| | [**Overview**](http://docs.google.com/overview-summary.html) | Package | Class | Use | [**Tree**](http://docs.google.com/overview-tree.html) | **Deprecated** | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV   NEXT | [**FRAMES**](http://docs.google.com/index.html?deprecated-list.html)    [**NO FRAMES**](http://docs.google.com/deprecated-list.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Deprecated API**

**Contents**

* [Deprecated Interfaces](#3znysh7)
* [Deprecated Classes](#2et92p0)
* [Deprecated Exceptions](#tyjcwt)
* [Deprecated Annotation Types](#3dy6vkm)
* [Deprecated Fields](#1t3h5sf)
* [Deprecated Methods](#4d34og8)
* [Deprecated Constructors](#2s8eyo1)
* [Deprecated Annotation Type Elements](#17dp8vu)

| **Deprecated Interfaces** | |
| --- | --- |
| [org.xml.sax.AttributeList](http://docs.google.com/org/xml/sax/AttributeList.html)  *This interface has been replaced by the SAX2* [*Attributes*](http://docs.google.com/org/xml/sax/Attributes.html) *interface, which includes Namespace support.* |
| [java.security.Certificate](http://docs.google.com/java/security/Certificate.html)  *A new certificate handling package is created in the Java platform. This Certificate interface is entirely deprecated and is here to allow for a smooth transition to the new package.* |
| [org.xml.sax.DocumentHandler](http://docs.google.com/org/xml/sax/DocumentHandler.html)  *This interface has been replaced by the SAX2* [*ContentHandler*](http://docs.google.com/org/xml/sax/ContentHandler.html) *interface, which includes Namespace support.* |
| [org.omg.CORBA.DynAny](http://docs.google.com/org/omg/CORBA/DynAny.html)  *Use the new* [*DynAny*](http://docs.google.com/org/omg/DynamicAny/DynAny.html) *instead* |
| [org.omg.CORBA.DynArray](http://docs.google.com/org/omg/CORBA/DynArray.html)  *Use the new* [*DynArