - the object to be converted

Error Index valueOf

```
public static String valueOf(char data[])
```

Returns a String that is equivalent to the specified character array. Uses the original array as the body of the String (ie. it does not copy it to a new array).

Parameters:

data - the character array

Error Index valueOf

```
public static String valueOf(char data[],  
                             int offset,  
                             int count)
```

Returns a String that is equivalent to the specified character array.

Parameters:

data - the character array

offset - the offset into the value of the String

count - the length of the value of the String

Error Index copyValueOf

```
public static String copyValueOf(char data[],  
                                 int offset,  
                                 int count)
```

Returns a String that is equivalent to the specified character array. It creates a new array and copies the characters into it.

Parameters:

data - the character array

offset - the offset into the value of the String

count - the length of the value of the String

Error Index copyValueOf

```
public static String copyValueOf(char data[])
```

Returns a String that is equivalent to the specified character array. It creates a new array and copies the characters into it.

Parameters:

data - the character array

Error Index **valueOf**

public static String valueOf(boolean b)

Returns a String object that represents the state of the specified boolean.

Parameters:

b - the boolean

Error Index **valueOf**

public static String valueOf(char c)

Returns a String object that contains a single character

Parameters:

c - the character

Returns:

the resulting String.

Error Index **valueOf**

public static String valueOf(int i)

Returns a String object that represents the value of the specified integer.

Parameters:

i - the integer

Error Index **valueOf**

public static String valueOf(long l)

Returns a String object that represents the value of the specified long.

Parameters:

l - the long

Error Index **valueOf**

```
public static String valueOf(float f)
```

Returns a String object that represents the value of the specified float.

Parameters:

f - the float

Error Index `valueOf`

```
public static String valueOf(double d)
```

Returns a String object that represents the value of the specified double.

Parameters:

d - the double

Error Index `intern`

```
public String intern()
```

Returns a String that is equal to this String but which is guaranteed to be from the unique String pool.

For example:

```
s1.intern() == s2.intern() s1.equals(s2).
```

Class java.lang.StringBuffer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.StringBuffer

[java.lang.Object](#)

|

+----java.lang.StringBuffer

public final class **StringBuffer**
extends [Object](#)

This Class is a growable buffer for characters. It is mainly used to create Strings. The compiler uses it to implement the "+" operator. For example:

```
"a" + 4 + "c"
```

is compiled to:

```
new StringBuffer().append("a").append(4).append("c").toString()
```

Note that the method `toString()` does not create a copy of the internal buffer. Instead the buffer is marked as shared. Any further changes to the buffer will cause a copy to be made.

See Also:

[String](#), [ByteArrayOutputStream](#)

Error Index

Error Index

StringBuffer()

Constructs an empty String buffer.

Error Index

StringBuffer(int)

Constructs an empty String buffer with the specified initial length.

Error Index

StringBuffer(String)

Constructs a String buffer with the specified initial value.

Error Index

Error Index

append(Object)

Appends an object to the end of this buffer.

Error Index

append(String)

Appends a String to the end of this buffer.

Error Index

append(char[])

Appends an array of characters to the end of this buffer.

Error Index

append(char[], int, int)

Appends a part of an array of characters to the end of this buffer.

Error Index

append(boolean)

Appends a boolean to the end of this buffer.

Error Index

append(char)

Appends a character to the end of this buffer.

Error Index

append(int)

Appends an integer to the end of this buffer.

Error Index

append(long)

Appends a long to the end of this buffer.

Error Index

append(float)

Appends a float to the end of this buffer.

Error Index

append(double)

Appends a double to the end of this buffer.

Error Index

capacity()

Returns the current capacity of the String buffer.

Error Index

charAt(int)

Returns the character at the specified index.

Error Index

ensureCapacity(int)

Ensures that the capacity of the buffer is at least equal to the specified minimum.

Error Index

getChars(int, int, char[], int)

Copies the characters of the specified substring (determined by srcBegin and srcEnd) into the character array, starting at the array's dstBegin location.

Error Index

insert(int, Object)

Inserts an object into the String buffer.

Error Index

insert(int, String)

Inserts a String into the String buffer.

Error Index

insert(int, char[])

Inserts an array of characters into the String buffer.

Error Index

insert(int, boolean)

Inserts a boolean into the String buffer.

Error Index

insert(int, char)

Inserts a character into the String buffer.

Error Index

insert(int, int)

Inserts an integer into the String buffer.

Error Index

insert(int, long)

Inserts a long into the String buffer.

Error Index

insert(int, float)

Inserts a float into the String buffer.

Error Index

insert(int, double)

Inserts a double into the String buffer.

Error Index

length()

Returns the length (character count) of the buffer.

Error Index

setCharAt(int, char)

Changes the character at the specified index to be ch.

Error Index

setLength(int)

Sets the length of the String.

Error Index

toString()

Converts to a String representing the data in the buffer.

Error Index

Error Index

StringBuffer

```
public StringBuffer()
```

Constructs an empty String buffer.

Error Index `StringBuffer`

```
public StringBuffer(int length)
```

Constructs an empty String buffer with the specified initial length.

Parameters:

length - the initial length

Error Index `StringBuffer`

```
public StringBuffer(String str)
```

Constructs a String buffer with the specified initial value.

Parameters:

str - the initial value of the buffer

Error Index

Error Index `length`

```
public int length()
```

Returns the length (character count) of the buffer.

Error Index `capacity`

```
public int capacity()
```

Returns the current capacity of the String buffer. The capacity is the amount of storage available for newly inserted characters; beyond which an allocation will occur.

Error Index ensureCapacity

```
public synchronized void ensureCapacity(int minimumCapacity)
```

Ensures that the capacity of the buffer is at least equal to the specified minimum.

Parameters:

minimumCapacity - the minimum desired capacity

Error Index setLength

```
public synchronized void setLength(int newLength)
```

Sets the length of the String. If the length is reduced, characters are lost. If the length is extended, the values of the new characters are set to 0.

Parameters:

newLength - the new length of the buffer

Throws: StringIndexOutOfBoundsException

If the length is invalid.

Error Index charAt

```
public synchronized char charAt(int index)
```

Returns the character at the specified index. An index ranges from 0..length()-1.

Parameters:

index - the index of the desired character

Throws: StringIndexOutOfBoundsException

If the index is invalid.

Error Index getChars

```
public synchronized void getChars(int srcBegin,  
    int srcEnd,  
    char dst[],  
    int dstBegin)
```

Copies the characters of the specified substring (determined by srcBegin and srcEnd) into the character array, starting at the array's dstBegin location. Both srcBegin and srcEnd must be legal indexes into the buffer.

Parameters:

srcBegin - begin copy at this offset in the String
srcEnd - stop copying at this offset in the String
dst - the array to copy the data into
dstBegin - offset into dst

Throws: StringIndexOutOfBoundsException

If there is an invalid index into the buffer.

Error Index

setCharAt

```
public synchronized void setCharAt(int index,  
                                   char ch)
```

Changes the character at the specified index to be ch.

Parameters:

index - the index of the character
ch - the new character

Throws: StringIndexOutOfBoundsException

If the index is invalid.

Error Index

append

```
public synchronized StringBuffer append(Object obj)
```

Appends an object to the end of this buffer.

Parameters:

obj - the object to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Index

append

```
public synchronized StringBuffer append(String str)
```

Appends a String to the end of this buffer.

Parameters:

str - the String to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Index **append**

public synchronized StringBuffer append(char str[])

Appends an array of characters to the end of this buffer.

Parameters:

str - the characters to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Index **append**

public synchronized StringBuffer append(char str[],
int offset,
int len)

Appends a part of an array of characters to the end of this buffer.

Parameters:

str - the characters to be appended

offset - where to start

len - the number of characters to add

Returns:

the StringBuffer itself, NOT a new one.

Error Index **append**

public StringBuffer append(boolean b)

Appends a boolean to the end of this buffer.

Parameters:

b - the boolean to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Index **append**

public synchronized StringBuffer append(char c)

Appends a character to the end of this buffer.

Parameters:

ch - the character to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Indexappend

```
public StringBuffer append(int i)
```

Appends an integer to the end of this buffer.

Parameters:

i - the integer to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Indexappend

```
public StringBuffer append(long l)
```

Appends a long to the end of this buffer.

Parameters:

l - the long to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Indexappend

```
public StringBuffer append(float f)
```

Appends a float to the end of this buffer.

Parameters:

f - the float to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Indexappend

```
public StringBuffer append(double d)
```

Appends a double to the end of this buffer.

Parameters:

d - the double to be appended

Returns:

the StringBuffer itself, NOT a new one.

Error Index insert

public synchronized StringBuffer insert(int offset,
Object obj)

Inserts an object into the String buffer.

Parameters:

offset - the offset at which to insert

obj - the object to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

public synchronized StringBuffer insert(int offset,
String str)

Inserts a String into the String buffer.

Parameters:

offset - the offset at which to insert

str - the String to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

public synchronized StringBuffer insert(int offset,
char str[])

Inserts an array of characters into the String buffer.

Parameters:

offset - the offset at which to insert
str - the characters to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

```
public StringBuffer insert(int offset,  
                           boolean b)
```

Inserts a boolean into the String buffer.

Parameters:

offset - the offset at which to insert
b - the boolean to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

```
public synchronized StringBuffer insert(int offset,  
                                           char c)
```

Inserts a character into the String buffer.

Parameters:

offset - the offset at which to insert
ch - the character to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

```
public StringBuffer insert(int offset,  
                           int i)
```

Inserts an integer into the String buffer.

Parameters:

offset - the offset at which to insert
i - the integer to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

```
public StringBuffer insert(int offset,  
                           long l)
```

Inserts a long into the String buffer.

Parameters:

offset - the offset at which to insert
l - the long to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

```
public StringBuffer insert(int offset,  
                           float f)
```

Inserts a float into the String buffer.

Parameters:

offset - the offset at which to insert
f - the float to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index insert

```
public StringBuffer insert(int offset,  
                           double d)
```

Inserts a double into the String buffer.

Parameters:

offset - the offset at which to insert
d - the double to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: StringIndexOutOfBoundsException

If the offset is invalid.

Error Index toString

public String toString()

Converts to a String representing the data in the buffer.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.StringIndexOutOfBoundsException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class

java.lang.StringIndexOutOfBoundsException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.lang.IndexOutOfBoundsException
                        |
                        +----java.lang.StringIndexOutOfBoundsException
```

```
public class StringIndexOutOfBoundsException
    extends IndexOutOfBoundsException
```

Signals that a String index is out of range.

See Also:
[charAt](#)

Error Index

Error Index

StringIndexOutOfBoundsException()

Constructs a StringIndexOutOfBoundsException with no detail message.

Error Index

StringIndexOutOfBoundsException(String)

Constructs a StringIndexOutOfBoundsException with the specified detail message.

Error Index

StringIndexOutOfBoundsException(int)

Constructs a `StringIndexOutOfBoundsException` initialized with the specified index.

Error Index

Error Index

`StringIndexOutOfBoundsException`

```
public StringIndexOutOfBoundsException()
```

Constructs a `StringIndexOutOfBoundsException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index

`StringIndexOutOfBoundsException`

```
public StringIndexOutOfBoundsException(String s)
```

Constructs a `StringIndexOutOfBoundsException` with the specified detail message. A detail message is a `String` that describes this particular exception.

Parameters:

s - the `String` containing a detail message about the error

Error Index

`StringIndexOutOfBoundsException`

```
public StringIndexOutOfBoundsException(int index)
```

Constructs a `StringIndexOutOfBoundsException` initialized with the specified index.

Parameters:

index - the offending index

Class java.lang.System

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.System

[java.lang.Object](#)

|
+----java.lang.System

```
public final class System  
extends Object
```

This Class provides a system-independent interface to system functionality. One of the more useful things provided by this Class are the standard input and output streams. The standard input streams are used for reading character data. The standard output streams are used for printing. For example:

```
System.out.println("Hello World!");
```

This Class cannot be instantiated or subclassed because all of the methods and variables are static.

Error Index

Error Index

Standard error stream.

err

Error Index

Standard input stream.

in

Error Index

Standard output stream.

out

Error Index

Error Index

arraycopy(Object, int, Object, int, int)

Copies an array from the source array, beginning at the specified position, to the specified position of the destination array.

Error Index

currentTimeMillis()

Returns the current time in milliseconds GMT since the epoch (00:00:00 UTC, January 1, 1970).

Error Index

exit(int)

Exits the virtual machine with an exit code.

Error Index

gc()

Runs the garbage collector.

Error Index

getProperties()

Gets the System properties.

Error Index

getProperty(String)

Gets the System property indicated by the specified key.

Error Index

getProperty(String, String)

Gets the System property indicated by the specified key and def.

Error Index

getSecurityManager()

Gets the system security interface.

Error Index

getenv(String)

Obsolete.

Error Index

load(String)

Loads a dynamic library, given a complete path name.

Error Index

loadLibrary(String)

Loads a dynamic library with the specified library name.

Error Index

runFinalization()

Runs the finalization methods of any objects pending finalization.

Error Index

setProperties(Properties)

Sets the System properties to the specified properties.

Error Index

setSecurityManager(SecurityManager)

Sets the System security.

Error Index

Error Index in

public static InputStream in

Standard input stream. This stream is used for reading in character data.

Error Index out

public static PrintStream out

Standard output stream. This stream is used for printing messages.

Error Index err

public static PrintStream err

Standard error stream. This stream can be used to print error messages. Many applications read in data from an InputStream and output messages via the PrintStream out statement. Often applications rely on command line redirection to specify source and destination files. A problem with redirecting standard output is the incapability of writing messages to the screen if the output has been redirected to a file. This problem can be overcome by sending some output to PrintStream out and other output to PrintStream err. The difference between PrintStream err and PrintStream out is that PrintStream err is often used for displaying error messages but may be used for any purpose.

Error Index

Error Index setSecurityManager

public static void setSecurityManager(SecurityManager s)

Sets the System security. This value can only be set once.

Parameters:

s - the security manager

Throws: SecurityException

If the SecurityManager has already been set.

Error Index getSecurityManager

public static SecurityManager getSecurityManager()

Gets the system security interface.

Error Index currentTimeMillis

```
public static long currentTimeMillis()
```

Returns the current time in milliseconds GMT since the epoch (00:00:00 UTC, January 1, 1970). It is a signed 64 bit integer, and so it will not overflow until the year 292280995.

See Also:

Date

Error Index arraycopy

```
public static void arraycopy(Object src,  
                             int src_position,  
                             Object dst,  
                             int dst_position,  
                             int length)
```

Copies an array from the source array, beginning at the specified position, to the specified position of the destination array. This method does not allocate memory for the destination array. The memory must already be allocated.

Parameters:

src - the source data
srcpos - start position in the source data
dest - the destination
destpos - start position in the destination data
length - the number of array elements to be copied

Throws: ArrayIndexOutOfBoundsException

If copy would cause access of data outside array bounds.

Throws: ArrayStoreException

If an element in the src array could not be stored into the destination array due to a type mismatch

Error Index getProperties

```
public static Properties getProperties()
```

Gets the System properties.

Error Index setProperties

```
public static void setProperties(Properties props)
```

Sets the System properties to the specified properties.

Parameters:

props - the properties to be set

Error Index `getProperty`

```
public static String getProperty(String key)
```

Gets the System property indicated by the specified key.

Parameters:

key - the name of the system property

Error Index `getProperty`

```
public static String getProperty(String key,  
                                String def)
```

Gets the System property indicated by the specified key and def.

Parameters:

key - the name of the system property

def - the default value to use if this property is not set

Error Index `getenv`

```
public static String getenv(String name)
```

Obsolete. Gets an environment variable. An environment variable is a system dependent external variable that has a string value.

Parameters:

name - the name of the environment variable

Returns:

the value of the variable, or null if the variable is not defined.

Error Index `exit`

```
public static void exit(int status)
```

Exits the virtual machine with an exit code. This method does not return, use with caution.

Parameters:

status - exit status, 0 if successful, other values indicate various error types.

See Also:

exit

Error Index_{gc}

public static void gc()

Runs the garbage collector.

See Also:

gc

Error Index_{runFinalization}

public static void runFinalization()

Runs the finalization methods of any objects pending finalization.

See Also:

gc

Error Index_{load}

public static void load(String filename)

Loads a dynamic library, given a complete path name.

Parameters:

filename - the file to load

Throws: UnsatisfiedLinkError

If the file does not exist.

See Also:

load

Error Index_{loadLibrary}

public static void loadLibrary(String libname)

Loads a dynamic library with the specified library name.

Parameters:

libname - the name of the library

Throws: UnsatisfiedLinkError

If the library does not exist.

See Also:

loadLibrary

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Thread

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Thread

[java.lang.Object](#)

|
+----java.lang.Thread

```
public class Thread
  extends Object
  implements Runnable
```

A Thread is a single sequential flow of control within a process. This simply means that while executing within a program, each thread has a beginning, a sequence, a point of execution occurring at any time during runtime of the thread and of course, an ending. Thread objects are the basis for multi-threaded programming. Multi-threaded programming allows a single program to conduct concurrently running threads that perform different tasks.

To create a new thread of execution, declare a new class which is a subclass of Thread and then override the run() method with code that you want executed in this Thread. An instance of the Thread subclass should be created next with a call to the start() method following the instance. The start() method will create the thread and execute the run() method. For example:

```
class PrimeThread extends Thread {
    public void run() {
        // compute primes...
    }
}
```

To start this thread you need to do the following:

```
PrimeThread p = new PrimeThread();
p.start();
...
```

Another way to create a thread is by using the Runnable interface. This way any object that implements the Runnable interface can be run in a thread. For example:

```
class Primes implements Runnable {
    public void run() {
        // compute primes...
    }
}
```

To start this thread you need to do the following:

```
Primes p = new Primes();  
new Thread(p).start();  
...
```

The virtual machine runs until all Threads that are not daemon Threads have died. A Thread dies when its run() method returns, or when the stop() method is called.

When a new Thread is created, it inherits the priority and the daemon flag from its parent (i.e.: the Thread that created it).

See Also:

[Runnable](#)

Error Index

Error Index

MAX_PRIORITY

The maximum priority that a Thread can have.

Error Index

MIN_PRIORITY

The minimum priority that a Thread can have.

Error Index

NORM_PRIORITY

The default priority that is assigned to a Thread.

Error Index

Error Index

Thread()

Constructs a new Thread.

Error Index

Thread(Runnable)

Constructs a new Thread which applies the run() method of the specified target.

Error Index

Thread(ThreadGroup, Runnable)

Constructs a new Thread in the specified Thread group that applies the run() method of the specified target.

Error Index

Thread(String)

Constructs a new Thread with the specified name.

Error Index

Thread(ThreadGroup, String)

Constructs a new Thread in the specified Thread group with the specified name.

Error Index

Thread(Runnable, String)

Constructs a new Thread with the specified name and applies the run() method of the specified target.

Error Index

Thread(ThreadGroup, Runnable, String)

Constructs a new Thread in the specified Thread group with the specified name and applies the run() method of the specified target.

Error Index

Error Index

activeCount()

Returns the current number of active Threads in this Thread group.

Error Index

checkAccess()

Checks whether the current Thread is allowed to modify this Thread.

Error Index

countStackFrames()

Returns the number of stack frames in this Thread.

Error Index

currentThread()

Returns a reference to the currently executing Thread object.

Error Index

destroy()

Destroy a thread, without any cleanup, i.e.

Error Index

dumpStack()

A debugging procedure to print a stack trace for the current Thread.

Error Index

enumerate(Thread[])

Copies, into the specified array, references to every active Thread in this Thread's group.

Error Index

getName()

Gets and returns this Thread's name.

Error Index

getPriority()

Gets and returns the Thread's priority.

Error Index

getThreadGroup()

Gets and returns this Thread group.

Error Index

interrupt()

Send an interrupt to a thread.

Error Index

interrupted()

Ask if you have been interrupted.

Error Index

isAlive()

Returns a boolean indicating if the Thread is active.

Error Index

isDaemon()

Returns the daemon flag of the Thread.

Error Index

isInterrupted()

Ask if another thread has been interrupted.

Error Index

join(long)

Waits for this Thread to die.

Error Index

join(long, int)

Waits for the Thread to die, with more precise time.

Error Index

join()

Waits forever for this Thread to die.

Error Index

resume()

Resumes this Thread execution.

Error Index

run()

The actual body of this Thread.

Error Index

setDaemon(boolean)

Marks this Thread as a daemon Thread or a user Thread.

Error Index

setName(String)

Sets the Thread's name.

Error Index

setPriority(int)

Sets the Thread's priority.

Error Index

sleep(long)

Causes the currently executing Thread to sleep for the specified number of milliseconds.

Error Index

sleep(long, int)

Sleep, in milliseconds and additional nanosecond.

Error Index

start()

Starts this Thread.

Error Index

stop()

Stops a Thread by tossing an object.

Error Index

stop(Throwable)

Stops a Thread by tossing an object.

Error Index

suspend()

Suspends this Thread's execution.

Error Index

toString()

Returns a String representation of the Thread, including the thread's name, priority and thread group.

Error Index

yield()

Causes the currently executing Thread object to yield.

Error Index

Error Index

MIN_PRIORITY

```
public final static int MIN_PRIORITY
```

The minimum priority that a Thread can have. The most minimal priority is equal to 1.

Error Index

NORM_PRIORITY

```
public final static int NORM_PRIORITY
```

The default priority that is assigned to a Thread. The default priority is equal to 5.

Error Index

MAX_PRIORITY

```
public final static int MAX_PRIORITY
```

The maximum priority that a Thread can have. The maximal priority value a Thread can have is 10.

Error Index

Error Index Thread

```
public Thread()
```

Constructs a new Thread. Threads created this way must have overridden their run() method to actually do anything. An example illustrating this method being used is shown.

```
import java.lang.*;
```

```
class plain01 implements Runnable {
    String name;
    plain01() {
        name = null;
    }
    plain01(String s) {
        name = s;
    }
    public void run() {
        if (name == null)
            System.out.println("A new thread created");
        else
            System.out.println("A new thread with name " + name + " created");
    }
}
```

```
class threadtest01 {
    public static void main(String args[] ) {
        int failed = 0 ;
```

```
        Thread t1 = new Thread();
        if(t1 != null) {
            System.out.println("new Thread() succeed");
        } else {
            System.out.println("new Thread() failed");
            failed++;
        }
    }
}
```

Error Index Thread

```
public Thread(Runnable target)
```

Constructs a new Thread which applies the run() method of the specified target.

Parameters:

target - the object whose run() method is called

Error Index Thread

```
public Thread(ThreadGroup group,  
             Runnable target)
```

Constructs a new Thread in the specified Thread group that applies the run() method of the specified target.

Parameters:

group - the Thread group

target - the object whose run() method is called

Error Index Thread

```
public Thread(String name)
```

Constructs a new Thread with the specified name.

Parameters:

name - the name of the new Thread

Error Index Thread

```
public Thread(ThreadGroup group,  
             String name)
```

Constructs a new Thread in the specified Thread group with the specified name.

Parameters:

group - the Thread group

name - the name of the new Thread

Error Index Thread

```
public Thread(Runnable target,  
             String name)
```

Constructs a new Thread with the specified name and applies the run() method of the specified target.

Parameters:

target - the object whose run() method is called
name - the name of the new Thread

Error Index Thread

```
public Thread(ThreadGroup group,  
             Runnable target,  
             String name)
```

Constructs a new Thread in the specified Thread group with the specified name and applies the run() method of the specified target.

Parameters:

group - the Thread group
target - the object whose run() method is called
name - the name of the new Thread

Error Index

Error Index currentThread

```
public static Thread currentThread()
```

Returns a reference to the currently executing Thread object.

Error Index yield

```
public static void yield()
```

Causes the currently executing Thread object to yield. If there are other runnable Threads they will be scheduled next.

Error Index sleep

```
public static void sleep(long millis) throws InterruptedException
```

Causes the currently executing Thread to sleep for the specified number of milliseconds.

Parameters:

 millis - the length of time to sleep in milliseconds

Throws: InterruptedException

 Another thread has interrupted this thread.

Error Index sleep

public static void sleep(long millis,
 int nanos) throws InterruptedException

Sleep, in milliseconds and additional nanosecond.

Parameters:

 millis - the length of time to sleep in milliseconds

 nanos - 0-999999 additional nanoseconds to sleep

Throws: InterruptedException

 Another thread has interrupted this thread.

Error Index start

public synchronized void start()

Starts this Thread. This will cause the run() method to be called. This method will return immediately.

Throws: IllegalThreadStateException

 If the thread was already started.

See Also:

run, stop

Error Index run

public void run()

The actual body of this Thread. This method is called after the Thread is started. You must either override this method by subclassing class Thread, or you must create the Thread with a Runnable target.

See Also:

start, stop

Error Index stop

public final void stop()

Stops a Thread by tossing an object. By default this routine tosses a new instance of ThreadDeath to the target Thread. ThreadDeath is not actually a subclass of Exception, but is a subclass of Object. Users should not normally try to catch ThreadDeath unless they must do some extraordinary cleanup operation. If ThreadDeath is caught it is important to rethrow the object so that the thread will actually die. The top-level error handler will not print out a message if ThreadDeath falls through.

See Also:

start, run

Error Index stop

public final synchronized void stop(Throwable o)

Stops a Thread by tossing an object. Normally, users should just call the stop() method without any argument. However, in some exceptional circumstances used by the stop() method to kill a Thread, another object is tossed. ThreadDeath, is not actually a subclass of Exception, but is a subclass of Throwable

Parameters:

o - the Throwable object to be thrown

See Also:

start, run

Error Index interrupt

public void interrupt()

Send an interrupt to a thread.

Error Index interrupted

public static boolean interrupted()

Ask if you have been interrupted.

Error Index isInterrupted

public boolean isInterrupted()

Ask if another thread has been interrupted.

Error Index destroy

```
public void destroy()
```

Destroy a thread, without any cleanup, i.e. just toss its state; any monitors it has locked remain locked. A last resort.

Error Index isAlive

```
public final boolean isAlive()
```

Returns a boolean indicating if the Thread is active. Having an active Thread means that the Thread has been started and has not been stopped.

Error Index suspend

```
public final void suspend()
```

Suspends this Thread's execution.

Error Index resume

```
public final void resume()
```

Resumes this Thread execution. This method is only valid after suspend() has been invoked.

Error Index setPriority

```
public final void setPriority(int newPriority)
```

Sets the Thread's priority.

Throws: IllegalArgumentException

If the priority is not within the range MIN_PRIORITY, MAX_PRIORITY.

See Also:

MIN_PRIORITY, MAX_PRIORITY, getPriority

Error Index **getPriority**

```
public final int getPriority()
```

Gets and returns the Thread's priority.

See Also:

setPriority

Error Index **setName**

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

Sets the Thread's name.

Parameters:

name - the new name of the Thread

See Also:

getName

Error Index **getName**

```
public final String getName()
```

Gets and returns this Thread's name.

See Also:

setName

Error Index **getThreadGroup**

```
public final ThreadGroup getThreadGroup()
```

Gets and returns this Thread group.

Error Index **activeCount**

```
public static int activeCount()
```

Returns the current number of active Threads in this Thread group.

Error Index enumerate

public static int enumerate(Thread tarray[])

Copies, into the specified array, references to every active Thread in this Thread's group.

Returns:

the number of Threads put into the array.

Error Index countStackFrames

public int countStackFrames()

Returns the number of stack frames in this Thread. The Thread must be suspended when this method is called.

Throws: IllegalThreadStateException

If the Thread is not suspended.

Error Index join

public final synchronized void join(long millis) throws InterruptedException

Waits for this Thread to die. A timeout in milliseconds can be specified. A timeout of 0 milliseconds means to wait forever.

Parameters:

millis - the time to wait in milliseconds

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index join

public final synchronized void join(long millis,
int nanos) throws InterruptedException

Waits for the Thread to die, with more precise time.

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index join

public final void join() throws InterruptedException

Waits forever for this Thread to die.

Throws: InterruptedException

Another thread has interrupted this thread.

Error Index dumpStack

public static void dumpStack()

A debugging procedure to print a stack trace for the current Thread.

See Also:

printStackTrace

Error Index setDaemon

public final void setDaemon(boolean on)

Marks this Thread as a daemon Thread or a user Thread. When there are only daemon Threads left running in the system, Java exits.

Parameters:

on - determines whether the Thread will be a daemon Thread

Throws: IllegalThreadStateException

If the Thread is active.

See Also:

isDaemon

Error Index isDaemon

public final boolean isDaemon()

Returns the daemon flag of the Thread.

See Also:

setDaemon

Error Index `checkAccess`

public void checkAccess()

Checks whether the current Thread is allowed to modify this Thread.

Throws: SecurityException

If the current Thread is not allowed to access this Thread group.

Error Index `toString`

public String toString()

Returns a String representation of the Thread, including the thread's name, priority and thread group.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ThreadDeath

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ThreadDeath

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.ThreadDeath
```

```
public class ThreadDeath
extends Error
```

An instance of ThreadDeath is thrown in the victim thread when thread.stop() is called. This is not a subclass of Exception, but rather a subclass of Error because too many people already catch Exception. Instances of this class should be caught explicitly only if you are interested in cleaning up when being asynchronously terminated. If ThreadDeath is caught, it is important to rethrow the object so that the Thread will actually die. The top-level error handler will not print out a message if ThreadDeath falls through.

Error Index

Error Index

ThreadDeath()

Error Index

Error Index

ThreadDeath

```
public ThreadDeath()
```

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ThreadGroup

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.ThreadGroup

[java.lang.Object](#)

|
+----java.lang.ThreadGroup

```
public class ThreadGroup  
extends Object
```

A group of Threads. A Thread group can contain a set of Threads as well as a set of other Thread groups. A Thread can access its Thread group, but it can't access the parent of its Thread group. This makes it possible to encapsulate a Thread in a Thread group and stop it from manipulating Threads in the parent group.

Error Index

Error Index

ThreadGroup(String)

Creates a new ThreadGroup.

Error Index

ThreadGroup(ThreadGroup, String)

Creates a new ThreadGroup with a specified name in the specified Thread group.

Error Index

Error Index

activeCount()

Returns an estimate of the number of active Threads in the Thread group.

Error Index

activeGroupCount()

Returns an estimate of the number of active groups in the Thread group.

Error Index

checkAccess()

Checks to see if the current Thread is allowed to modify this group.

Error Index

Destroys a Thread group.

destroy()

Error Index

Copies, into the specified array, references to every active Thread in this Thread group.

enumerate(Thread[])

Error Index

Copies, into the specified array, references to every active Thread in this Thread group.

enumerate(Thread[], boolean)

Error Index

Copies, into the specified array, references to every active Thread group in this Thread group.

enumerate(ThreadGroup[])

Error Index

Copies, into the specified array, references to every active Thread group in this Thread group.

enumerate(ThreadGroup[], boolean)

Error Index

Gets the maximum priority of the group.

getMaxPriority()

Error Index

Gets the name of this Thread group.

getName()

Error Index

Gets the parent of this Thread group.

getParent()

Error Index

Returns the daemon flag of the Thread group.

isDaemon()

Error Index

Lists this Thread group.

list()

Error Index

Checks to see if this Thread group is a parent of or is equal to another Thread group.

parentOf(ThreadGroup)

Error Index

Resumes all the Threads in this Thread group and all of its sub groups.

resume()

Error Index

Changes the daemon status of this group.

setDaemon(boolean)

Error Index

Sets the maximum priority of the group.

setMaxPriority(int)

Error Index

Stops all the Threads in this Thread group and all of its sub groups.

stop()

Error Index

suspend()

Suspends all the Threads in this Thread group and all of its sub groups.

Error Index

toString()

Returns a String representation of the Thread group.

Error Index

uncaughtException(Thread, Throwable)

Called when a thread in this group exists because of an uncaught exception.

Error Index

Error Index

ThreadGroup

```
public ThreadGroup(String name)
```

Creates a new ThreadGroup. Its parent will be the Thread group of the current Thread.

Parameters:

name - the name of the new Thread group created

Error Index

ThreadGroup

```
public ThreadGroup(ThreadGroup parent,  
                  String name)
```

Creates a new ThreadGroup with a specified name in the specified Thread group.

Parameters:

parent - the specified parent Thread group

name - the name of the new Thread group being created

Throws: NullPointerException

If the given thread group is equal to null.

Error Index

Error Index

getName

```
public final String getName()
```


Gets the name of this Thread group.

Error Index getParent

```
public final ThreadGroup getParent()
```

Gets the parent of this Thread group.

Error Index getMaxPriority

```
public final int getMaxPriority()
```

Gets the maximum priority of the group. Threads that are part of this group cannot have a higher priority than the maximum priority.

Error Index isDaemon

```
public final boolean isDaemon()
```

Returns the daemon flag of the Thread group. A daemon Thread group is automatically destroyed when it is found empty after a Thread group or Thread is removed from it.

Error Index setDaemon

```
public final void setDaemon(boolean daemon)
```

Changes the daemon status of this group.

Parameters:

daemon - the daemon boolean which is to be set.

Error Index setMaxPriority

```
public final synchronized void setMaxPriority(int pri)
```

Sets the maximum priority of the group. Threads that are already in the group **can** have a higher priority than the set maximum.

Parameters:

pri - the priority of the Thread group

Error Index

parentOf

```
public final boolean parentOf(ThreadGroup g)
```

Checks to see if this Thread group is a parent of or is equal to another Thread group.

Parameters:

g - the Thread group to be checked

Returns:

true if this Thread group is equal to or is the parent of another Thread group; false otherwise.

Error Index

checkAccess

```
public final void checkAccess()
```

Checks to see if the current Thread is allowed to modify this group.

Throws: SecurityException

If the current Thread is not allowed to access this Thread group.

Error Index

activeCount

```
public synchronized int activeCount()
```

Returns an estimate of the number of active Threads in the Thread group.

Error Index

enumerate

```
public int enumerate(Thread list[])
```

Copies, into the specified array, references to every active Thread in this Thread group. You can use the activeCount() method to get an estimate of how big the array should be.

Parameters:

list - an array of Threads

Returns:

the number of Threads put into the array

Error Index

enumerate

```
public int enumerate(Thread list[],  
                    boolean recurse)
```

Copies, into the specified array, references to every active Thread in this Thread group. You can use the activeCount() method to get an estimate of how big the array should be.

Parameters:

list - an array list of Threads

recurse - a boolean indicating whether a Thread has reappeared

Returns:

the number of Threads placed into the array.

Error Index

activeGroupCount

```
public synchronized int activeGroupCount()
```

Returns an estimate of the number of active groups in the Thread group.

Error Index

enumerate

```
public int enumerate(ThreadGroup list[])
```

Copies, into the specified array, references to every active Thread group in this Thread group. You can use the activeGroupCount() method to get an estimate of how big the array should be.

Parameters:

list - an array of Thread groups

Returns:

the number of Thread groups placed into the array.

Error Index

enumerate

```
public int enumerate(ThreadGroup list[],  
                    boolean recurse)
```

Copies, into the specified array, references to every active Thread group in this Thread group. You can use the activeGroupCount() method to get an estimate of how big the array should be.

Parameters:

list - an array list of Thread groups

recurse - a boolean indicating if a Thread group has reappeared

Returns:

the number of Thread groups placed into the array.

Error Index stop

```
public final synchronized void stop()
```

Stops all the Threads in this Thread group and all of its sub groups.

Error Index suspend

```
public final synchronized void suspend()
```

Suspends all the Threads in this Thread group and all of its sub groups.

Error Index resume

```
public final synchronized void resume()
```

Resumes all the Threads in this Thread group and all of its sub groups.

Error Index destroy

```
public final synchronized void destroy()
```

Destroys a Thread group. This does **NOT** stop the Threads in the Thread group.

Throws: IllegalThreadStateException

If the Thread group is not empty

or if the Thread group was already destroyed.

Error Index list

```
public synchronized void list()
```

Lists this Thread group. Useful for debugging only.

Error Index uncaughtException

```
public void uncaughtException(Thread t,  
                               Throwable e)
```

Called when a thread in this group exists because of an uncaught exception.

Error Index toString

public String toString()

Returns a String representation of the Thread group.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Throwable

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.Throwable

[java.lang.Object](#)

|
+----java.lang.Throwable

```
public class Throwable  
extends Object
```

An object signalling that an exceptional condition has occurred. All exceptions are a subclass of Exception. An exception contains a snapshot of the execution stack, this snapshot is used to print a stack backtrace. An exception also contains a message string. Here is an example of how to catch an exception:

```
try {  
    int a[] = new int[2];  
    a[4];  
} catch (ArrayIndexOutOfBoundsException e) {  
    System.out.println("an exception occurred: " + e.getMessage());  
    e.printStackTrace();  
}
```

Error Index

Error Index

Throwable()

Constructs a new Throwable with no detail message.

Error Index

Throwable(String)

Constructs a new Throwable with the specified detail message.

Error Index

Error Index

fillInStackTrace()

Fills in the execution stack trace.

Error Index

getMessage()

Gets the detail message of the Throwable.

Error Index

printStackTrace()

Prints the Throwable and the Throwable's stack trace.

Error Index

printStackTrace(PrintStream)

Error Index

toString()

Returns a short description of the Throwable.

Error Index

Error Index

Throwable

public Throwable()

Constructs a new Throwable with no detail message. The stack trace is automatically filled in.

Error Index

Throwable

public Throwable(String message)

Constructs a new Throwable with the specified detail message. The stack trace is automatically filled in.

Parameters:

message - the detailed message

Error Index

Error Index

getMessage

public String getMessage()

Gets the detail message of the Throwable. A detail message is a String that describes the Throwable that has taken place.

Returns:

the detail message of the throwable.

Error Index toString

```
public String toString()
```

Returns a short description of the Throwable.

Overrides:

toString in class Object

Error Index printStackTrace

```
public void printStackTrace()
```

Prints the Throwable and the Throwable's stack trace.

Error Index printStackTrace

```
public void printStackTrace(PrintStream s)
```

Error Index fillInStackTrace

```
public Throwable fillInStackTrace()
```

Fills in the execution stack trace. This is useful only when rethrowing a Throwable. For example:

```
try {  
    a = b / c;  
} catch(ArithmeticThrowable e) {  
    a = Number.MAX_VALUE;  
    throw e.fillInStackTrace();  
}
```

Returns:

the Throwable itself.

See Also:

printStackTrace

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.UnknownError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.UnknownError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.VirtualMachineError
|
+----java.lang.UnknownError
```

```
public class UnknownError
extends VirtualMachineError
```

Signals that an unknown but serious exception has occurred.

Error Index

Error Index

UnknownError()

Constructs an UnknownError with no detail message.

Error Index

UnknownError(String)

Constructs an UnknownError with the specified detail message.

Error Index

Error Index

UnknownError

```
public UnknownError()
```

Constructs an UnknownError with no detail message. A detail message is a String that describes this particular exception.

Error Index

UnknownError

public UnknownError(String s)

Constructs an UnknownError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.UnsatisfiedLinkError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.UnsatisfiedLinkError

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Error
|
+----java.lang.LinkageError
|
+----java.lang.UnsatisfiedLinkError
```

```
public class UnsatisfiedLinkError
    extends LinkageError
```

Signals an unsatisfied link.

See Also:

[Runtime](#)

Error Index

Error Index

UnsatisfiedLinkError()

Constructs an UnsatisfiedLinkError with no detail message.

Error Index

UnsatisfiedLinkError(String)

Constructs an UnsatisfiedLinkError with the specified detail message.

Error Index

Error Index

UnsatisfiedLinkError

```
public UnsatisfiedLinkError()
```

Constructs an UnsatisfiedLinkError with no detail message. A detail message is a String that describes this particular exception.

Error Index

UnsatisfiedLinkError

```
public UnsatisfiedLinkError(String s)
```

Constructs an UnsatisfiedLinkError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.VerifyError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.VerifyError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.LinkageError
                  |
                  +----java.lang.VerifyError
```

```
public class VerifyError
    extends LinkageError
```

Signals that a Verification Error occurred.

Error Index

Error Index

[VerifyError\(\)](#)

Constructor.

Error Index

[VerifyError](#)(String)

Constructor with a detail message.

Error Index

Error Index

VerifyError

```
public VerifyError()
```

Constructor.

Error Index

VerifyError

```
public VerifyError(String s)
```

Constructor with a detail message.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.VirtualMachineError

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.lang.VirtualMachineError

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Error
            |
            +----java.lang.VirtualMachineError
```

```
public class VirtualMachineError
    extends Error
```

A VirtualMachineError indicates that the virtual machine is broken or has run out of resources.

Error Index

Error Index

VirtualMachineError()

Constructs a VirtualMachineError with no detail message.

Error Index

VirtualMachineError(String)

Constructs a VirtualMachineError with the specified detail message.

Error Index

Error Index

VirtualMachineError

```
public VirtualMachineError()
```

Constructs a VirtualMachineError with no detail message. A detail message is a String that describes this particular exception.

Error Index

VirtualMachineError

public VirtualMachineError(String s)

Constructs a VirtualMachineError with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.ContentHandler

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.ContentHandler

[java.lang.Object](#)

|

+----java.net.ContentHandler

```
public class ContentHandler
    extends Object
```

A class to read data from a `URLConnection` and construct an `Object`. Specific subclasses of `ContentHandler` handle specific mime types. It is the responsibility of a `ContentHandlerFactory` to select an appropriate `ContentHandler` for the mime-type of the `URLConnection`. Applications should never call `ContentHandlers` directly, rather they should use `URL.getContent()` or `URLConnection.getContent()`

Error Index

Error Index

[ContentHandler\(\)](#)

Error Index

Error Index

[getContent\(URLConnection\)](#)

Given an input stream positioned at the beginning of the representation of an object, reads that stream and recreates the object from it.

Error Index

Error Index

ContentHandler

```
public ContentHandler()
```

Error Index

Error Index

getContent

public abstract Object getContent(URLConnection urlc) throws IOException

Given an input stream positioned at the beginning of the representation of an object, reads that stream and recreates the object from it.

Throws: IOException

An IO error occurred while reading the object.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.net.ContentHandlerFactory

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.net.ContentHandlerFactory

public interface **ContentHandlerFactory**
extends [Object](#)

This interface defines a factory for ContentHandler instances. It is used by the URLStreamHandler class to create ContentHandlers for various streams.

Error Index

Error Index

[createContentHandler](#)(String)

Creates a new ContentHandler to read an object from a URLStreamHandler.

Error Index

Error Index

[createContentHandler](#)

public abstract [ContentHandler](#) createContentHandler([String](#) mimetype)

Creates a new ContentHandler to read an object from a URLStreamHandler.

Parameters:

mimetype - The mime type for which a content handler is desired.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.DatagramPacket

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.DatagramPacket

[java.lang.Object](#)

|

+----java.net.DatagramPacket

public final class **DatagramPacket**
extends [Object](#)

A class that represents a datagram packet containing packet data, packet length, internet addresses and port.

Error Index

Error Index

DatagramPacket(byte[], int)

This constructor is used to create a DatagramPacket object used for receiving datagrams.

Error Index

DatagramPacket(byte[], int, InetAddress, int)

This constructor is used construct the DatagramPacket to be sent.

Error Index

Error Index

getAddress()

Error Index

getData()

Error Index

getLength()

Error Index

getPort()

Error Index

Error Index

DatagramPacket

```
public DatagramPacket(byte ibuf[],  
    int ilength)
```

This constructor is used to create a DatagramPacket object used for receiving datagrams.

Parameters:

ibuf - is where packet data is to be received.
ilength - is the number of bytes to be received.

Error Index

DatagramPacket

```
public DatagramPacket(byte ibuf[],  
    int ilength,  
    InetAddress iaddr,  
    int iport)
```

This constructor is used to construct the DatagramPacket to be sent.

Parameters:

ibuf - contains the packet data.
ilength - contains the packet length
iaddr - and iport contains destination ip addr and port number.

Error Index

Error Index

getAddress

```
public InetAddress getAddress()
```

Error Index

getPort

```
public int getPort()
```

Error Index **getData**

```
public byte[] getData()
```

Error Index **getLength**

```
public int getLength()
```

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.DatagramSocket

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.DatagramSocket

[java.lang.Object](#)

|

+----java.net.DatagramSocket

```
public class DatagramSocket
    extends Object
```

The datagram socket class implements unreliable datagrams.

Error Index

Error Index

[**DatagramSocket\(\)**](#)

Creates a datagram socket

Error Index

[**DatagramSocket\(int\)**](#)

Creates a datagram socket

Error Index

Error Index

[**close\(\)**](#)

Close the datagram socket.

Error Index

[**finalize\(\)**](#)

Code to perform when this object is garbage collected.

Error Index

[**getLocalPort\(\)**](#)

Returns the local port that this socket is bound to.

Error Index

[**receive\(DatagramPacket\)**](#)

Receives datagram packet.

Error Index

send(DatagramPacket)

Sends Datagram Packet to the destination address

Error Index

Error Index

DatagramSocket

public DatagramSocket() throws SocketException

Creates a datagram socket

Error Index

DatagramSocket

public DatagramSocket(int port) throws SocketException

Creates a datagram socket

Parameters:

local - port to use

Error Index

Error Index

send

public void send(DatagramPacket p) throws IOException

Sends Datagram Packet to the destination address

Parameters:

DatagramPacket - to be sent. The packet contains the buffer of bytes, length and destination
InetAddress and port.

Throws: IOException

i/o error occurred

Error Index

receive

public synchronized void receive(DatagramPacket p) throws IOException

Receives datagram packet.

Parameters:

DatagramPacket - to be received. On return, the DatagramPacket contains the buffer in which the data is received, packet length, sender's address and sender's port number. Blocks until some input is available.

Throws: IOException
i/o error occurred

Error Index `getLocalPort`

```
public int getLocalPort()
```

Returns the local port that this socket is bound to.

Error Index `close`

```
public synchronized void close()
```

Close the datagram socket.

Error Index `finalize`

```
protected synchronized void finalize()
```

Code to perform when this object is garbage collected.

Overrides:

finalize in class Object

Class java.net.InetAddress

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.InetAddress

[java.lang.Object](#)
|
+----java.net.InetAddress

public final class **InetAddress**
extends [Object](#)

A class that represents Internet addresses.

Error Index

Error Index

[equals](#)(Object)

Compares this object against the specified object.

Error Index

[getAddress](#)()

Returns the raw IP address in network byte order.

Error Index

[getAllByName](#)(String)

Given a hostname, returns an array of all the corresponding InetAddresses.

Error Index

[getByName](#)(String)

Returns a network address for the indicated host.

Error Index

[getHostName](#)()

Gets the hostname for this address; also the key in the hashtable.

Error Index

[getLocalHost](#)()

Returns the local host.

Error Index

[hashCode](#)()

Returns a hashcode for this InetAddress.

Error Index

toString()

Converts the InetAddress to a String.

Error Index

Error Index

getHostName

```
public String getHostName()
```

Gets the hostname for this address; also the key in the hashtable. If the host is equal to null, then this address refers to any of the local machine's available network addresses.

Error Index

getAddress

```
public byte[] getAddress()
```

Returns the raw IP address in network byte order. The highest order byte position is in addr[0]. To be prepared for 64-bit IP addresses an array of bytes is returned.

Returns:

raw IP address in network byte order.

Error Index

hashCode

```
public int hashCode()
```

Returns a hashcode for this InetAddress.

Overrides:

hashCode in class Object

Error Index

equals

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

Compares this object against the specified object.

Parameters:

obj - the object to compare against.

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index toString

public String toString()

Converts the InetAddress to a String.

Overrides:

toString in class Object

Error Index getByName

public static synchronized InetAddress getByName(String host) throws UnknownHostException

Returns a network address for the indicated host. A host name of null refers to default address for the local machine. A local cache is used to speed access to addresses. If all addresses for a host are needed, use the getAllByName() method.

Parameters:

host - the specified host

Throws: UnknownHostException

If the address is unknown.

Error Index getAllByName

public static synchronized InetAddress[] getAllByName(String host) throws UnknownHostException

Given a hostname, returns an array of all the corresponding InetAddresss.

Throws: UnknownHostException

If the host name could not be resolved

Error Index getLocalHost

public static InetAddress getLocalHost() throws UnknownHostException

Returns the local host.

Throws: [UnknownHostException](#)

If the host name could not be resolved

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class `java.net.MalformedURLException`

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class `java.net.MalformedURLException`

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.io.IOException
|
+----java.net.MalformedURLException
```

```
public class MalformedURLException
    extends IOException
```

Signals that a malformed URL has occurred.

Error Index

Error Index

MalformedURLException()

Constructs a `MalformedURLException` with no detail message.

Error Index

MalformedURLException(String)

Constructs a `MalformedURLException` with the specified detail message.

Error Index

Error Index

MalformedURLException

```
public MalformedURLException()
```

Constructs a `MalformedURLException` with no detail message. A detail message is a `String` that describes this particular exception.

Error Index

MalformedURLException

public MalformedURLException(String msg)

Constructs a MalformedURLException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

msg - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.ProtocolException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.ProtocolException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
                  |
                  +----java.net.ProtocolException
```

```
public class ProtocolException
    extends IOException
```

Signals when connect gets an EPROTO. This exception is specifically caught in class Socket.

Error Index

Error Index

ProtocolException(String)

Constructs a new ProtocolException with the specified detail message.

Error Index

ProtocolException()

Constructs a new ProtocolException with no detail message.

Error Index

Error Index

ProtocolException

```
public ProtocolException(String host)
```

Constructs a new ProtocolException with the specified detail message. A detail message is a String that gives a specific description of this error.

Parameters:

host - the detail message

Error Index **ProtocolException**

public ProtocolException()

Constructs a new ProtocolException with no detail message. A detail message is a String that gives a specific description of this error.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.ServerSocket

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.ServerSocket

[java.lang.Object](#)

|

+----java.net.ServerSocket

public final class **ServerSocket**
extends [Object](#)

The server Socket class. It uses a SocketImpl to implement the actual socket operations. It is done this way so that you are able to change socket implementations depending on the kind of firewall being used. You can change socket implementations by setting the SocketImplFactory.

Error Index

Error Index

[ServerSocket](#)(int)

Creates a server socket on a specified port.

Error Index

[ServerSocket](#)(int, int)

Creates a server socket, binds it to the specified local port and listens to it.

Error Index

Error Index

[accept](#)()

Accepts a connection.

Error Index

[close](#)()

Closes the server socket.

Error Index

[getInetAddress](#)()

Gets the address to which the socket is connected.

Error Index

getLocalPort()

Gets the port on which the socket is listening.

Error Index

setSocketFactory(SocketImplFactory)

Sets the system's server SocketImplFactory.

Error Index

toString()

Returns the implementation address and implementation port of this ServerSocket as a String.

Error Index

Error Index

ServerSocket

public ServerSocket(int port) throws IOException

Creates a server socket on a specified port.

Parameters:

port - the port

Throws: IOException

IO error when opening the socket.

Error Index

ServerSocket

public ServerSocket(int port,
int count) throws IOException

Creates a server socket, binds it to the specified local port and listens to it. You can connect to an anonymous port by specifying the port number to be 0.

Parameters:

port - the specified port

count - the amount of time to listen for a connection

Error Index

Error Index

getInetAddress

public InetAddress getInetAddress()

Gets the address to which the socket is connected.

Error Index **getLocalPort**

```
public int getLocalPort()
```

Gets the port on which the socket is listening.

Error Index **accept**

```
public Socket accept() throws IOException
```

Accepts a connection. This method will block until the connection is made.

Throws: IOException

IO error when waiting for the connection.

Error Index **close**

```
public void close() throws IOException
```

Closes the server socket.

Throws: IOException

IO error when closing the socket.

Error Index **toString**

```
public String toString()
```

Returns the implementation address and implementation port of this ServerSocket as a String.

Overrides:

toString in class Object

Error Index **setSocketFactory**

```
public static synchronized void setSocketFactory(SocketImplFactory fac) throws IOException
```

Sets the system's server SocketImplFactory. The factory can be specified only once.

Parameters:

fac - the desired factory

Throws: SocketException

If the factory has already been defined.

Throws: IOException

IO error when setting the socket factor.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.Socket

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.Socket

[java.lang.Object](#)

|

+----java.net.Socket

public final class **Socket**
extends [Object](#)

The client Socket class. It uses a SocketImpl to implement the actual socket operations. It is done this way so that you are able to change socket implementations depending on the kind of firewall that is used. You can change socket implementations by setting the SocketImplFactory.

Error Index

Error Index

[Socket](#)(String, int)

Creates a stream socket and connects it to the specified port on the specified host.

Error Index

[Socket](#)(String, int, boolean)

Creates a socket and connects it to the specified port on the specified host.

Error Index

[Socket](#)(InetAddress, int)

Creates a stream socket and connects it to the specified address on the specified port.

Error Index

[Socket](#)(InetAddress, int, boolean)

Creates a socket and connects it to the specified address on the specified port.

Error Index

Error Index

[close](#)()

Closes the socket.

Error Index

getInetAddress()

Gets the address to which the socket is connected.

Error Index

getInputStream()

Gets an InputStream for this socket.

Error Index

getLocalPort()

Gets the local port to which the socket is connected.

Error Index

getOutputStream()

Gets an OutputStream for this socket.

Error Index

getPort()

Gets the remote port to which the socket is connected.

Error Index

setSocketImplFactory(SocketImplFactory)

Sets the system's client SocketImplFactory.

Error Index

toString()

Converts the Socket to a String.

Error Index

Error Index

Socket

```
public Socket(String host,  
             int port) throws UnknownHostException, IOException
```

Creates a stream socket and connects it to the specified port on the specified host.

Parameters:

host - the host

port - the port

Error Index

Socket

```
public Socket(String host,  
             int port,  
             boolean stream) throws IOException
```

Creates a socket and connects it to the specified port on the specified host. The last argument lets

you specify whether you want a stream or datagram socket.

Parameters:

host - the specified host

port - the specified port

stream - a boolean indicating whether this is a stream or datagram socket

Error Index Socket

```
public Socket(InetAddress address,  
             int port) throws IOException
```

Creates a stream socket and connects it to the specified address on the specified port.

Parameters:

address - the specified address

port - the specified port

Error Index Socket

```
public Socket(InetAddress address,  
             int port,  
             boolean stream) throws IOException
```

Creates a socket and connects it to the specified address on the specified port. The last argument lets you specify whether you want a stream or datagram socket.

Parameters:

address - the specified address

port - the specified port

stream - a boolean indicating whether this is a stream or datagram socket

Error Index

Error Index getInetAddress

```
public InetAddress getInetAddress()
```

Gets the address to which the socket is connected.

Error Index getPort

```
public int getPort()
```

Gets the remote port to which the socket is connected.

Error Index `getLocalPort`

```
public int getLocalPort()
```

Gets the local port to which the socket is connected.

Error Index `getInputStream`

```
public InputStream getInputStream() throws IOException
```

Gets an InputStream for this socket.

Error Index `getOutputStream`

```
public OutputStream getOutputStream() throws IOException
```

Gets an OutputStream for this socket.

Error Index `close`

```
public synchronized void close() throws IOException
```

Closes the socket.

Error Index `toString`

```
public String toString()
```

Converts the Socket to a String.

Overrides:

`toString` in class Object

Error Index `setSocketImplFactory`

public static synchronized void setSocketImplFactory([SocketImplFactory](#) fac) throws [IOException](#)

Sets the system's client SocketImplFactory. The factory can be specified only once.

Parameters:

fac - the desired factory

Throws: [SocketException](#)

If the factory is already defined.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.SocketException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.SocketException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
                  |
                  +----java.net.SocketException
```

```
public class SocketException
    extends IOException
```

Signals that an error occurred while attempting to use a socket.

Error Index

Error Index

[SocketException](#)(String)

Constructs a new SocketException with the specified detail message.

Error Index

[SocketException](#)()

Constructs a new SocketException with no detail message.

Error Index

Error Index

SocketException

```
public SocketException(String msg)
```

Constructs a new SocketException with the specified detail message. A detail message is a String that gives a specific description of this error.

Parameters:

msg - the detail message

Error Index **SocketException**

```
public SocketException()
```

Constructs a new SocketException with no detail message. A detail message is a String that gives a specific description of this error.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.SocketImpl

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.SocketImpl

[java.lang.Object](#)

|

+----java.net.SocketImpl

public class **SocketImpl**
extends [Object](#)

This is the Socket implementation class. It is an abstract class that must be subclassed to provide an actual implementation.

Error Index

Error Index

address

The internet address where the socket will make a connection.

Error Index

fd

The file descriptor object

Error Index

localport

Error Index

port

The port where the socket will make a connection.

Error Index

Error Index

SocketImpl()

Error Index

Error Index

Accepts a connection.

accept(SocketImpl)

Error Index

available()

Returns the number of bytes that can be read without blocking.

Error Index

bind(InetAddress, int)

Binds the socket to the specified port on the specified host.

Error Index

close()

Closes the socket.

Error Index

connect(String, int)

Connects the socket to the specified port on the specified host.

Error Index

connect(InetAddress, int)

Connects the socket to the specified address on the specified port.

Error Index

create(boolean)

Creates a socket with a boolean that specifies whether this is a stream socket or a datagram socket.

Error Index

getFileDescriptor()

Error Index

getInetAddress()

Error Index

getInputStream()

Gets an InputStream for this socket.

Error Index

getLocalPort()

Error Index

getOutputStream()

Gets an OutputStream for this socket.

Error Index

getPort()

Error Index

listen(int)

Listens for connections over a specified amount of time.

Error Index

toString()

Returns the address and port of this Socket as a String.

Error Index

Error Index_{fd}

protected FileDescriptor fd

The file descriptor object

Error Index_{address}

protected InetAddress address

The internet address where the socket will make a connection.

Error Index_{port}

protected int port

The port where the socket will make a connection.

Error Index_{localport}

protected int localport

Error Index

Error Index_{SocketImpl}

public SocketImpl()

Error Index

Error Index_{create}

protected abstract void create(boolean stream) throws IOException

Creates a socket with a boolean that specifies whether this is a stream socket or a datagram socket.

Parameters:

stream - a boolean indicating whether this is a stream or datagram socket

Error Index connect

protected abstract void connect(String host,
int port) throws IOException

Connects the socket to the specified port on the specified host.

Parameters:

host - the specified host of the connection

port - the port where the connection is made

Error Index connect

protected abstract void connect(InetAddress address,
int port) throws IOException

Connects the socket to the specified address on the specified port.

Parameters:

address - the specified address of the connection

port - the specified port where connection is made

Error Index bind

protected abstract void bind(InetAddress host,
int port) throws IOException

Binds the socket to the specified port on the specified host.

Parameters:

host - the host

port - the port

Error Index listen

protected abstract void listen(int count) throws IOException

Listens for connections over a specified amount of time.

Parameters:

count - the amount of time this socket will listen for connections

Error Index accept

protected abstract void accept(SocketImpl s) throws IOException

Accepts a connection.

Parameters:

s - the accepted connection

Error Index getInputStream

protected abstract InputStream getInputStream() throws IOException

Gets an InputStream for this socket.

Error Index getOutputStream

protected abstract OutputStream getOutputStream() throws IOException

Gets an OutputStream for this socket.

Error Index available

protected abstract int available() throws IOException

Returns the number of bytes that can be read without blocking.

Error Index close

protected abstract void close() throws IOException

Closes the socket.

Error Index `getFileDescriptor`

protected FileDescriptor getFileDescriptor()

Error Index `getInetAddress`

protected InetAddress getInetAddress()

Error Index `getPort`

protected int getPort()

Error Index `getLocalPort`

protected int getLocalPort()

Error Index `toString`

public String toString()

Returns the address and port of this Socket as a String.

Overrides:

toString in class Object

Interface java.net.SocketImplFactory

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.net.SocketImplFactory

public interface **SocketImplFactory**
extends [Object](#)

This interface defines a factory for SocketImpl instances. It is used by the socket class to create socket implementations that implement various policies.

Error Index

Error Index

[createSocketImpl\(\)](#)

Creates a new SocketImpl instance.

Error Index

Error Index

[createSocketImpl](#)

public abstract [SocketImpl](#) createSocketImpl()

Creates a new SocketImpl instance.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.UnknownHostException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.UnknownHostException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.io.IOException
                  |
                  +----java.net.UnknownHostException
```

```
public class UnknownHostException
    extends IOException
```

Signals that the address of the server specified by a network client could not be resolved.

Error Index

Error Index

[UnknownHostException](#)(String)

Constructs a new UnknownHostException with the specified detail message.

Error Index

[UnknownHostException](#)()

Constructs a new UnknownHostException with no detail message.

Error Index

Error Index

[UnknownHostException](#)

```
public UnknownHostException(String host)
```

Constructs a new UnknownHostException with the specified detail message. A detail message is a String that gives a specific description of this error.

Parameters:

host - the detail message

Error Index **UnknownHostException**

public UnknownHostException()

Constructs a new UnknownHostException with no detail message. A detail message is a String that gives a specific description of this error.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.UnknownServiceException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.UnknownServiceException

```
java.lang.Object
|
+----java.lang.Throwable
|
+----java.lang.Exception
|
+----java.io.IOException
|
+----java.net.UnknownServiceException
```

```
public class UnknownServiceException
extends IOException
```

Signals that an unknown service exception has occurred.

Error Index

Error Index

[UnknownServiceException\(\)](#)

Constructs a new UnknownServiceException with no detail message.

Error Index

[UnknownServiceException\(String\)](#)

Constructs a new UnknownServiceException with the specified detail message.

Error Index

Error Index

UnknownServiceException

```
public UnknownServiceException()
```

Constructs a new UnknownServiceException with no detail message. A detail message is a String that gives a specific description of this error.

Error Index

UnknownServiceException

public UnknownServiceException(String msg)

Constructs a new UnknownServiceException with the specified detail message. A detail message is a String that gives a specific description of this error.

Parameters:

msg - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URL

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URL

[java.lang.Object](#)

|

+----java.net.URL

public final class **URL**
extends [Object](#)

Class URL represents a Uniform Reference Locator -- a reference to an object on the World Wide Web. This is a constant object, once it is created, its fields cannot be changed.

Error Index

Error Index

[URL](#)(String, String, int, String)

Creates an absolute URL from the specified protocol, host, port and file.

Error Index

[URL](#)(String, String, String)

Creates an absolute URL from the specified protocol, host, and file.

Error Index

[URL](#)(String)

Creates a URL from the unparsed absolute URL.

Error Index

[URL](#)(URL, String)

Creates a URL from the unparsed URL in the specified context. If spec is an absolute URL it is used as is.

Error Index

Error Index

[equals](#)(Object)

Compares two URLs.

Error Index

getContent()

Gets the contents from this opened connection.

Error Index

getFile()

Gets the file name.

Error Index

getHost()

Gets the host name.

Error Index

getPort()

Gets the port number.

Error Index

getProtocol()

Gets the protocol name.

Error Index

getRef()

Gets the ref.

Error Index

hashCode()

Creates an integer suitable for hash table indexing.

Error Index

openConnection()

Creates (if not already in existence) a `URLConnection` object that contains a connection to the remote object referred to by the URL.

Error Index

openStream()

Opens an input stream.

Error Index

sameFile(URL)

Compares two URLs, excluding the "ref" fields: `sameFile` is true if the true references the same remote object, but not necessarily the same subpiece of that object.

Error Index

set(String, String, int, String, String)

Sets the fields of the URL.

Error Index

setURLStreamHandlerFactory(URLStreamHandlerFactory)

Sets the `URLStreamHandler` factory.

Error Index

toExternalForm()

Reverses the parsing of the URL.

Error Index

toString()

Converts to a human-readable form.

Error Index

Error Index URL

public URL(String protocol,
 String host,
 int port,
 String file) throws MalformedURLException

Creates an absolute URL from the specified protocol, host, port and file.

Parameters:

protocol - the protocol to use
host - the host to connect to
port - the port at that host to connect to
file - the file on that host

Throws: MalformedURLException
If an unknown protocol is found.

Error Index URL

public URL(String protocol,
 String host,
 String file) throws MalformedURLException

Creates an absolute URL from the specified protocol, host, and file. The port number used will be the default for the protocol.

Parameters:

protocol - the protocol to use
host - the host to connect to
file - the file on that host

Throws: MalformedURLException
If an unknown protocol is found.

Error Index URL

public URL(String spec) throws MalformedURLException

Creates a URL from the unparsed absolute URL.

Parameters:

spec - the URL String to parse

Error Index URL

public URL(URL context,
String spec) throws MalformedURLException

Creates a URL from the unparsed URL in the specified context. If spec is an absolute URL it is used as is. Otherwise it is parsed in terms of the context. Context may be null (indicating no context).

Parameters:

context - the context to parse the URL to

spec - the URL String to parse

Throws: MalformedURLException

If the protocol is equal to null.

Error Index

Error Index set

protected void set(String protocol,
String host,
int port,
String file,
String ref)

Sets the fields of the URL. This is not a public method so that only URLStreamHandlers can modify URL fields. URLs are otherwise constant. REMIND: this method will be moved to URLStreamHandler

Parameters:

protocol - the protocol to use

host - the host name to connect to

port - the protocol port to connect to

file - the specified file name on that host

ref - the reference

Error Index getPort

public int getPort()

Gets the port number. Returns -1 if the port is not set.

Error IndexgetProtocol

```
public String getProtocol()
```

Gets the protocol name.

Error IndexgetHost

```
public String getHost()
```

Gets the host name.

Error IndexgetFile

```
public String getFile()
```

Gets the file name.

Error IndexgetRef

```
public String getRef()
```

Gets the ref.

Error Indexequals

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

Compares two URLs.

Parameters:

obj - the URL to compare against.

Returns:

true if and only if they are equal, false otherwise.

Overrides:

equals in class Object

Error IndexhashCode

```
public int hashCode()
```

Creates an integer suitable for hash table indexing.

Overrides:

hashCode in class Object

Error Index sameFile

public boolean sameFile(URL other)

Compares two URLs, excluding the "ref" fields: sameFile is true if the true references the same remote object, but not necessarily the same subpiece of that object.

Parameters:

other - the URL to compare against.

Returns:

true if and only if they are equal, false otherwise.

Error Index toString

public String toString()

Converts to a human-readable form.

Returns:

the textual representation.

Overrides:

toString in class Object

Error Index toExternalForm

public String toExternalForm()

Reverses the parsing of the URL.

Returns:

the textual representation of the fully qualified URL (i.e. after the context and canonicalization have been applied).

Error Index openConnection

public URLConnection openConnection() throws IOException

Creates (if not already in existence) a URLConnection object that contains a connection to the remote object referred to by the URL. Invokes the appropriate protocol handler. Failure is indicated by throwing an exception.

Throws: IOException

If an I/O exception has occurred.

See Also:

URLConnection

Error Index

openStream

public final InputStream openStream() throws IOException

Opens an input stream.

Throws: IOException

If an I/O exception has occurred.

Error Index

getContent

public final Object getContent() throws IOException

Gets the contents from this opened connection.

Throws: IOException

If an I/O exception has occurred.

Error Index

setURLStreamHandlerFactory

public static synchronized void setURLStreamHandlerFactory(URLStreamHandlerFactory fac)

Sets the URLStreamHandler factory.

Parameters:

fac - the desired factory

Throws: Error

If the factory has already been defined.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URLConnection

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URLConnection

[java.lang.Object](#)

|

+----java.net.URLConnection

```
public class URLConnection
    extends Object
```

A class to represent an active connection to an object represented by a URL. It is an abstract class that must be subclassed to implement a connection.

Error Index

Error Index

[allowUserInteraction](#)

Error Index

[connected](#)

Error Index

[doInput](#)

Error Index

[doOutput](#)

Error Index

[ifModifiedSince](#)

Error Index

[url](#)

Error Index

[useCaches](#)

Error Index

Error Index

URLConnection(URL)

Constructs a URL connection to the specified URL.

Error Index

Error Index

connect()

URLConnection objects go through two phases: first they are created, then they are connected.

Error Index

getAllowUserInteraction()

Error Index

getContent()

Gets the object referred to by this URL.

Error Index

getContentEncoding()

Gets the content encoding.

Error Index

getContentLength()

Gets the content length.

Error Index

getContentType()

Gets the content type.

Error Index

getDate()

Gets the sending date of the object.

Error Index

getDefaultAllowUserInteraction()

Error Index

getDefaultRequestProperty (String)

Error Index

getDefaultUseCaches()

Sets/gets the default value of the UseCaches flag.

Error Index

getDoInput()

Error Index

getDoOutput()

Error Index

getExpiration()

Gets the expiration date of the object.

Error Index

getHeaderField(String)

Gets a header field by name.

Error Index

getHeaderField(int)

Returns the value for the nth header field.

Error Index

getHeaderFieldDate(String, long)

Gets a header field by name.

Error Index

getHeaderFieldInt(String, int)

Gets a header field by name.

Error Index

getHeaderFieldKey(int)

Returns the key for the nth header field.

Error Index

getIfModifiedSince()

Error Index

getInputStream()

Calls this routine to get an InputStream that reads from the object.

Error Index

getLastModified()

Gets the last modified date of the object.

Error Index

getOutputStream()

Calls this routine to get an OutputStream that writes to the object.

Error Index

getRequestProperty(String)

Error Index

getURL()

Gets the URL for this connection.

Error Index

getUseCaches()

Error Index

guessContentTypeFromName (String)

A useful utility routine that tries to guess the content-type of an object based upon its extension.

Error Index

guessContentTypeFromStream(InputStream)

This method is used to check for files that have some type that can be determined by inspection.

Error Index

setAllowUserInteraction(boolean)

Some URL connections occasionally need to to interactions with the user.

Error Index

setContentHandlerFactory(ContentHandlerFactory)

Sets the ContentHandler factory.

Error Index

setDefaultAllowUserInteraction(boolean)

Sets/gets the default value of the allowUserInteraction flag.

Error Index

setDefaultRequestProperty(String, String)

Sets/gets the default value of a general request property.

Error Index

setDefaultUseCaches(boolean)

Error Index

setDoInput(boolean)

A URL connection can be used for input and/or output.

Error Index

setDoOutput(boolean)

A URL connection can be used for input and/or output.

Error Index

setIfModifiedSince(long)

Some protocols support skipping fetching unless the object is newer than some amount of time.

Error Index

setRequestProperty(String, String)

Sets/gets a general request property.

Error Index

setUseCaches(boolean)

Some protocols do caching of documents.

Error Index

toString()

Returns the String representation of the URL connection.

Error Index

Error Index_{url}

protected URL url

Error Index_{doInput}

protected boolean doInput

Error Index_{doOutput}

protected boolean doOutput

Error Index allowUserInteraction

protected boolean allowUserInteraction

Error Index useCaches

protected boolean useCaches

Error Index ifModifiedSince

protected long ifModifiedSince

Error Index connected

protected boolean connected

Error Index

Error Index URLConnection

protected URLConnection(URL url)

Constructs a URL connection to the specified URL.

Parameters:

url - the specified URL

Error Index

Error Index connect

public abstract void connect() throws IOException

URLConnection objects go through two phases: first they are created, then they are connected. After being created, and before being connected, various options can be specified (eg. doInput,

UseCaches, ...). After connecting, it is an Error to try to set them. Operations that depend on being connected, like getContentLength, will implicitly perform the connection if necessary. Connecting when already connected does nothing.

Error Index **getURL**

```
public URL getURL()
```

Gets the URL for this connection.

Error Index **getContentLength**

```
public int getContentLength()
```

Gets the content length. Returns -1 if not known.

Error Index **getContentType**

```
public String getContentType()
```

Gets the content type. Returns null if not known.

Error Index **getContentEncoding**

```
public String getContentEncoding()
```

Gets the content encoding. Returns null if not known.

Error Index **getExpiration**

```
public long getExpiration()
```

Gets the expiration date of the object. Returns 0 if not known.

Error Index **getDate**

```
public long getDate()
```

Gets the sending date of the object. Returns 0 if not known.

Error Index **getLastModified**

```
public long getLastModified()
```

Gets the last modified date of the object. Returns 0 if not known.

Error Index **getHeaderField**

```
public String getHeaderField(String name)
```

Gets a header field by name. Returns null if not known.

Parameters:

name - the name of the header field

Error Index **getHeaderFieldInt**

```
public int getHeaderFieldInt(String name,  
                             int Default)
```

Gets a header field by name. Returns null if not known. The field is parsed as an integer. This form of getHeaderField exists because some connection types (e.g. http-ng) have pre-parsed headers and this allows them to override this method and short-circuit the parsing.

Parameters:

name - the name of the header field

Default - the value to return if the field is missing or malformed.

Error Index **getHeaderFieldDate**

```
public long getHeaderFieldDate(String name,  
                               long Default)
```

Gets a header field by name. Returns null if not known. The field will be parsed as a date. This form of getHeaderField exists because some connection types (eg. http-ng) have pre-parsed headers. This allows them to override this method and short-circuit the parsing.

Parameters:

name - the name of the header field

Default - the value to return if the field is missing or malformed.

Error Index `getHeaderFieldKey`

public String getHeaderFieldKey(int n)

Returns the key for the nth header field. Returns null if there are fewer than n fields. This can be used to iterate through all the headers in the message.

Error Index `getHeaderField`

public String getHeaderField(int n)

Returns the value for the nth header field. Returns null if there are fewer than n fields. This can be used in conjunction with getHeaderFieldKey to iterate through all the headers in the message.

Error Index `getContent`

public Object getContent() throws IOException

Gets the object referred to by this URL. For example, if it refers to an image the object will be some subclass of Image. The instanceof operator should be used to determine what kind of object was returned.

Returns:

the object that was fetched.

Throws: UnknownServiceException

If the protocol does not support content.

Error Index `getInputStream`

public InputStream getInputStream() throws IOException

Calls this routine to get an InputStream that reads from the object. Protocol implementors should use this if appropriate.

Throws: UnknownServiceException

If the protocol does not support input.

Error Index `getOutputStream`

public OutputStream getOutputStream() throws IOException

Calls this routine to get an OutputStream that writes to the object. Protocol implementors should use this if appropriate.

Throws: UnknownServiceException

If the protocol does not support output.

Error Index toString

```
public String toString()
```

Returns the String representation of the URL connection.

Overrides:

toString in class Object

Error Index setDoInput

```
public void setDoInput(boolean doinput)
```

A URL connection can be used for input and/or output. Set the DoInput flag to true if you intend to use the URL connection for input, false if not. The default is true unless DoOutput is explicitly set to true, in which case DoInput defaults to false.

Error Index getDoInput

```
public boolean getDoInput()
```

Error Index setDoOutput

```
public void setDoOutput(boolean dooutput)
```

A URL connection can be used for input and/or output. Set the DoOutput flag to true if you intend to use the URL connection for output, false if not. The default is false.

Error Index getDoOutput

```
public boolean getDoOutput()
```

Error Index setAllowUserInteraction

```
public void setAllowUserInteraction(boolean allowuserinteraction)
```

Some URL connections occasionally need to to interactions with the user. For example, the http protocol may need to pop up an authentication dialog. But this is only appropriate if the application is running in a context where there *is* a user. The allowUserInteraction flag allows these interactions when true. When it is false, they are not allowed and an exception is tossed. The default value can be set/gotten using setDefaultAllowUserInteraction, which defaults to false.

Error Index

getAllowUserInteraction

```
public boolean getAllowUserInteraction()
```

Error Index

setDefaultAllowUserInteraction

```
public static void setDefaultAllowUserInteraction(boolean defaultallowuserinteraction)
```

Sets/gets the default value of the allowUserInteraction flag. This default is "sticky", being a part of the static state of all URLConnections. This flag applies to the next, and all following URLConnections that are created.

Error Index

getDefaultAllowUserInteraction

```
public static boolean getDefaultAllowUserInteraction()
```

Error Index

setUseCaches

```
public void setUseCaches(boolean usecaches)
```

Some protocols do caching of documents. Occasionally, it is important to be able to "tunnel through" and ignore the caches (e.g. the "reload" button in a browser). If the UseCaches flag on a connection is true, the connection is allowed to use whatever caches it can. If false, caches are to be ignored. The default value comes from DefaultUseCaches, which defaults to true.

Error Index

getUseCaches

```
public boolean getUseCaches()
```

Error Index

setIfModifiedSince

```
public void setIfModifiedSince(long ifmodifiedsince)
```

Some protocols support skipping fetching unless the object is newer than some amount of time.
The ifModifiedSince field may be set/gotten to define this time.

Error Index `getIfModifiedSince`

```
public long getIfModifiedSince()
```

Error Index `getDefaultUseCaches`

```
public boolean getDefaultUseCaches()
```

Sets/gets the default value of the UseCaches flag. This default is "sticky", being a part of the static state of all URLConnections. This flag applies to the next, and all following, URLConnections that are created.

Error Index `setDefaultUseCaches`

```
public void setDefaultUseCaches(boolean defaultusecaches)
```

Error Index `setRequestProperty`

```
public void setRequestProperty(String key,  
                               String value)
```

Sets/gets a general request property.

Parameters:

key - The keyword by which the request is known (eg "accept")

value - The value associated with it.

Error Index `getRequestProperty`

```
public String getRequestProperty(String key)
```

Error Index `setDefaultRequestProperty`

```
public static void setDefaultRequestProperty(String key,
```

String value)

Sets/gets the default value of a general request property. When a URLConnection is created, it is initialized with these properties.

Parameters:

key - The keyword by which the request is known (eg "accept")

value - The value associated with it.

Error Index

getDefaultRequestProperty

public static String getDefaultRequestProperty(String key)

Error Index

setContentHandlerFactory

public static synchronized void setContentHandlerFactory(ContentHandlerFactory fac)

Sets the ContentHandler factory.

Parameters:

fac - the desired factory

Throws: Error

If the factory has already been defined.

Error Index

guessContentTypeFromName

protected static String guessContentTypeFromName(String fname)

A useful utility routine that tries to guess the content-type of an object based upon its extension.

Error Index

guessContentTypeFromStream

protected static String guessContentTypeFromStream(InputStream is) throws IOException

This method is used to check for files that have some type that can be determined by inspection. The bytes at the beginning of the file are examined loosely. In an ideal world, this routine would not be needed, but in a world where http servers lie about content-types and extensions are often non-standard, direct inspection of the bytes can make the system more robust. The stream must support marks (e.g. have a BufferedInputStream somewhere).

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URLEncoder

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URLEncoder

[java.lang.Object](#)

|

+----java.net.URLEncoder

```
public class URLEncoder
extends Object
```

Turns Strings of text into x-www-form-urlencoded format.

Error Index

Error Index

[encode](#)(String)

Translates String into x-www-form-urlencoded format.

Error Index

Error Index

encode

```
public static String encode(String s)
```

Translates String into x-www-form-urlencoded format.

Parameters:

s - String to be translated

Returns:

the translated String.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URLStreamHandler

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.net.URLStreamHandler

[java.lang.Object](#)

|

+----java.net.URLStreamHandler

public class **URLStreamHandler**
extends [Object](#)

Abstract class for URL stream openers. Subclasses of this class know how to create streams for particular protocol types.

Error Index

Error Index

URLStreamHandler()

Error Index

Error Index

openConnection(URL)

Opens an input stream to the object referenced by the URL.

Error Index

parseURL(URL, String, int, int)

This method is called to parse the string spec into URL u.

Error Index

setURL(URL, String, String, int, String, String)

Calls the (protected) set method out of the URL given.

Error Index

toExternalForm(URL)

Reverses the parsing of the URL.

Error Index

Error Index

URLConnection

public URLStreamHandler()

Error Index

Error Index

URLConnection

protected abstract URLConnection openConnection(URL u) throws IOException

Opens an input stream to the object referenced by the URL. This method should be overridden by a subclass.

Parameters:

u - the URL that this connects to

Error Index

URLConnection

protected void parseURL(URL u,
 String spec,
 int start,
 int limit)

This method is called to parse the string spec into URL u. If there is any inherited context then it has already been copied into u. The parameters start and limit refer to the range of characters in spec that should be parsed. The default method uses parsing rules that match the http spec, which most URL protocol families follow. If you are writing a protocol handler that has a different syntax, override this routine.

Parameters:

u - the URL to receive the result of parsing the spec

spec - the URL string to parse

start - the character position to start parsing at. This is just past the ':' (if there is one).

limit - the character position to stop parsing at. This is the end of the string or the position of the "#" character if present (the "#" reference syntax is protocol independent).

Error Index

URLConnection

protected String toExternalForm(URL u)

Reverses the parsing of the URL. This should probably be overridden if you override `parseURL()`.

Parameters:

u - the URL

Returns:

the textual representation of the fully qualified URL (i.e. after the context and canonicalization have been applied).

Error Index `setURL`

```
protected void setURL(URL u,  
                     String protocol,  
                     String host,  
                     int port,  
                     String file,  
                     String ref)
```

Calls the (protected) `set` method out of the URL given. Only classes derived from `URLStreamHandler` are supposed to be able to call the `set()` method on a URL.

See Also:

[set](#)

Interface

java.net.URLStreamHandlerFactory

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.net.URLStreamHandlerFactory

public interface **URLStreamHandlerFactory**
extends [Object](#)

This interface defines a factory for URLStreamHandler instances. It is used by the URL class to create URLStreamHandlers for various streams.

Error Index

Error Index

[createURLStreamHandler](#)(String)

Creates a new URLStreamHandler instance with the specified protocol.

Error Index

Error Index

[createURLStreamHandler](#)

public abstract [URLStreamHandler](#) [createURLStreamHandler](#)([String](#) protocol)

Creates a new URLStreamHandler instance with the specified protocol.

Parameters:

protocol - the protocol to use (ftp, http, nntp, etc.)

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.BitSet

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.BitSet

[java.lang.Object](#)

|

+----java.util.BitSet

public final class **BitSet**
extends [Object](#)
implements [Cloneable](#)

A set of bits. The set automatically grows as more bits are needed.

Error Index

Error Index

BitSet()

Creates an empty set.

Error Index

BitSet(int)

Creates an empty set with the specified size.

Error Index

Error Index

and(BitSet)

Logically ANDs this bit set with the specified set of bits.

Error Index

clear(int)

Clears a bit.

Error Index

clone()

Clones the BitSet.

Error Index

equals(Object)

Compares this object against the specified object.

Error Index

get(int)

Gets a bit.

Error Index

hashCode()

Gets the hashCode.

Error Index

or(BitSet)

Logically ORs this bit set with the specified set of bits.

Error Index

set(int)

Sets a bit.

Error Index

size()

Calculates and returns the set's size

Error Index

toString()

Converts the BitSet to a String.

Error Index

xor(BitSet)

Logically XORs this bit set with the specified set of bits.

Error Index

Error Index

BitSet

public BitSet()

Creates an empty set.

Error Index

BitSet

public BitSet(int nbits)

Creates an empty set with the specified size.

Parameters:

nbits - the size of the set

Error Index

Error Index set

```
public void set(int bit)
```

Sets a bit.

Parameters:

bit - the bit to be set

Error Index clear

```
public void clear(int bit)
```

Clears a bit.

Parameters:

bit - the bit to be cleared

Error Index get

```
public boolean get(int bit)
```

Gets a bit.

Parameters:

bit - the bit to be gotten

Error Index and

```
public void and(BitSet set)
```

Logically ANDs this bit set with the specified set of bits.

Parameters:

set - the bit set to be ANDed with

Error Index_{or}

public void or(BitSet set)

Logically ORs this bit set with the specified set of bits.

Parameters:

set - the bit set to be ORed with

Error Index_{xor}

public void xor(BitSet set)

Logically XORs this bit set with the specified set of bits.

Parameters:

set - the bit set to be XORed with

Error Index_{hashCode}

public int hashCode()

Gets the hashcode.

Overrides:

hashCode in class Object

Error Index_{size}

public int size()

Calculates and returns the set's size

Error Index_{equals}

public boolean equals(Object obj)

Compares this object against the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index clone

public Object clone()

Clones the BitSet.

Overrides:

clone in class Object

Error Index toString

public String toString()

Converts the BitSet to a String.

Overrides:

toString in class Object

Class java.util.Date

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Date

[java.lang.Object](#)

|
+----java.util.Date

```
public class Date  
extends Object
```

A wrapper for a date. This class lets you manipulate dates in a system independent way. To print today's date use:

```
Date d = new Date();  
System.out.println("today = " + d);
```

To find out what day corresponds to a particular date:

```
Date d = new Date(63, 0, 16); // January 16, 1963  
System.out.println("Day of the week: " + d.getDay());
```

The date can be set and examined according to the local time zone into the year, month, day, hour, minute and second.

While the API is intended to reflect UTC, Coordinated Universal Time, it doesn't do so exactly. This inexact behavior is inherited from the time system of the underlying OS. All modern OS's that I (jag) am aware of assume that 1 day = 24*60*60 seconds. In UTC, about once a year there is an extra second, called a "leap second" added to a day to account for the wobble of the earth. Most computer clocks are not accurate enough to be able to reflect this distinction. Some computer standards are defined in GMT, which is equivalent to UT, Universal Time. GMT is the "civil" name for the standard, UT is the "scientific" name for the same standard. The distinction between UTC and UT is that the first is based on an atomic clock and the second is based on astronomical observations, which for all practical purposes is an invisibly fine hair to split. An interesting source of further information is the US Naval Observatory, particularly the <http://tycho.usno.navy.mil> **Directorate of Time** and their definitions of <http://tycho.usno.navy.mil/systemtime.html> **Systems of Time**.

Error Index

Error Index

[Date\(\)](#)

Creates today's date/time.

Error Index

Creates a date.

Date(long)

Error Index

Creates a date.

Date(int, int, int)

Error Index

Creates a date.

Date(int, int, int, int, int)

Error Index

Creates a date.

Date(int, int, int, int, int, int)

Error Index

Creates a date from a string according to the syntax accepted by parse().

Date(String)

Error Index

Error Index

Calculates a UTC value from YMDHMS.

UTC(int, int, int, int, int, int)

Error Index

Checks whether this date comes after the specified date.

after(Date)

Error Index

Checks whether this date comes before the specified date.

before(Date)

Error Index

Compares this object against the specified object.

equals(Object)

Error Index

Returns the day of the month.

getDate()

Error Index

Returns the day of the week.

getDay()

Error Index

Returns the hour.

getHours()

Error Index

Returns the minute.

getMinutes()

Error Index

getMonth()

Returns the month.

Error Index

getSeconds()

Returns the second.

Error Index

getTime()

Returns the time in milliseconds since the epoch.

Error Index

getTimezoneOffset()

Return the time zone offset in minutes for the current locale that is appropriate for this time.

Error Index

getYear()

Returns the year after 1900.

Error Index

hashCode()

Computes a hashCode.

Error Index

parse(String)

Given a string representing a time, parse it and return the time value.

Error Index

setDate(int)

Sets the date.

Error Index

setHours(int)

Sets the hours.

Error Index

setMinutes(int)

Sets the minutes.

Error Index

setMonth(int)

Sets the month.

Error Index

setSeconds(int)

Sets the seconds.

Error Index

setTime(long)

Sets the time.

Error Index

setYear(int)

Sets the year.

Error Index

toGMTString()

Converts a date to a String, using the Internet GMT conventions.

Error Index

toLocaleString()

Converts a date to a String, using the locale conventions.

Error Index

toString()

Converts a date to a String, using the UNIX ctime conventions.

Error Index

Error Index

Date

```
public Date()
```

Creates today's date/time.

Error Index

Date

```
public Date(long date)
```

Creates a date. The fields are normalized before the Date object is created. The argument does not have to be in the correct range. For example, the 32nd of January is correctly interpreted as the 1st of February. You can use this to figure out what day a particular date falls on.

Parameters:

date - the value of the argument to be created

Error Index

Date

```
public Date(int year,  
            int month,  
            int date)
```

Creates a date. The fields are normalized before the Date object is created. The arguments do not have to be in the correct range. For example, the 32nd of January is correctly interpreted as the 1st of February. You can use this to figure out what day a particular date falls on.

Parameters:

year - a year after 1900

month - a month between 0-11

date - day of the month between 1-31

Error Index

Date

```
public Date(int year,
```

```
int month,  
int date,  
int hrs,  
int min)
```

Creates a date. The fields are normalized before the Date object is created. The arguments do not have to be in the correct range. For example, the 32nd of January is correctly interpreted as the 1st of February. You can use this to figure out what day a particular date falls on.

Parameters:

- year - a year after 1900
- month - a month between 0-11
- date - day of the month between 1-31
- hrs - hours between 0-23
- min - minutes between 0-59

Error Index Date

```
public Date(int year,  
int month,  
int date,  
int hrs,  
int min,  
int sec)
```

Creates a date. The fields are normalized before the Date object is created. The arguments do not have to be in the correct range. For example, the 32nd of January is correctly interpreted as the 1st of February. You can use this to figure out what day a particular date falls on.

Parameters:

- year - a year after 1900
- month - a month between 0-11
- date - day of the month between 1-31
- hrs - hours between 0-23
- min - minutes between 0-59
- sec - seconds between 0-59

Error Index Date

```
public Date(String s)
```

Creates a date from a string according to the syntax accepted by parse().

Error Index

Error Index UTC

```
public static long UTC(int year,  
    int month,  
    int date,  
    int hrs,  
    int min,  
    int sec)
```

Calculates a UTC value from YMDHMS. Interpretes the parameters in UTC, *not in the local time zone*.

Parameters:

year - a year after 1900
month - a month between 0-11
date - day of the month between 1-31
hrs - hours between 0-23
min - minutes between 0-59
sec - seconds between 0-59

Error Index parse

```
public static long parse(String s)
```

Given a string representing a time, parse it and return the time value. It accepts many syntaxes, but most importantly, it accepts the IETF standard date syntax: "Sat, 12 Aug 1995 13:30:00 GMT". It understands the continental US time zone abbreviations, but for general use, a timezone offset should be used: "Sat, 12 Aug 1995 13:30:00 GMT+0430" (4 hours, 30 minutes west of the Greenwich meridian). If no time zone is specified, the local time zone is assumed. GMT and UTC are considered equivalent.

Error Index getYear

```
public int getYear()
```

Returns the year after 1900.

Error Index setYear

```
public void setYear(int year)
```

Sets the year.

Parameters:

year - the year value

Error Index **getMonth**

```
public int getMonth()
```

Returns the month. This method assigns months with the values 0-11, with January beginning at value 0.

Error Index **setMonth**

```
public void setMonth(int month)
```

Sets the month.

Parameters:

month - the month value (0-11)

Error Index **getDate**

```
public int getDate()
```

Returns the day of the month. This method assigns days with the values of 1 to 31.

Error Index **setDate**

```
public void setDate(int date)
```

Sets the date.

Parameters:

date - the day value

Error Index **getDay**

```
public int getDay()
```

Returns the day of the week. This method assigns days of the week with the values 0-6, with 0 being Sunday.

Error Index **getHours**

```
public int getHours()
```

Returns the hour. This method assigns the value of the hours of the day to range from 0 to 23, with midnight equal to 0.

Error Index **setHours**

```
public void setHours(int hours)
```

Sets the hours.

Parameters:

hours - the hour value

Error Index **getMinutes**

```
public int getMinutes()
```

Returns the minute. This method assigns the minutes of an hour to be any value from 0 to 59.

Error Index **setMinutes**

```
public void setMinutes(int minutes)
```

Sets the minutes.

Parameters:

minutes - the value of the minutes

Error Index **getSeconds**

```
public int getSeconds()
```

Returns the second. This method assigns the seconds of a minute to values of 0-59.

Error Index **setSeconds**


```
public void setSeconds(int seconds)
```

Sets the seconds.

Parameters:

seconds - the second value

Error Index `getTime`

```
public long getTime()
```

Returns the time in milliseconds since the epoch.

Error Index `setTime`

```
public void setTime(long time)
```

Sets the time.

Parameters:

time - The new time value in milliseconds since the epoch.

Error Index `before`

```
public boolean before(Date when)
```

Checks whether this date comes before the specified date.

Parameters:

when - the date to compare

Returns:

true if the original date comes before the specified one; false otherwise.

Error Index `after`

```
public boolean after(Date when)
```

Checks whether this date comes after the specified date.

Parameters:

when - the date to compare

Returns:

true if the original date comes after the specified one; false otherwise.

Error Index equals

public boolean equals(Object obj)

Compares this object against the specified object.

Parameters:

obj - the object to compare with

Returns:

true if the objects are the same; false otherwise.

Overrides:

equals in class Object

Error Index hashCode

public int hashCode()

Computes a hashCode.

Overrides:

hashCode in class Object

Error Index toString

public String toString()

Converts a date to a String, using the UNIX ctime conventions.

Overrides:

toString in class Object

Error Index toLocaleString

public String toLocaleString()

Converts a date to a String, using the locale conventions.

Error Index toGMTString

public String toGMTString()

Converts a date to a String, using the Internet GMT conventions.

Error Index getTimezoneOffset

public int getTimezoneOffset()

Return the time zone offset in minutes for the current locale that is appropriate for this time. This value would be a constant except for daylight savings time.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Dictionary

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Dictionary

[java.lang.Object](#)

|

+----java.util.Dictionary

public class **Dictionary**

extends [Object](#)

The Dictionary class is the abstract parent of Hashtable, which maps keys to values. Any object can be used as a key and/or value.

See Also:

[Hashtable](#), [hashCode](#), [equals](#)

Error Index

Error Index

[Dictionary\(\)](#)

Error Index

Error Index

[elements\(\)](#)

Returns an enumeration of the elements.

Error Index

[get\(Object\)](#)

Gets the object associated with the specified key in the Dictionary.

Error Index

[isEmpty\(\)](#)

Returns true if the Dictionary contains no elements.

Error Index

[keys\(\)](#)

Returns an enumeration of the Dictionary's keys.

Error Index

put(Object, Object)

Puts the specified element into the Dictionary, using the specified key.

Error Index

remove(Object)

Removes the element corresponding to the key.

Error Index

size()

Returns the number of elements contained within the Dictionary.

Error Index

Error Index

Dictionary

public Dictionary()

Error Index

Error Index

size

public abstract int size()

Returns the number of elements contained within the Dictionary.

Error Index

isEmpty

public abstract boolean isEmpty()

Returns true if the Dictionary contains no elements.

Error Index

keys

public abstract Enumeration keys()

Returns an enumeration of the Dictionary's keys.

See Also:

elements, Enumeration

Error Indexelements

public abstract Enumeration elements()

Returns an enumeration of the elements. Use the Enumeration methods on the returned object to fetch the elements sequentially.

See Also:

keys, Enumeration

Error Indexget

public abstract Object get(Object key)

Gets the object associated with the specified key in the Dictionary.

Parameters:

key - the key in the hash table

Returns:

s the element for the key, or null if the key is not defined in the hash table.

See Also:

put

Error Indexput

public abstract Object put(Object key,
Object value)

Puts the specified element into the Dictionary, using the specified key. The element may be retrieved by doing a get() with the same key. The key and the element cannot be null.

Parameters:

key - the specified hashtable key

value - the specified element

Returns:

the old value of the key, or null if it did not have one.

Throws: NullPointerException

If the value of the specified element is null.

See Also:

get

Error Indexremove

public abstract [Object](#) remove([Object](#) key)

Removes the element corresponding to the key. Does nothing if the key is not present.

Parameters:

key - the key that needs to be removed

Returns:

the value of key, or null if the key was not found.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.EmptyStackException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.EmptyStackException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.util.EmptyStackException
```

```
public class EmptyStackException
    extends RuntimeException
```

Signals that the stack is empty.

See Also:

[Stack](#)

Error Index

Error Index

[EmptyStackException\(\)](#)

Constructs a new EmptyStackException with no detail message.

Error Index

Error Index

[EmptyStackException](#)

```
public EmptyStackException()
```

Constructs a new EmptyStackException with no detail message. A detail message is a String that describes the exception.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.util.Enumeration

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.util.Enumeration

public interface **Enumeration**
extends [Object](#)

The Enumeration interface specifies a set of methods that may be used to enumerate, or count through, a set of values. The enumeration is consumed by use; its values may only be counted once.

For example, to print all elements of a Vector v:

```
for (Enumeration e = v.elements() ; e.hasMoreElements() ;) {  
    System.out.println(e.nextElement());  
}
```

See Also:

[Vector](#), [Hashtable](#)

Error Index

Error Index

[hasMoreElements\(\)](#)

Returns true if the enumeration contains more elements; false if its empty.

Error Index

[nextElement\(\)](#)

Returns the next element of the enumeration.

Error Index

Error Index

[hasMoreElements](#)

public abstract boolean hasMoreElements()

Returns true if the enumeration contains more elements; false if its empty.

Error Index

nextElement

public abstract Object nextElement()

Returns the next element of the enumeration. Calls to this method will enumerate successive elements.

Throws: NoSuchElementException
If no more elements exist.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Hashtable

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Hashtable

[java.lang.Object](#)

```
|
+----java.util.Dictionary
      |
      +----java.util.Hashtable
```

```
public class Hashtable
  extends Dictionary
  implements Cloneable
```

Hashtable class. Maps keys to values. Any object can be used as a key and/or value.

To successfully store and retrieve objects from a hash table, the object used as the key must implement the hashCode() and equals() methods.

This example creates a hashtable of numbers. It uses the names of the numbers as keys:

```
Hashtable numbers = new Hashtable();
numbers.put("one", new Integer(1));
numbers.put("two", new Integer(2));
numbers.put("three", new Integer(3));
```

To retrieve a number use:

```
Integer n = (Integer)numbers.get("two");
if (n != null) {
    System.out.println("two = " + n);
}
```

See Also:

[hashCode](#), [equals](#)

Error Index

Error Index

[Hashtable](#)(int, float)

Constructs a new, empty hashtable with the specified initial capacity and the specified load factor.

Error Index

Hashtable(int)

Constructs a new, empty hashtable with the specified initial capacity.

Error Index

Hashtable()

Constructs a new, empty hashtable.

Error Index

Error Index

clear()

Clears the hash table so that it has no more elements in it.

Error Index

clone()

Creates a clone of the hashtable.

Error Index

contains(Object)

Returns true if the specified object is an element of the hashtable.

Error Index

containsKey(Object)

Returns true if the collection contains an element for the key.

Error Index

elements()

Returns an enumeration of the elements.

Error Index

get(Object)

Gets the object associated with the specified key in the hashtable.

Error Index

isEmpty()

Returns true if the hashtable contains no elements.

Error Index

keys()

Returns an enumeration of the hashtable's keys.

Error Index

put(Object, Object)

Puts the specified element into the hashtable, using the specified key.

Error Index

rehash()

Rehashes the content of the table into a bigger table.

Error Index

remove(Object)

Removes the element corresponding to the key.

Error Index

size()

Returns the number of elements contained in the hashtable.

Error Index

toString()

Converts to a rather lengthy String.

Error Index

Error Index

Hashtable

```
public Hashtable(int initialCapacity,  
                 float loadFactor)
```

Constructs a new, empty hashtable with the specified initial capacity and the specified load factor.

Parameters:

initialCapacity - the initial number of buckets

loadFactor - a number between 0.0 and 1.0, it defines the threshold for rehashing the hashtable into a bigger one.

Throws: IllegalArgumentException

If the initial capacity is less than or equal to zero.

Throws: IllegalArgumentException

If the load factor is less than or equal to zero.

Error Index

Hashtable

```
public Hashtable(int initialCapacity)
```

Constructs a new, empty hashtable with the specified initial capacity.

Parameters:

initialCapacity - the initial number of buckets

Error Index

Hashtable

```
public Hashtable()
```

Constructs a new, empty hashtable. A default capacity and load factor is used. Note that the hashtable will automatically grow when it gets full.

Error Index

Error Index size

public int size()

Returns the number of elements contained in the hashtable.

Overrides:

size in class Dictionary

Error Index isEmpty

public boolean isEmpty()

Returns true if the hashtable contains no elements.

Overrides:

isEmpty in class Dictionary

Error Index keys

public synchronized Enumeration keys()

Returns an enumeration of the hashtable's keys.

Overrides:

keys in class Dictionary

See Also:

elements, Enumeration

Error Index elements

public synchronized Enumeration elements()

Returns an enumeration of the elements. Use the Enumeration methods on the returned object to fetch the elements sequentially.

Overrides:

elements in class Dictionary

See Also:

keys, Enumeration

Error Index contains

public synchronized boolean contains(Object value)

Returns true if the specified object is an element of the hashtable. This operation is more expensive than the containsKey() method.

Parameters:

value - the value that we are looking for

Throws: NullPointerException

If the value being searched for is equal to null.

See Also:

containsKey

Error Index containsKey

public synchronized boolean containsKey(Object key)

Returns true if the collection contains an element for the key.

Parameters:

key - the key that we are looking for

See Also:

contains

Error Index get

public synchronized Object get(Object key)

Gets the object associated with the specified key in the hashtable.

Parameters:

key - the specified key

Returns:

s the element for the key or null if the key is not defined in the hash table.

Overrides:

get in class Dictionary

See Also:

put

Error Index rehash

protected void rehash()

Rehashes the content of the table into a bigger table. This method is called automatically when the hashtable's size exceeds the threshold.

Error Index put

public synchronized Object put(Object key,
Object value)

Puts the specified element into the hashtable, using the specified key. The element may be retrieved by doing a get() with the same key. The key and the element cannot be null.

Parameters:

key - the specified key in the hashtable
value - the specified element

Returns:

the old value of the key, or null if it did not have one.

Throws: NullPointerException

If the value of the element is equal to null.

Overrides:

put in class Dictionary

See Also:

get

Error Index remove

public synchronized Object remove(Object key)

Removes the element corresponding to the key. Does nothing if the key is not present.

Parameters:

key - the key that needs to be removed

Returns:

the value of key, or null if the key was not found.

Overrides:

remove in class Dictionary

Error Index clear

public synchronized void clear()

Clears the hash table so that it has no more elements in it.

Error Index clone

public synchronized Object clone()

Creates a clone of the hashtable. A shallow copy is made, the keys and elements themselves are NOT cloned. This is a relatively expensive operation.

Overrides:

clone in class Object

Error Index toString

public synchronized String toString()

Converts to a rather lengthy String.

Overrides:

toString in class Object

Class java.util.NoSuchElementException

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.NoSuchElementException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----java.lang.RuntimeException
                  |
                  +----java.util.NoSuchElementException
```

```
public class NoSuchElementException
    extends RuntimeException
```

Signals that an enumeration is empty.

See Also:

[Enumeration](#)

Error Index

Error Index

[NoSuchElementException\(\)](#)

Constructs a NoSuchElementException with no detail message.

Error Index

[NoSuchElementException\(String\)](#)

Constructs a NoSuchElementException with the specified detail message.

Error Index

Error Index

[NoSuchElementException](#)

```
public NoSuchElementException()
```

Constructs a NoSuchElementException with no detail message. A detail message is a String that describes this particular exception.

Error Index

NoSuchElementException

```
public NoSuchElementException(String s)
```

Constructs a NoSuchElementException with the specified detail message. A detail message is a String that describes this particular exception.

Parameters:

s - the detail message

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Observable

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Observable

[java.lang.Object](#)
|
+----java.util.Observable

```
public class Observable  
extends Object
```

This class should be subclassed by observable object, or "data" in the Model-View paradigm. An Observable object may have any number of Observers. Whenever the Observable instance changes, it notifies all of its observers. Notification is done by calling the update() method on all observers.

Error Index

Error Index

Observable()

Error Index

Error Index

addObserver(Observer)

Adds an observer to the observer list.

Error Index

clearChanged()

Clears an observable change.

Error Index

countObservers()

Counts the number of observers.

Error Index

deleteObserver(Observer)

Deletes an observer from the observer list.

Error Index

deleteObservers()

Deletes observers from the observer list.

Error Index

hasChanged()

Returns a true boolean if an observable change has occurred.

Error Index

notifyObservers()

Notifies all observers if an observable change occurs.

Error Index

notifyObservers(Object)

Notifies all observers of the specified observable change which occurred.

Error Index

setChanged()

Sets a flag to note an observable change.

Error Index

Error Index

Observable

public Observable()

Error Index

Error Index

addObserver

public synchronized void addObserver(Observer o)

Adds an observer to the observer list.

Parameters:

o - the observer to be added

Error Index

deleteObserver

public synchronized void deleteObserver(Observer o)

Deletes an observer from the observer list.

Parameters:

o - the observer to be deleted

Error Index notifyObservers

```
public void notifyObservers()
```

Notifies all observers if an observable change occurs.

Error Index notifyObservers

```
public synchronized void notifyObservers(Object arg)
```

Notifies all observers of the specified observable change which occurred.

Parameters:

arg - what is being notified

Error Index deleteObservers

```
public synchronized void deleteObservers()
```

Deletes observers from the observer list.

Error Index setChanged

```
protected synchronized void setChanged()
```

Sets a flag to note an observable change.

Error Index clearChanged

```
protected synchronized void clearChanged()
```

Clears an observable change.

Error Index hasChanged

```
public synchronized boolean hasChanged()
```

Returns a true boolean if an observable change has occurred.

Error Index **countObservers**

```
public synchronized int countObservers()
```

Counts the number of observers.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.util.Observer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface java.util.Observer

public interface **Observer**
extends [Object](#)

When implemented, this interface allows all classes to be observable by instances of class Observer.

Error Index

Error Index

update([Observable](#), [Object](#))

Called when observers in the observable list need to be updated.

Error Index

Error Index

update

public abstract void **update**([Observable](#) o,
[Object](#) arg)

Called when observers in the observable list need to be updated.

Parameters:

o - the list of observers

arg - the argument being notified

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Properties

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Properties

```
java.lang.Object
|
+----java.util.Dictionary
      |
      +----java.util.Hashtable
            |
            +----java.util.Properties
```

```
public class Properties
    extends Hashtable
```

Persistent properties class. This class is basically a hashtable that can be saved/loaded from a stream. If a property is not found, a property list containing defaults is searched. This allows arbitrary nesting.

Error Index

Error Index

[defaults](#)

Error Index

Error Index

[Properties\(\)](#)

Creates an empty property list.

Error Index

[Properties\(Properties\)](#)

Creates an empty property list with specified defaults.

Error Index

Error Index

getProperty(String)

Gets a property with the specified key.

Error Index

getProperty(String, String)

Gets a property with the specified key and default.

Error Index

list(PrintStream)

List properties, for debugging

Error Index

load(InputStream)

Loads properties from an InputStream.

Error Index

propertyNames()

Enumerates all the keys.

Error Index

save(OutputStream, String)

Save properties to an OutputStream.

Error Index

Error Index

defaults

protected Properties defaults

Error Index

Error Index

Properties

public Properties()

Creates an empty property list.

Error Index

Properties

public Properties(Properties defaults)

Creates an empty property list with specified defaults.

Parameters:

defaults - the defaults

Error Index

Error Index **load**

public synchronized void load(InputStream in) throws IOException

Loads properties from an InputStream.

Parameters:

in - the input stream

Throws: IOException

Error when reading from input stream.

Error Index **save**

public synchronized void save(OutputStream out,
String header)

Save properties to an OutputStream. Use the header as a comment at the top of the file.

Error Index **getProperty**

public String getProperty(String key)

Gets a property with the specified key. If the key is not found in this property list, tries the defaults. This method returns null if the property is not found.

Parameters:

key - the hashtable key

Error Index **getProperty**

public String getProperty(String key,
String defaultValue)

Gets a property with the specified key and default. If the key is not found in this property list, tries the defaults. This method returns defaultValue if the property is not found.

Error Index `propertyNames`

```
public Enumeration propertyNames()
```

Enumerates all the keys.

Error Index `list`

```
public void list(PrintStream out)
```

List properties, for debugging

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Random

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Random

[java.lang.Object](#)

|
+----java.util.Random

```
public class Random
    extends Object
```

A Random class generates a stream of pseudo-random numbers.

To create a new random number generator, use one of the following methods:

```
new Random()
new Random(long seed)
```

The form new Random() initializes the generator to a value based on the current time. The form new Random(long seed) seeds the random number generator with a specific initial value; use this if an application requires a repeatable stream of pseudo-random numbers.

The random number generator uses a 48-bit seed, which is modified using a linear congruential formula. See Donald Knuth, *The Art of Computer Programming, Volume 2*, Section 3.2.1. The generator's seed can be reset with the following method:

```
setSeed(long seed)
```

To create a pseudo-random number, use one of the following functions:

```
nextInt()
nextLong()
nextFloat()
nextDouble()
nextGaussian()
```

See Also:
[random](#)

Error Index

Error Index

Random()

Creates a new random number generator.

Error Index

Random(long)

Creates a new random number generator using a single long seed.

Error Index

Error Index

nextDouble()

Generates a pseudorandom uniformly distributed double value between 0.0 and 1.0.

Error Index

nextFloat()

Generates a pseudorandom uniformly distributed float value between 0.0 and 1.0.

Error Index

nextGaussian()

Generates a pseudorandom Gaussian distributed double value with mean 0.0 and standard deviation 1.0.

Error Index

nextInt()

Generates a pseudorandom uniformly distributed int value.

Error Index

nextLong()

Generate a pseudorandom uniformly distributed long value.

Error Index

setSeed(long)

Sets the seed of the random number generator using a single long seed.

Error Index

Error Index

Random

public Random()

Creates a new random number generator. Its seed will be initialized to a value based on the current time.

Error Index

Random

public Random(long seed)

Creates a new random number generator using a single long seed.

Parameters:

seed - the initial seed

See Also:

setSeed

Error Index

Error Index `setSeed`

```
public synchronized void setSeed(long seed)
```

Sets the seed of the random number generator using a single long seed.

Parameters:

seed - the initial seed

Error Index `nextInt`

```
public int nextInt()
```

Generates a pseudorandom uniformly distributed int value.

Returns:

an integer value.

Error Index `nextLong`

```
public long nextLong()
```

Generate a pseudorandom uniformly distributed long value.

Returns:

A long integer value

Error Index `nextFloat`

public float nextFloat()

Generates a pseudorandom uniformly distributed float value between 0.0 and 1.0.

Returns:

a float between 0.0 and 1.0 .

Error Index nextDouble

public double nextDouble()

Generates a pseudorandom uniformly distributed double value between 0.0 and 1.0.

Returns:

a float between 0.0 and 1.0 .

Error Index nextGaussian

public synchronized double nextGaussian()

Generates a pseudorandom Gaussian distributed double value with mean 0.0 and standard deviation 1.0.

Returns:

a Gaussian distributed double.

Class java.util.Stack

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Stack

```
java.lang.Object
|
+----java.util.Vector
      |
      +----java.util.Stack
```

```
public class Stack
    extends Vector
```

A Last-In-First-Out(LIFO) stack of objects.

Error Index

Error Index

Stack()

Error Index

Error Index

empty()

Returns true if the stack is empty.

Error Index

peek()

Peeks at the top of the stack.

Error Index

pop()

Pops an item off the stack.

Error Index

push(Object)

Pushes an item onto the stack.

Error Index

search(Object)

Sees if an object is on the stack.

Error Index

Error Index

Stack

public Stack()

Error Index

Error Index

push

public Object push(Object item)

Pushes an item onto the stack.

Parameters:

item - the item to be pushed on.

Error Index

pop

public Object pop()

Pops an item off the stack.

Throws: EmptyStackException

If the stack is empty.

Error Index

peek

public Object peek()

Peeks at the top of the stack.

Throws: EmptyStackException

If the stack is empty.

Error Index **empty**

```
public boolean empty()
```

Returns true if the stack is empty.

Error Index **search**

```
public int search(Object o)
```

Sees if an object is on the stack.

Parameters:

o - the desired object

Returns:

the distance from the top, or -1 if it is not found.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.StringTokenizer

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.StringTokenizer

[java.lang.Object](#)

|
+----java.util.StringTokenizer

```
public class StringTokenizer
    extends Object
    implements Enumeration
```

StringTokenizer is a class that controls simple linear tokenization of a String. The set of delimiters, which defaults to common whitespace characters, may be specified at creation time or on a per-token basis.

Example usage:

```
String s = "this is a test";
StringTokenizer st = new StringTokenizer(s);
while (st.hasMoreTokens()) {
    println(st.nextToken());
}
```

Prints the following on the console:

```
this
is
a
test
```

Error Index

Error Index

StringTokenizer(String, String, boolean)

Constructs a StringTokenizer on the specified String, using the specified delimiter set.

Error Index

StringTokenizer(String, String)

Constructs a StringTokenizer on the specified String, using the specified delimiter set.

Error Index

StringTokenizer(String)

Constructs a StringTokenizer on the specified String, using the default delimiter set (which is " \t\n\r").

Error Index

Error Index

countTokens()

Returns the next number of tokens in the String using the current delimiter set.

Error Index

hasMoreElements()

Returns true if the Enumeration has more elements.

Error Index

hasMoreTokens()

Returns true if more tokens exist.

Error Index

nextElement()

Returns the next element in the Enumeration.

Error Index

nextToken()

Returns the next token of the String.

Error Index

nextToken(String)

Returns the next token, after switching to the new delimiter set.

Error Index

Error Index

StringTokenizer

```
public StringTokenizer(String str,  
                     String delim,  
                     boolean returnTokens)
```

Constructs a StringTokenizer on the specified String, using the specified delimiter set.

Parameters:

str - the input String

delim - the delimiter String

returnTokens - returns delimiters as tokens or skip them

Error Index

StringTokenizer

```
public StringTokenizer(String str,
```

String delim)

Constructs a StringTokenizer on the specified String, using the specified delimiter set.

Parameters:

str - the input String

delim - the delimiter String

Error Index StringTokenizer

public StringTokenizer(String str)

Constructs a StringTokenizer on the specified String, using the default delimiter set (which is " \t\n\r").

Parameters:

str - the String

Error Index

Error Index hasMoreTokens

public boolean hasMoreTokens()

Returns true if more tokens exist.

Error Index nextToken

public String nextToken()

Returns the next token of the String.

Throws: NoSuchElementException

If there are no more tokens in the String.

Error Index nextToken

public String nextToken(String delim)

Returns the next token, after switching to the new delimiter set. The new delimiter set remains the

default after this call.

Parameters:

delim - the new delimiters

Error Index **hasMoreElements**

```
public boolean hasMoreElements()
```

Returns true if the Enumeration has more elements.

Error Index **nextElement**

```
public Object nextElement()
```

Returns the next element in the Enumeration.

Throws: NoSuchElementException

If there are no more elements in the enumeration.

Error Index **countTokens**

```
public int countTokens()
```

Returns the next number of tokens in the String using the current delimiter set. This is the number of times nextToken() can return before it will generate an exception. Use of this routine to count the number of tokens is faster than repeatedly calling nextToken() because the substrings are not constructed and returned for each token.

Class java.util.Vector

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class java.util.Vector

[java.lang.Object](#)

|

+----java.util.Vector

```
public class Vector
  extends Object
  implements Cloneable
```

Vector class (a growable array).

Each vector tries to optimize storage management by maintaining a capacity and a capacityIncrement. The capacity is always at least as large as the vector size; it is usually larger because as elements are added to the vector, the vector's storage increases in chunks the size of capacityIncrement. Setting the capacity to what you want before inserting a large number of objects will reduce the amount of incremental reallocation. You can safely ignore the capacity and the vector will still work correctly.

Error Index

Error Index

[capacityIncrement](#)

The size of the increment.

Error Index

[elementCount](#)

The number of elements in the buffer.

Error Index

[elementData](#)

The buffer where elements are stored.

Error Index

Error Index

[Vector](#)(int, int)

Constructs an empty vector with the specified storage capacity and the specified capacityIncrement.

Error Index

Vector(int)

Constructs an empty vector with the specified storage capacity.

Error Index

Vector()

Constructs an empty vector.

Error Index

Error Index

addElement(Object)

Adds the specified object as the last element of the vector.

Error Index

capacity()

Returns the current capacity of the vector.

Error Index

clone()

Clones this vector.

Error Index

contains(Object)

Returns true if the specified object is a value of the collection.

Error Index

copyInto(Object[])

Copies the elements of this vector into the specified array.

Error Index

elementAt(int)

Returns the element at the specified index.

Error Index

elements()

Returns an enumeration of the elements.

Error Index

ensureCapacity(int)

Ensures that the vector has at least the specified capacity.

Error Index

firstElement()

Returns the first element of the sequence.

Error Index

indexOf(Object)

Searches for the specified object, starting from the first position and returns an index to it.

Error Index

indexOf(Object, int)

Searches for the specified object, starting at the specified position and returns an index to it.

Error Index

insertElementAt(Object, int)

Inserts the specified object as an element at the specified index.

Error Index

isEmpty()

Returns true if the collection contains no values.

Error Index

lastElement()

Returns the last element of the sequence.

Error Index

lastIndexOf(Object)

Searches backwards for the specified object, starting from the last position and returns an index to it.

Error Index

lastIndexOf(Object, int)

Searches backwards for the specified object, starting from the specified position and returns an index to it.

Error Index

removeAllElements()

Removes all elements of the vector.

Error Index

removeElement(Object)

Removes the element from the vector.

Error Index

removeElementAt(int)

Deletes the element at the specified index.

Error Index

setElementAt(Object, int)

Sets the element at the specified index to be the specified object.

Error Index

setSize(int)

Sets the size of the vector.

Error Index

size()

Returns the number of elements in the vector.

Error Index

toString()

Converts the vector to a string.

Error Index

trimToSize()

Trims the vector's capacity down to size.

Error Index

Error Index

elementData

protected Object elementData[]

The buffer where elements are stored.

Error Index `elementCount`

protected int elementCount

The number of elements in the buffer.

Error Index `capacityIncrement`

protected int capacityIncrement

The size of the increment. If it is 0 the size of the the buffer is doubled everytime it needs to grow.

Error Index

Error Index `Vector`

```
public Vector(int initialCapacity,  
              int capacityIncrement)
```

Constructs an empty vector with the specified storage capacity and the specified capacityIncrement.

Parameters:

initialCapacity - the initial storage capacity of the vector

capacityIncrement - how much to increase the element's size by.

Error Index `Vector`

```
public Vector(int initialCapacity)
```

Constructs an empty vector with the specified storage capacity.

Parameters:

initialCapacity - the initial storage capacity of the vector

Error Index `Vector`

```
public Vector()
```

Constructs an empty vector.

Error Index

Error Index

copyInto

```
public final synchronized void copyInto(Object anArray[])
```

Copies the elements of this vector into the specified array.

Parameters:

anArray - the array where elements get copied into

Error Index

trimToSize

```
public final synchronized void trimToSize()
```

Trims the vector's capacity down to size. Use this operation to minimize the storage of a vector. Subsequent insertions will cause reallocation.

Error Index

ensureCapacity

```
public final synchronized void ensureCapacity(int minCapacity)
```

Ensures that the vector has at least the specified capacity.

Parameters:

minCapacity - the desired minimum capacity

Error Index

setSize

```
public final synchronized void setSize(int newSize)
```

Sets the size of the vector. If the size shrinks, the extra elements (at the end of the vector) are lost; if the size increases, the new elements are set to null.

Parameters:

newSize - the new size of the vector

Error Index capacity

```
public final int capacity()
```

Returns the current capacity of the vector.

Error Index size

```
public final int size()
```

Returns the number of elements in the vector. Note that this is not the same as the vector's capacity.

Error Index isEmpty

```
public final boolean isEmpty()
```

Returns true if the collection contains no values.

Error Index elements

```
public final synchronized Enumeration elements()
```

Returns an enumeration of the elements. Use the Enumeration methods on the returned object to fetch the elements sequentially.

Error Index contains

```
public final boolean contains(Object elem)
```

Returns true if the specified object is a value of the collection.

Parameters:

elem - the desired element

Error Index indexOf

```
public final int indexOf(Object elem)
```

Searches for the specified object, starting from the first position and returns an index to it.

Parameters:

elem - the desired element

Returns:

the index of the element, or -1 if it was not found.

Error Index indexOf

```
public final synchronized int indexOf(Object elem,  
                                     int index)
```

Searches for the specified object, starting at the specified position and returns an index to it.

Parameters:

elem - the desired element

index - the index where to start searching

Returns:

the index of the element, or -1 if it was not found.

Error Index lastIndexOf

```
public final int lastIndexOf(Object elem)
```

Searches backwards for the specified object, starting from the last position and returns an index to it.

Parameters:

elem - the desired element

Returns:

the index of the element, or -1 if it was not found.

Error Index lastIndexOf

```
public final synchronized int lastIndexOf(Object elem,  
                                     int index)
```

Searches backwards for the specified object, starting from the specified position and returns an index to it.

Parameters:

elem - the desired element

index - the index where to start searching

Returns:

the index of the element, or -1 if it was not found.

Error Index **elementAt**

public final synchronized Object elementAt(int index)

Returns the element at the specified index.

Parameters:

index - the index of the desired element

Throws: ArrayIndexOutOfBoundsException

If an invalid index was given.

Error Index **firstElement**

public final synchronized Object firstElement()

Returns the first element of the sequence.

Throws: NoSuchElementException

If the sequence is empty.

Error Index **lastElement**

public final synchronized Object lastElement()

Returns the last element of the sequence.

Throws: NoSuchElementException

If the sequence is empty.

Error Index **setElementAt**

public final synchronized void setElementAt(Object obj,
int index)

Sets the element at the specified index to be the specified object. The previous element at that position is discarded.

Parameters:

obj - what the element is to be set to

index - the specified index

Throws: ArrayIndexOutOfBoundsException
If the index was invalid.

Error Index removeElementAt

public final synchronized void removeElementAt(int index)

Deletes the element at the specified index. Elements with an index greater than the current index are moved down.

Parameters:

index - the element to remove

Throws: ArrayIndexOutOfBoundsException
If the index was invalid.

Error Index insertElementAt

public final synchronized void insertElementAt(Object obj,
int index)

Inserts the specified object as an element at the specified index. Elements with an index greater or equal to the current index are shifted up.

Parameters:

obj - the element to insert

index - where to insert the new element

Throws: ArrayIndexOutOfBoundsException
If the index was invalid.

Error Index addElement

public final synchronized void addElement(Object obj)

Adds the specified object as the last element of the vector.

Parameters:

obj - the element to be added

Error Index removeElement

public final synchronized boolean removeElement(Object obj)

Removes the element from the vector. If the object occurs more than once, only the first is removed. If the object is not an element, returns false.

Parameters:

obj - the element to be removed

Returns:

true if the element was actually removed; false otherwise.

Error Index removeAllElements

```
public final synchronized void removeAllElements()
```

Removes all elements of the vector. The vector becomes empty.

Error Index clone

```
public synchronized Object clone()
```

Clones this vector. The elements are **not** cloned.

Overrides:

clone in class Object

Error Index toString

```
public final synchronized String toString()
```

Converts the vector to a string. Useful for debugging.

Overrides:

toString in class Object

Interface

sun.tools.debug.DebuggerCallback

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Interface sun.tools.debug.DebuggerCallback

public interface **DebuggerCallback**
extends [Object](#)

The DebuggerCallback interface is used to communicate asynchronous information from the debugger to its client. This may be the actual client object, or a delegate of its choosing.

Error Index

Error Index

[breakpointEvent](#)(RemoteThread)

A breakpoint has been hit in the specified thread.

Error Index

[exceptionEvent](#)(RemoteThread, String)

An exception has occurred.

Error Index

[printToConsole](#)(String)

Print text to the debugger's console window.

Error Index

[quitEvent](#)()

The client interpreter has exited, either by returning from its main thread, or by calling System.exit().

Error Index

[threadDeathEvent](#)(RemoteThread)

A thread has died.

Error Index

Error Index

[printToConsole](#)

public abstract void **[printToConsole](#)**([String](#) text) throws [Exception](#)

Print text to the debugger's console window.

Error Index

breakpointEvent

public abstract void breakpointEvent(RemoteThread t) throws Exception

A breakpoint has been hit in the specified thread.

Error Index

exceptionEvent

public abstract void exceptionEvent(RemoteThread t,
String errorText) throws Exception

An exception has occurred.

Error Index

threadDeathEvent

public abstract void threadDeathEvent(RemoteThread t) throws Exception

A thread has died.

Error Index

quitEvent

public abstract void quitEvent() throws Exception

The client interpreter has exited, either by returning from its main thread, or by calling System.exit().

Class sun.tools.debug.RemoteArray

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteArray

```
java.lang.Object
|
+----sun.tools.debug.RemoteValue
      |
      +----sun.tools.debug.RemoteObject
            |
            +----sun.tools.debug.RemoteArray
```

```
public class RemoteArray
    extends RemoteObject
```

The RemoteArray class allows remote debugging of arrays.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[arrayTypeName](#)(int)

Return the element type as a string.

Error Index

[description](#)()

Return a description of the array.

Error Index

[getElement](#)(int)

Return an array element.

Error Index

[getElementType](#)()

Return the element type as a "TC_" constant, such as "TC_CHAR".

Error Index

[getElements](#)()

Returns a copy of the array as instances of RemoteValue.

Error Index

[getElements](#)(int, int)

Returns a copy of a portion of the array as instances of RemoteValue.

Error Index

getSize()

Return the number of elements in the array.

Error Index

toString()

Return a string version of the array.

Error Index

typeName()

Return this RemoteValue's type ("array").

Error Index

Error Index

getSize

```
public final int getSize()
```

Return the number of elements in the array.

Error Index

typeName

```
public String typeName()
```

Return this RemoteValue's type ("array").

Overrides:

typeName in class RemoteObject

Error Index

arrayTypeName

```
public String arrayTypeName(int type)
```

Return the element type as a string.

Error Index

getElementType

```
public final int getElementType() throws Exception
```

Return the element type as a "TC_" constant, such as "TC_CHAR".

Error Index getElement

public final RemoteValue getElement(int index) throws Exception

Return an array element.

Parameters:

index - the index of the element

Returns:

the element as a RemoteValue

Throws: ArrayIndexOutOfBoundsException

when the index is greater than the size of the array

Error Index getElements

public final RemoteValue[] getElements() throws Exception

Returns a copy of the array as instances of RemoteValue.

Error Index getElements

public final RemoteValue[] getElements(int beginIndex,
int endIndex) throws Exception

Returns a copy of a portion of the array as instances of RemoteValue.

Parameters:

beginIndex - the beginning array index

endIndex - the final array index

Throws: ArrayIndexOutOfBoundsException

when the index is greater than the size of the array

Error Index description

public String description()

Return a description of the array.

Overrides:

description in class RemoteObject

Error Index toString

public String toString()

Return a string version of the array.

Overrides:

toString in class RemoteObject

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteBoolean

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteBoolean

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteBoolean
```

```
public class RemoteBoolean
    extends RemoteValue
```

The RemoteBoolean class extends RemoteValue for booleans.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the boolean's value.

Error Index

[toString\(\)](#)

Return the boolean's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("boolean").

Error Index

Error Index

get

```
public boolean get()
```

Return the boolean's value.

Error Index typeName

```
public String typeName()
```

Print this RemoteValue's type ("boolean").

Overrides:

typeName in class RemoteValue

Error Index toString

```
public String toString()
```

Return the boolean's value as a string.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteByte

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteByte

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteByte
```

```
public class RemoteByte
extends RemoteValue
```

The RemoteByte class extends RemoteValue for bytes.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the byte's value.

Error Index

[toString\(\)](#)

Return the byte's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("byte").

Error Index

Error Index

get

public byte get()

Return the byte's value.

Error Index typeName

```
public String typeName()
```

Print this RemoteValue's type ("byte").

Overrides:

typeName in class RemoteValue

Error Index toString

```
public String toString()
```

Return the byte's value as a string.

Overrides:

toString in class Object

Class sun.tools.debug.RemoteChar

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteChar

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
      |
      +----sun.tools.debug.RemoteChar
```

```
public class RemoteChar
    extends RemoteValue
```

The RemoteChar class extends RemoteValue for chars.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the char's value.

Error Index

[toString\(\)](#)

Return the char's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("char").

Error Index

Error Index

get

```
public char get()
```

Return the char's value.

Error Index `typeName`

```
public String typeName()
```

Print this RemoteValue's type ("char").

Overrides:

`typeName` in class RemoteValue

Error Index `toString`

```
public String toString()
```

Return the char's value as a string.

Overrides:

`toString` in class Object

Class sun.tools.debug.RemoteClass

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteClass

```
java.lang.Object
|
+----<u>sun.tools.debug.RemoteValue</u>
    |
    +----<u>sun.tools.debug.RemoteObject</u>
        |
        +----<u>sun.tools.debug.RemoteClass</u>
```

```
public class RemoteClass
    extends <u>RemoteObject</u>
```

The RemoteClass class allows access to a class in a remote Java interpreter.

See Also:

[RemoteDebugger](#)

Error Index

Error Index

[catchExceptions\(\)](#)

Enter the debugger when an instance of this class is thrown.

Error Index

[clearBreakpoint\(int\)](#)

Clear a breakpoint at a specific address in a class.

Error Index

[clearBreakpointLine\(int\)](#)

Clear a breakpoint at a specified line.

Error Index

[clearBreakpointMethod\(RemoteField\)](#)

Clear a breakpoint at the start of a specified method.

Error Index

[description\(\)](#)

Return a (somewhat verbose) description.

Error Index

[getClassLoader\(\)](#)

Return the classloader for this class.

Error Index

getField(int)

Return the static field, specified by index.

Error Index

getField(String)

Return the static field, specified by name.

Error Index

getFieldValue(int)

Return the value of a static field, specified by its index

Error Index

getFieldValue(String)

Return the value of a static field, specified by name.

Error Index

getFields()

Return all the static fields for this class.

Error Index

getInstanceField(int)

Return the instance field, specified by its index.

Error Index

getInstanceFields()

Return all the instance fields for this class.

Error Index

getInterfaces()

Return the interfaces for this class.

Error Index

getMethod(String)

Return the method, specified by name.

Error Index

getMethodNames()

Return the names of all methods supported by this class.

Error Index

getMethods()

Return the class's methods.

Error Index

getName()

Returns the name of the class.

Error Index

getSourceFile()

Get the source file referenced by this stackframe.

Error Index

getSourceFileName()

Get the name of the source file referenced by this stackframe.

Error Index

getStaticFields()

Return all the static fields for this class.

Error Index

getSuperclass()

Return the superclass for this class.

Error Index

ignoreExceptions()

Don't enter the debugger when an instance of this class is thrown.

Error Index

isInterface()

Is this RemoteClass an interface?

Error Index

setBreakpointLine(int)

Set a breakpoint at a specified source line number in a class.

Error Index

setBreakpointMethod(RemoteField)

Set a breakpoint at the first line of a class method.

Error Index

toString()

Return a (somewhat verbose) description.

Error Index

typeName()

Returns the name of the class as its type.

Error Index

Error Index

getName

public String getName() throws Exception

Returns the name of the class.

Error Index

typeName

public String typeName() throws Exception

Returns the name of the class as its type.

Overrides:

typeName in class RemoteObject

Error Index

isInterface

public boolean isInterface() throws Exception

Is this RemoteClass an interface?

Error Index **getSuperclass**

public RemoteClass getSuperclass() throws Exception

Return the superclass for this class.

Error Index **getClassLoader**

public RemoteObject getClassLoader() throws Exception

Return the classloader for this class.

Error Index **getInterfaces**

public RemoteClass[] getInterfaces() throws Exception

Return the interfaces for this class.

Error Index **getSourceFileName**

public String getSourceFileName()

Get the name of the source file referenced by this stackframe.

Error Index **getSourceFile**

public InputStream getSourceFile() throws Exception

Get the source file referenced by this stackframe.

Error Index **getFields**

public RemoteField[] getFields() throws Exception

Return all the static fields for this class.

Overrides:

getFields in class RemoteObject

Error Index

getStaticFields

public RemoteField[] getStaticFields() throws Exception

Return all the static fields for this class.

Error Index

getInstanceFields

public RemoteField[] getInstanceFields() throws Exception

Return all the instance fields for this class. Note: because this is a RemoteClass method, only the name and type methods will be valid, not the data.

Error Index

getField

public RemoteField getField(int n) throws Exception

Return the static field, specified by index.

Throws: ArrayIndexOutOfBoundsException

when the index is greater than the number of instance variables

Overrides:

getField in class RemoteObject

Error Index

getField

public RemoteField getField(String name) throws Exception

Return the static field, specified by name.

Overrides:

getField in class RemoteObject

Error Index

getInstanceField

public RemoteField getInstanceField(int n) throws Exception

Return the instance field, specified by its index. Note: because this is a RemoteClass method, only the name and type information is valid, not the data.

Throws: ArrayIndexOutOfBoundsException

when the index is greater than the number of instance variables

Error Index **getFieldValue**

public RemoteValue getFieldValue(int n) throws Exception

Return the value of a static field, specified by its index

Overrides:

getFieldValue in class RemoteObject

Error Index **getFieldValue**

public RemoteValue getFieldValue(String name) throws Exception

Return the value of a static field, specified by name.

Overrides:

getFieldValue in class RemoteObject

Error Index **getMethod**

public RemoteField getMethod(String name) throws Exception

Return the method, specified by name.

Error Index **getMethods**

public RemoteField[] getMethods() throws Exception

Return the class's methods.

Error Index **getMethodNames**

public String[] getMethodNames() throws Exception

Return the names of all methods supported by this class.

Error Index

setBreakpointLine

public String setBreakpointLine(int lineno) throws Exception

Set a breakpoint at a specified source line number in a class.

Parameters:

lineno - the line number where the breakpoint is set

Returns:

an empty string if successful, otherwise a description of the error.

Error Index

setBreakpointMethod

public String setBreakpointMethod(RemoteField method) throws Exception

Set a breakpoint at the first line of a class method.

Parameters:

method - the method where the breakpoint is set

Returns:

an empty string if successful, otherwise a description of the error.

Error Index

clearBreakpoint

public String clearBreakpoint(int pc) throws Exception

Clear a breakpoint at a specific address in a class.

Parameters:

pc - the address of the breakpoint to be cleared

Returns:

an empty string if successful, otherwise a description of the error.

Error Index

clearBreakpointLine

public String clearBreakpointLine(int lineno) throws Exception

Clear a breakpoint at a specified line.

Parameters:

lineno - the line number of the breakpoint to be cleared

Returns:

an empty string if successful, otherwise a description of the error.

Error Index

clearBreakpointMethod

public String clearBreakpointMethod(RemoteField method) throws Exception

Clear a breakpoint at the start of a specified method.

Parameters:

method - the method of the breakpoint to be cleared

Returns:

an empty string if successful, otherwise a description of the error.

Error Index

catchExceptions

public void catchExceptions() throws Exception

Enter the debugger when an instance of this class is thrown.

Throws: ClassCastException

when this class isn't an exception class

Error Index

ignoreExceptions

public void ignoreExceptions() throws Exception

Don't enter the debugger when an instance of this class is thrown.

Throws: ClassCastException

when this class isn't an exception class

Error Index

description

public String description()

Return a (somewhat verbose) description.

Overrides:

description in class RemoteObject

Error Index toString

public String toString()

Return a (somewhat verbose) description.

Overrides:

toString in class RemoteObject

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteDebugger

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteDebugger

[java.lang.Object](#)

|

+----sun.tools.debug.RemoteDebugger

public class **RemoteDebugger**
extends [Object](#)

The RemoteDebugger class defines a client interface to the Java debugging classes. It is used to instantiate a connection with the Java interpreter being debugged.

Error Index

Error Index

[RemoteDebugger](#)(String, String, DebuggerCallback, boolean)

Create a remote debugger, connecting it with a running Java interpreter.

To connect to a running interpreter, it must be started with the "-debug" option, whereupon it will print out the password for that debugging session.

Error Index

[RemoteDebugger](#)(String, DebuggerCallback, boolean)

Create a remote debugger, and connect it to a new client interpreter.

Error Index

Error Index

[close](#)()

Close the connection to the remote debugging agent.

Error Index

[findClass](#)(String)

Find a specified class.

Error Index

freeMemory()

Report the free memory available to the Java interpreter being debugged.

Error Index

gc(RemoteObject[])

Free all objects referenced by the debugger.

Error Index

get(Integer)

Get an object from the remote object cache.

Error Index

getExceptionCatchList()

Return the list of the exceptions the debugger will stop on.

Error Index

getSourcePath()

Return the source file path the Agent is currently using.

Error Index

itrace(boolean)

Turn on/off instruction tracing.

Error Index

listBreakpoints()

Return a list of the breakpoints which are currently set.

Error Index

listClasses()

List the currently known classes.

Error Index

listThreadGroups(RemoteThreadGroup)

List threadgroups

Error Index

run(int, String[])

Load and run a runnable Java class, with any optional parameters.

Error Index

setSourcePath(String)

Specify the list of paths to use when searching for a source file.

Error Index

totalMemory()

Report the total memory usage of the Java interpreter being debugged.

Error Index

trace(boolean)

Turn on/off method call tracing.

Error Index

Error Index

RemoteDebugger

```
public RemoteDebugger(String host,  
    String password,  
    DebuggerCallback client,  
    boolean verbose) throws Exception
```

Create a remote debugger, connecting it with a running Java interpreter.

To connect to a running interpreter, it must be started with the "-debug" option, whereupon it will print out the password for that debugging session.

Parameters:

- host - the name of the system where a debuggable Java instance is running (default is localhost).
- password - the password reported by the debuggable Java instance. This should be null when starting a client interpreter.
- client - the object to which notification messages are sent (it must support the DebuggerCallback interface)
- verbose - turn on internal debugger message text

Error Index RemoteDebugger

```
public RemoteDebugger(String javaArgs,  
    DebuggerCallback client,  
    boolean verbose) throws Exception
```

Create a remote debugger, and connect it to a new client interpreter.

Parameters:

- javaArgs - optional java command-line parameters, such as -classpath
- client - the object to which notification messages are sent (it must support the DebuggerCallback interface)
- verbose - turn on internal debugger message text

Error Index

Error Index close

```
public void close()
```

Close the connection to the remote debugging agent.

Error Index **get**

public RemoteObject get(Integer id)

Get an object from the remote object cache.

Parameters:

id - the remote object's id

Returns:

the specified RemoteObject, or null if not cached.

Error Index **listClasses**

public RemoteClass[] listClasses() throws Exception

List the currently known classes.

Error Index **findClass**

public RemoteClass findClass(String name) throws Exception

Find a specified class. If the class isn't already known by the remote debugger, the lookup request will be passed to the Java interpreter being debugged. NOTE: Substrings, such as "String" for "java.lang.String" will return with the first match, and will not be successfully found if the request is passed to the remote interpreter.

Parameters:

name - the name (or a substring of the name) of the class

Returns:

the specified (Remote)Class, or null if not found.

Error Index **listThreadGroups**

public RemoteThreadGroup[] listThreadGroups(RemoteThreadGroup tg) throws Exception

List threadgroups

Parameters:

tg - the threadgroup which hold the groups to be listed, or null for all threadgroups

Error Index **gc**

public void gc(RemoteObject save_list[]) throws Exception

Free all objects referenced by the debugger. The remote debugger maintains a copy of each object it has examined, so that references won't become invalidated by the garbage collector of the Java interpreter being debugged. The gc() method frees all all of these references, except those specified to save.

Parameters:

save_list - the list of objects to save.

Error Index **trace**

public void trace(boolean traceOn) throws Exception

Turn on/off method call tracing. When turned on, each method call is reported to the stdout of the Java interpreter being debugged. This output is not captured in any way by the remote debugger.

Parameters:

traceOn - turn tracing on or off

Error Index **itrace**

public void itrace(boolean traceOn) throws Exception

Turn on/off instruction tracing. When turned on, each Java instruction is reported to the stdout of the Java interpreter being debugged. This output is not captured in any way by the remote debugger.

Parameters:

traceOn - turn tracing on or off

Error Index **totalMemory**

public int totalMemory() throws Exception

Report the total memory usage of the Java interpreter being debugged.

Error Index **freeMemory**

public int freeMemory() throws Exception

Report the free memory available to the Java interpreter being debugged.

Error Indexrun

```
public RemoteThreadGroup run(int argc,  
    String argv[]) throws Exception
```

Load and run a runnable Java class, with any optional parameters. The class is started inside a new threadgroup in the Java interpreter being debugged. NOTE: Although it is possible to run multiple runnable classes from the same Java interpreter, there is no guarantee that all applets will work cleanly with each other. For example, two applets may want exclusive access to the same shared resource, such as a specific port.

Parameters:

argc - the number of parameters

argv - the array of parameters: the class to be run is first, followed by any optional parameters used by that class.

Returns:

the new ThreadGroup the class is running in, or null on error

Error IndexlistBreakpoints

```
public String[] listBreakpoints() throws Exception
```

Return a list of the breakpoints which are currently set.

Returns:

an array of Strings of the form "class_name:line_number".

Error IndexgetExceptionCatchList

```
public String[] getExceptionCatchList() throws Exception
```

Return the list of the exceptions the debugger will stop on.

Returns:

an array of exception class names, which may be zero-length.

Error IndexgetSourcePath

```
public String getSourcePath() throws Exception
```

Return the source file path the Agent is currently using.

Returns:

a string consisting of a list of colon-delineated paths.

Error Index **setSourcePath**

public void setSourcePath(String pathList) throws Exception

Specify the list of paths to use when searching for a source file.

Parameters:

pathList - a string consisting of a list of colon-delineated paths.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteDouble

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteDouble

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteDouble
```

```
public class RemoteDouble
    extends RemoteValue
```

The RemoteDouble class extends RemoteValue for doubles.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the double's value.

Error Index

[toString\(\)](#)

Return the double's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("double").

Error Index

Error Index

get

```
public double get()
```

Return the double's value.

Error Index `typeName`

```
public String typeName()
```

Print this RemoteValue's type ("double").

Overrides:

`typeName` in class RemoteValue

Error Index `toString`

```
public String toString()
```

Return the double's value as a string.

Overrides:

`toString` in class Object

Class sun.tools.debug.RemoteField

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteField

[java.lang.Object](#)

```
|
+----sun.tools.debug.Field
      |
      +----sun.tools.debug.RemoteField
```

public class **RemoteField**
extends [Field](#)
implements [AgentConstants](#)

A RemoteField allows access to a variable or method of an object or class in a remote Java interpreter.

See Also:

[RemoteStackVariable](#)

Error Index

Error Index

[getModifiers\(\)](#)

Returns a string with the field's modifiers, such as "public", "static", "final", etc.

Error Index

[getName\(\)](#)

Returns the name of the field.

Error Index

[getType\(\)](#)

Returns a type string describing the field.

Error Index

[isStatic\(\)](#)

Returns whether the field is static (a class variable or method).

Error Index

[toString\(\)](#)

Returns a String that represents the value of this Object.

Error Index

Error Index `getName`

```
public String getName()
```

Returns the name of the field.

Error Index `getType`

```
public String getType()
```

Returns a type string describing the field.

Error Index `getModifiers`

```
public String getModifiers()
```

Returns a string with the field's modifiers, such as "public", "static", "final", etc. If the field has no modifiers, an empty String is returned.

Error Index `isStatic`

```
public boolean isStatic()
```

Returns whether the field is static (a class variable or method).

Error Index `toString`

```
public String toString()
```

Returns a String that represents the value of this Object.

Overrides:

toString in class Object

Class sun.tools.debug.RemoteFloat

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteFloat

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteFloat
```

```
public class RemoteFloat
extends RemoteValue
```

The RemoteFloat class extends RemoteValue for floats.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the float's value.

Error Index

[toString\(\)](#)

Return the float's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("float").

Error Index

Error Index

get

```
public float get()
```

Return the float's value.

Error Index `typeName`

```
public String typeName()
```

Print this RemoteValue's type ("float").

Overrides:

`typeName` in class RemoteValue

Error Index `toString`

```
public String toString()
```

Return the float's value as a string.

Overrides:

`toString` in class Object

Class sun.tools.debug.RemoteInt

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteInt

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteInt
```

```
public class RemoteInt
    extends RemoteValue
```

The RemoteInt class extends RemoteValue for ints.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[RemoteInt](#)(int)

Error Index

Error Index

[get](#)()

Return the int's value.

Error Index

[toString](#)()

Return the int's value as a string.

Error Index

[typeName](#)()

Print this RemoteValue's type ("int").

Error Index

Error Index

RemoteInt

```
public RemoteInt(int i)
```

Error Index

Error Index

get

```
public int get()
```

Return the int's value.

Error Index

typeName

```
public String typeName()
```

Print this RemoteValue's type ("int").

Overrides:

typeName in class RemoteValue

Error Index

toString

```
public String toString()
```

Return the int's value as a string.

Overrides:

toString in class Object

Class sun.tools.debug.RemoteLong

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteLong

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteLong
```

```
public class RemoteLong
    extends RemoteValue
```

The RemoteLong class extends RemoteValue for longs.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the long's value.

Error Index

[toString\(\)](#)

Return the long's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("long").

Error Index

Error Index

get

```
public long get()
```


Return the long's value.

Error Index typeName

```
public String typeName()
```

Print this RemoteValue's type ("long").

Overrides:

typeName in class RemoteValue

Error Index toString

```
public String toString()
```

Return the long's value as a string.

Overrides:

toString in class Object

Class sun.tools.debug.RemoteObject

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteObject

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteObject
```

public class **RemoteObject**
extends [RemoteValue](#)

The RemoteObject class allows access to an object in a remote Java interpreter. Remote objects are not created by the local debugger, but are returned by the remote debugging agent when queried for the values of instance or static variables of known objects (such as classes), or from local (stack) variables. Each remote object has a reference cached by the remote Java interpreter, so that the object will not be garbage-collected during examination. The RemoteDebugger's gc() operation frees references to objects that are no longer being examined.

See Also:

[RemoteDebugger](#), [RemoteClass](#), [RemoteString](#), [RemoteThread](#), [RemoteThreadGroup](#)

Error Index

Error Index

[description\(\)](#)

Return a description of the object.

Error Index

[getClazz\(\)](#)

Returns the object's class.

Error Index

[getField\(int\)](#)

Return an instance variable, specified by slot number.

Error Index

[getField\(String\)](#)

Return an instance variable, specified by name.

Error Index

[getFieldValue\(int\)](#)

Returns the value of an object's instance variable.

Error Index

getFieldValue(String)

Returns the value of an object's instance variable.

Error Index

getFields()

Return the instance (non-static) fields of an object.

Error Index

getId()

Returns the id of the object.

Error Index

toString()

Return object as a string.

Error Index

typeName()

Returns the RemoteValue's type name ("Object").

Error Index

Error Index

typeName

public String typeName() throws Exception

Returns the RemoteValue's type name ("Object").

Overrides:

typeName in class RemoteValue

Error Index

getId

public final int getId()

Returns the id of the object.

Error Index

getClazz

public final RemoteClass getClazz()

Returns the object's class.

Error Index getFieldValue

public RemoteValue getFieldValue(int n) throws Exception

Returns the value of an object's instance variable.

Parameters:

n - the slot number of the variable to be returned.

Error Index getFieldValue

public RemoteValue getFieldValue(String name) throws Exception

Returns the value of an object's instance variable.

Parameters:

name - the name of the instance variable

Returns:

the variable as a RemoteValue, or null if name not found.

Error Index getFields

public RemoteField[] getFields() throws Exception

Return the instance (non-static) fields of an object.

Error Index getField

public RemoteField getField(int n) throws Exception

Return an instance variable, specified by slot number.

Parameters:

n - the slot number of the variable to be returned.

Error Index getField

public RemoteField getField(String name) throws Exception

Return an instance variable, specified by name.

Parameters:

name - the name of the instance variable

Returns:

the variable as a RemoteField, or null if name not found.

Error Index description

```
public String description()
```

Return a description of the object.

Overrides:

description in class RemoteValue

Error Index toString

```
public String toString()
```

Return object as a string.

Overrides:

toString in class Object

Class sun.tools.debug.RemoteShort

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteShort

[java.lang.Object](#)

```
|
+----sun.tools.debug.RemoteValue
|
+----sun.tools.debug.RemoteShort
```

```
public class RemoteShort
    extends RemoteValue
```

The RemoteShort class extends RemoteValue for shorts.

See Also:

[RemoteValue](#), [RemoteDebugger](#)

Error Index

Error Index

[get\(\)](#)

Return the short's value.

Error Index

[toString\(\)](#)

Return the short's value as a string.

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("short").

Error Index

Error Index

get

public short get()

Return the short's value.

Error Index typeName

```
public String typeName()
```

Print this RemoteValue's type ("short").

Overrides:

typeName in class RemoteValue

Error Index toString

```
public String toString()
```

Return the short's value as a string.

Overrides:

toString in class Object

Class sun.tools.debug.RemoteStackFrame

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteStackFrame

[java.lang.Object](#)

```
|
+----sun.tools.debug.StackFrame
|
+----sun.tools.debug.RemoteStackFrame
```

```
public class RemoteStackFrame
    extends StackFrame
```

The RemoteStackFrame class provides access to a stackframe of a suspended thread.

See Also:

[RemoteDebugger](#), [RemoteThread](#)

Error Index

Error Index

[getLineNumber\(\)](#)

Return the source file line number.

Error Index

[getLocalVariable\(String\)](#)

Return a specific (named) stack variable.

Error Index

[getLocalVariables\(\)](#)

Return an array of all valid local variables and method arguments for this stack frame.

Error Index

[getMethodName\(\)](#)

Get the method name referenced by this stackframe.

Error Index

[getPC\(\)](#)

Get the program counter referenced by this stackframe.

Error Index

[getRemoteClass\(\)](#)

Get the class this stackframe references.

Error Index

Error Index

getLocalVariable

public RemoteStackVariable getLocalVariable(String name) throws Exception

Return a specific (named) stack variable. A slot number of -1 indicates that the variable is not currently in scope.

Error Index

getLocalVariables

public RemoteStackVariable[] getLocalVariables() throws Exception

Return an array of all valid local variables and method arguments for this stack frame.

Error Index

getLineNumber

public int getLineNumber()

Return the source file line number.

Error Index

getMethodName

public String getMethodName()

Get the method name referenced by this stackframe.

Error Index

getPC

public int getPC()

Get the program counter referenced by this stackframe.

Error Index

getRemoteClass

public RemoteClass getRemoteClass()

Get the class this stackframe references.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteStackVariable

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteStackVariable

[java.lang.Object](#)

```
|
+----sun.tools.debug.LocalVariable
|
+----sun.tools.debug.RemoteStackVariable
```

```
public class RemoteStackVariable
    extends LocalVariable
```

A RemoteStackVariable represents a method argument or local variable. It is similar to a RemoteField, but is much more transient in nature.

See Also:

[RemoteField](#)

Error Index

Error Index

[getName\(\)](#)

Return the name of a stack variable or argument.

Error Index

[getValue\(\)](#)

Return the value of a stack variable or argument.

Error Index

[inScope\(\)](#)

Return whether variable is in scope.

Error Index

Error Index

[getName](#)

```
public String getName()
```

Return the name of a stack variable or argument.

Error Index **getValue**

```
public RemoteValue getValue()
```

Return the value of a stack variable or argument.

Error Index **inScope**

```
public boolean inScope()
```

Return whether variable is in scope.

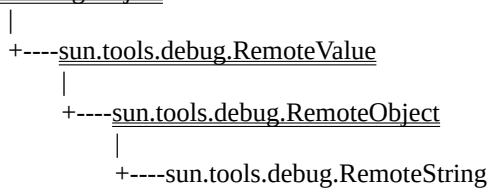
[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteString

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteString

[java.lang.Object](#)



```
public class RemoteString
    extends RemoteObject
```

The RemoteString class allows access to a string in a remote Java interpreter.

See Also:

[RemoteDebugger](#)

Error Index

Error Index

[description\(\)](#)

Return the string value, or "null"

Error Index

[toString\(\)](#)

Return the string value, or "null"

Error Index

[typeName\(\)](#)

Print this RemoteValue's type ("String").

Error Index

Error Index

[typeName](#)

```
public String typeName()
```

Print this RemoteValue's type ("String").

Overrides:

typeName in class RemoteObject

Error Indexdescription

```
public String description()
```

Return the string value, or "null"

Overrides:

description in class RemoteObject

Error IndextoString

```
public String toString()
```

Return the string value, or "null"

Overrides:

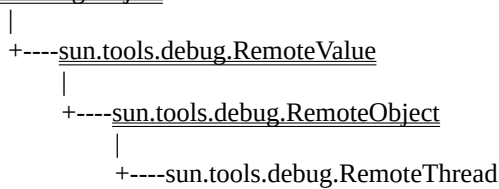
toString in class RemoteObject

Class sun.tools.debug.RemoteThread

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteThread

[java.lang.Object](#)



```
public class RemoteThread
    extends RemoteObject
```

The RemoteThread class allows access to a thread in a remote Java interpreter.

See Also:

[RemoteDebugger](#), [RemoteThreadGroup](#)

Error Index

Error Index

[cont\(\)](#)

Resume this thread from a breakpoint, unless it previously suspended.

Error Index

[down\(int\)](#)

Change the current stackframe to be one or more frames lower (as in, toward the current program counter).

Error Index

[dumpStack\(\)](#)

Dump the stack.

Error Index

[getCurrentFrame\(\)](#)

Get the current stack frame.

Error Index

[getCurrentFrameIndex\(\)](#)

Return the current stackframe index

Error Index

getName()

Return the name of the thread.

Error Index

getStackVariable(String)

Return a stack variable from the current stackframe.

Error Index

getStackVariables()

Return the arguments and local variable from the current stackframe.

Error Index

getStatus()

Return the thread status description

Error Index

isSuspended()

Return whether this thread is suspended.

Error Index

next()

Continue execution of this thread to the next line, but don't step into a method call.

Error Index

resetCurrentFrameIndex()

Reset the current stackframe

Error Index

resume()

Resume execution of this thread.

Error Index

setCurrentFrameIndex(int)

Set the current stackframe index

Error Index

step(boolean)

Continue execution of this thread to the next instruction or line.

Error Index

stop()

Stop the remote thread.

Error Index

suspend()

Suspend execution of this thread.

Error Index

up(int)

Change the current stackframe to be one or more frames higher (as in, away from the current program counter).

Error Index

Error Index `getName`

public String getName() throws Exception

Return the name of the thread.

Error Index `getCurrentFrameIndex`

public int getCurrentFrameIndex()

Return the current stackframe index

Error Index `setCurrentFrameIndex`

public void setCurrentFrameIndex(int iFrame)

Set the current stackframe index

Error Index `resetCurrentFrameIndex`

public void resetCurrentFrameIndex()

Reset the current stackframe

Error Index `up`

public void up(int nFrames) throws Exception

Change the current stackframe to be one or more frames higher (as in, away from the current program counter).

Parameters:

nFrames - the number of stackframes

Throws: IllegalAccessError

when the thread isn't suspended or waiting at a breakpoint

Throws: ArrayIndexOutOfBoundsException

when the requested frame is beyond the stack boundary

Error Index `down`

public void down(int nFrames) throws Exception

Change the current stackframe to be one or more frames lower (as in, toward the current program counter).

Parameters:

nFrames - the number of stackframes

Throws: IllegalAccessError

when the thread isn't suspended or waiting at a breakpoint

Throws: ArrayIndexOutOfBoundsException

when the requested frame is beyond the stack boundary

Error Index **getStatus**

public String getStatus() throws Exception

Return the thread status description

Error Index **dumpStack**

public RemoteStackFrame[] dumpStack() throws Exception

Dump the stack.

Error Index **getCurrentFrame**

public RemoteStackFrame getCurrentFrame() throws Exception

Get the current stack frame.

Throws: IllegalAccessError

when the thread isn't suspended or waiting at a breakpoint

Error Index **suspend**

public void suspend() throws Exception

Suspend execution of this thread.

Error Index **resume**

public void resume() throws Exception

Resume execution of this thread.

Error Index **step**

public void step(boolean skipLine) throws Exception

Continue execution of this thread to the next instruction or line.

Parameters:

skipLine - true to execute to next source line, false to next instruction.

Throws: IllegalAccessError

when the thread isn't suspended or waiting at a breakpoint

Error Index **next**

public void next() throws Exception

Continue execution of this thread to the next line, but don't step into a method call. If no line information is available, next() is equivalent to step().

Throws: IllegalAccessError

when the thread isn't suspended or waiting at a breakpoint

Error Index **isSuspended**

public boolean isSuspended()

Return whether this thread is suspended.

Error Index **cont**

public void cont() throws Exception

Resume this thread from a breakpoint, unless it previously suspended.

Error Index **stop**

public void stop() throws Exception

Stop the remote thread.

Error Index `getStackVariable`

public [RemoteStackVariable](#) getStackVariable([String](#) name) throws [Exception](#)

Return a stack variable from the current stackframe.

Returns:

the variable as a RemoteValue, or null if not found.

Error Index `getStackVariables`

public [RemoteStackVariable](#)[] getStackVariables() throws [Exception](#)

Return the arguments and local variable from the current stackframe.

Returns:

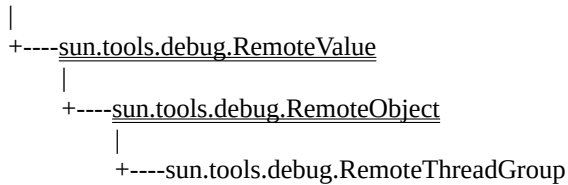
an array of RemoteValues.

Class sun.tools.debug.RemoteThreadGroup

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteThreadGroup

[java.lang.Object](#)



```
public class RemoteThreadGroup
    extends RemoteObject
```

The RemoteThreadGroup class allows access to a threadgroup in a remote Java interpreter.

See Also:

[RemoteDebugger](#), [RemoteThreadGroup](#)

Error Index

Error Index

[getName\(\)](#)

Return the threadgroup's name.

Error Index

[listThreads\(boolean\)](#)

List a threadgroup's threads

Error Index

[stop\(\)](#)

Stop the remote threadgroup.

Error Index

Error Index

[getName](#)

public String getName() throws Exception

Return the threadgroup's name.

Error Index stop

public void stop() throws Exception

Stop the remote threadgroup.

Error Index listThreads

public RemoteThread[] listThreads(boolean recurse) throws Exception

List a threadgroup's threads

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteValue

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.RemoteValue

[java.lang.Object](#)

|
+----sun.tools.debug.RemoteValue

public class **RemoteValue**
extends [Object](#)
implements [AgentConstants](#)

The RemoteValue class allows access to a copy of a value in the remote Java interpreter. This value may be a primitive type, such as a boolean or float, or an object, class, array, etc. Remote values are not created by the local debugger, but are returned by the remote debugging agent when queried for the values of instance or static variables of known objects, or from local (stack) variables.

See Also:

[RemoteDebugger](#), [RemoteArray](#), [RemoteBoolean](#), [RemoteByte](#), [RemoteChar](#), [RemoteClass](#), [RemoteDouble](#), [RemoteFloat](#), [RemoteInt](#), [RemoteLong](#), [RemoteObject](#), [RemoteShort](#), [RemoteString](#), [RemoteThread](#), [RemoteThreadGroup](#)

Error Index

Error Index

[description\(\)](#)

Return a description of the RemoteValue.

Error Index

[fromHex](#)(String)

Convert hexadecimal strings to ints.

Error Index

[getType](#)()

Returns the RemoteValue's type.

Error Index

[isObject](#)()

Returns whether the RemoteValue is an Object (as opposed to a primitive type, such as int).

Error Index

[toHex](#)(int)

Convert an int to a hexadecimal string.

Error Index

typeName()

Returns the RemoteValue's type as a string.

Error Index

Error Index

getType

```
public final int getType()
```

Returns the RemoteValue's type.

Error Index

isObject

```
public final boolean isObject()
```

Returns whether the RemoteValue is an Object (as opposed to a primitive type, such as int).

Error Index

typeName

```
public abstract String typeName() throws Exception
```

Returns the RemoteValue's type as a string.

Error Index

description

```
public String description()
```

Return a description of the RemoteValue.

Error Index

toHex

```
public static String toHex(int n)
```

Convert an int to a hexadecimal string.

Error Index

fromHex


```
public static int fromHex(String hexStr)
```

Convert hexadecimal strings to ints.

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.StackFrame

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class sun.tools.debug.StackFrame

[java.lang.Object](#)

|

+----sun.tools.debug.StackFrame

public class **StackFrame**
extends [Object](#)

The StackFrame class represents a stack frame of a suspended thread.

See Also:

[RemoteDebugger](#), [RemoteStackFrame](#), [RemoteThread](#)

Error Index

Error Index

[StackFrame\(\)](#)

Error Index

Error Index

[toString\(\)](#)

Returns a String that represents the value of this Object.

Error Index

Error Index

[StackFrame](#)

public StackFrame()

Error Index

Error Index toString

public String toString()

Returns a String that represents the value of this Object.

Overrides:

toString in class Object

[All Packages](#) [Class Hierarchy](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class Hierarchy

[All Packages](#) [Index](#)

Class Hierarchy

- class java.lang.[Object](#)
 - interface java.applet.[AppletContext](#)
 - interface java.applet.[AppletStub](#)
 - interface java.applet.[AudioClip](#)
 - class java.util.[BitSet](#) (implements java.lang.[Cloneable](#))
 - class java.lang.[Boolean](#)
 - class java.awt.[BorderLayout](#) (implements java.awt.[LayoutManager](#))
 - interface java.awt.peer.[ButtonPeer](#) (extends java.awt.peer.[ComponentPeer](#))
 - interface java.awt.peer.[CanvasPeer](#) (extends java.awt.peer.[ComponentPeer](#))
 - class java.awt.[CardLayout](#) (implements java.awt.[LayoutManager](#))
 - class java.lang.[Character](#)
 - class java.awt.[CheckboxGroup](#)
 - interface java.awt.peer.[CheckboxMenuItemPeer](#) (extends java.awt.peer.[MenuItemPeer](#))
 - interface java.awt.peer.[CheckboxPeer](#) (extends java.awt.peer.[ComponentPeer](#))
 - interface java.awt.peer.[ChoicePeer](#) (extends java.awt.peer.[ComponentPeer](#))
 - class java.lang.[Class](#)
 - class java.lang.[ClassLoader](#)
 - interface java.lang.[Cloneable](#)
 - class java.awt.[Color](#)
 - class java.awt.image.[ColorModel](#)
 - class java.awt.image.[DirectColorModel](#)
 - class java.awt.image.[IndexColorModel](#)
 - class java.lang.[Compiler](#)
 - class java.awt.[Component](#) (implements java.awt.image.[ImageObserver](#))
 - class java.awt.[Button](#)
 - class java.awt.[Canvas](#)
 - class java.awt.[Checkbox](#)
 - class java.awt.[Choice](#)
 - class java.awt.[Container](#)
 - class java.awt.[Panel](#)
 - class java.applet.[Applet](#)
 - class java.awt.[Window](#)
 - class java.awt.[Dialog](#)
 - class java.awt.[FileDialog](#)
 - class java.awt.[Frame](#) (implements java.awt.[MenuContainer](#))
 - class java.awt.[Label](#)
 - class java.awt.[List](#)
 - class java.awt.[Scrollbar](#)
 - class java.awt.[TextComponent](#)
 - class java.awt.[TextArea](#)
 - class java.awt.[TextField](#)
 - interface java.awt.peer.[ComponentPeer](#)
 - interface sun.tools.java.[Constants](#) (extends sun.tools.java.[RuntimeConstants](#))
 - interface java.awt.peer.[ContainerPeer](#) (extends java.awt.peer.[ComponentPeer](#))
 - class java.net.[ContentHandler](#)
 - interface java.net.[ContentHandlerFactory](#)

- interface java.io.DataInput
- interface java.io.DataOutput
- class java.net.DatagramPacket
- class java.net.DatagramSocket
- class java.util.Date
- interface sun.tools.debug.DebuggerCallback
- interface java.awt.peer.DialogPeer (extends java.awt.peer.WindowPeer)
- class java.util.Dictionary
 - class java.util.Hashtable (implements java.lang. Cloneable)
 - class java.util.Properties
- class java.awt.Dimension
- interface java.util.Enumeration
- class java.awt.Event
- class sun.tools.debug.Field
 - class sun.tools.debug.RemoteField (implements sun.tools.debug.AgentConstants)
- class java.io.File
- class java.io.FileDescriptor
- interface java.awt.peer.FileDialogPeer (extends java.awt.peer.DialogPeer)
- interface java.io.FilenameFilter
- class java.awt.image.FilteredImageSource (implements java.awt.image. ImageProducer)
- class java.awt.FlowLayout (implements java.awt.LayoutManager)
- class java.awt.Font
- class java.awt.FontMetrics
- interface java.awt.peer.FramePeer (extends java.awt.peer.WindowPeer)
- class java.awt.Graphics
- class java.awt.GridBagConstraints (implements java.lang.Cloneable)
- class java.awt.GridBagLayout (implements java.awt.LayoutManager)
- class java.awt.GridLayout (implements java.awt.LayoutManager)
- class java.awt.Image
- interface java.awt.image.ImageConsumer
- class java.awt.image.ImageFilter (implements java.awt.image.ImageConsumer , java.lang.Cloneable)
 - class java.awt.image.CropImageFilter
 - class java.awt.image.RGBImageFilter
- interface java.awt.image.ImageObserver
- interface java.awt.image.ImageProducer
- class java.net.InetAddress
- class java.io.InputStream
 - class java.io.ByteArrayInputStream
 - class java.io.FileInputStream
 - class java.io.FilterInputStream
 - class java.io.BufferedInputStream
 - class java.io.DataInputStream (implements java.io.DataInput)
 - class java.io.LineNumberInputStream
 - class java.io.PushbackInputStream
 - class java.io.PipedInputStream
 - class java.io.SequenceInputStream
 - class java.io.StringBufferInputStream
- class java.awt.Insets (implements java.lang. Cloneable)
- interface java.awt.peer.LabelPeer (extends java.awt.peer.ComponentPeer)
- interface java.awt.LayoutManager
- interface java.awt.peer.ListPeer (extends java.awt.peer.ComponentPeer)
- class sun.tools.debug.LocalVariable
 - class sun.tools.debug.RemoteStackVariable
- class java.lang.Math
- class java.awt.MediaTracker

- class java.awt.image.MemoryImageSource (implements java.awt.image.ImageProducer)
- interface java.awt.peer.MenuBarPeer (extends java.awt.peer.MenuComponentPeer)
- class java.awt.MenuComponent
 - class java.awt.MenuBar (implements java.awt. MenuContainer)
 - class java.awt.MenuItem
 - class java.awt.CheckboxMenuItem
 - class java.awt.Menu (implements java.awt.MenuContainer)
- interface java.awt.peer.MenuComponentPeer
- interface java.awt.MenuContainer
- interface java.awt.peer.MenuItemPeer (extends java.awt.peer.MenuComponentPeer)
- interface java.awt.peer.MenuPeer (extends java.awt.peer.MenuItemPeer)
- class java.lang.Number
 - class java.lang.Double
 - class java.lang.Float
 - class java.lang.Integer
 - class java.lang.Long
- class java.util.Observable
- interface java.util.Observer
- class java.io.OutputStream
 - class java.io.ByteArrayOutputStream
 - class java.io.FileOutputStream
 - class java.io.FilterOutputStream
 - class java.io.BufferedOutputStream
 - class java.io.DataOutputStream (implements java.io.DataOutput)
 - class java.io.PrintStream
 - class java.io.PipedOutputStream
- interface java.awt.peer.PanelPeer (extends java.awt.peer.ContainerPeer)
- class java.awt.image.PixelGrabber (implements java.awt.image.ImageConsumer)
- class java.awt.Point
- class java.awt.Polygon
- class java.lang.Process
- class java.util.Random
- class java.io.RandomAccessFile (implements java.io.DataOutput , java.io.DataInput)
- class java.awt.Rectangle
- class sun.tools.debug.RemoteDebugger
- class sun.tools.debug.RemoteValue (implements sun.tools.debug.AgentConstants)
 - class sun.tools.debug.RemoteBoolean
 - class sun.tools.debug.RemoteByte
 - class sun.tools.debug.RemoteChar
 - class sun.tools.debug.RemoteDouble
 - class sun.tools.debug.RemoteFloat
 - class sun.tools.debug.RemoteInt
 - class sun.tools.debug.RemoteLong
 - class sun.tools.debug.RemoteObject
 - class sun.tools.debug.RemoteArray
 - class sun.tools.debug.RemoteClass
 - class sun.tools.debug.RemoteString
 - class sun.tools.debug.RemoteThread
 - class sun.tools.debug.RemoteThreadGroup
 - class sun.tools.debug.RemoteShort
- interface java.lang.Runnable
- class java.lang.Runtime
- interface sun.tools.java.RuntimeConstants
- interface java.awt.peer.ScrollbarPeer (extends java.awt.peer.ComponentPeer)
- class java.lang.SecurityManager
- class java.net.ServerSocket

- class java.net.Socket
- class java.net.SocketImpl
- interface java.net.SocketImplFactory
- class sun.tools.debug.StackFrame
 - class sun.tools.debug.RemoteStackFrame
- class java.io.StreamTokenizer
- class java.lang.String
- class java.lang.StringBuffer
- class java.util.StringTokenizer (implements java.util.Enumeration)
- class java.lang.System
- interface java.awt.peer.TextAreaPeer (extends java.awt.peer.TextComponentPeer)
- interface java.awt.peer.TextComponentPeer (extends java.awt.peer.ComponentPeer)
- interface java.awt.peer.TextFieldPeer (extends java.awt.peer.TextComponentPeer)
- class java.lang.Thread (implements java.lang.Runnable)
- class java.lang.ThreadGroup
- class java.lang.Throwable
 - class java.lang.Error
 - class java.awt.AWTError
 - class java.lang.LinkageError
 - class java.lang.ClassCircularityError
 - class java.lang.ClassFormatError
 - class java.lang.IncompatibleClassChangeError
 - class java.lang.AbstractMethodError
 - class java.lang.IllegalAccessError
 - class java.lang.InstantiationError
 - class java.lang.NoSuchFieldError
 - class java.lang.NoSuchMethodError
 - class java.lang.NoClassDefFoundError
 - class java.lang.UnsatisfiedLinkError
 - class java.lang.VerifyError
 - class java.lang.ThreadDeath
 - class java.lang.VirtualMachineError
 - class java.lang.InternalError
 - class java.lang.OutOfMemoryError
 - class java.lang.StackOverflowError
 - class java.lang.UnknownError
 - class java.lang.Exception
 - class java.awt.AWTException
 - class java.lang.ClassNotFoundException
 - class java.lang.CloneNotSupportedException
 - class java.io.IOException
 - class java.io.EOFException
 - class java.io.FileNotFoundException
 - class java.io.InterruptedIOException
 - class java.net.MalformedURLException
 - class java.net.ProtocolException
 - class java.net.SocketException
 - class java.io.UTFDataFormatException
 - class java.net.UnknownHostException
 - class java.net.UnknownServiceException
 - class java.lang.IllegalAccessException
 - class java.lang.InstantiationException
 - class java.lang.InterruptedException
 - class java.lang.NoSuchMethodException
 - class java.lang.RuntimeException
 - class java.lang.ArithmeticException

- class java.lang.ArrayStoreException
- class java.lang.ClassCastException
- class java.util.EmptyStackException
- class java.lang.IllegalArgumentException
 - class java.lang.IllegalThreadStateException
 - class java.lang.NumberFormatException
- class java.lang.IllegalMonitorStateException
- class java.lang.IndexOutOfBoundsException
 - class java.lang.ArrayIndexOutOfBoundsException
 - class java.lang.StringIndexOutOfBoundsException
- class java.lang.NegativeArraySizeException
- class java.util.NoSuchElementException
- class java.lang.NullPointerException
- class java.lang.SecurityException
- class java.awt.Toolkit
- class java.net.URL
- class java.net.URLConnection
- class java.net.URLEncoder
- class java.net.URLStreamHandler
- interface java.net.URLStreamHandlerFactory
- class java.util.Vector (implements java.lang.Cloneable)
 - class java.util.Stack
- interface java.awt.peer.WindowPeer (extends java.awt.peer.ContainerPeer)

file:/D:/APIGOLD/ALLNAM~1.HTM#StackFrame()

file:/D:/APIGOLD/ALLNAM~1.HTM#toString()

file:/D:/APIGOLD/AVAU~14.HTM

file:/D:/APIGOLD/JAVAA~22.HTM#drawImage(java.awt.Image, int, int, int, int,
java.awt.image.ImageObserver)

file:/D:/APIGOLD/JAVAA~33.HM

file:/D:/APIGOLD/JAVAA~33.HM#HEIGHT

URL

file:/D:/APIGOLD/JAVAL~53.HTM#capacit()

file:/D:/APIGOLD/JAVA~58.HTM

file:/D:/APIGOLD/SUNTO~13HTM

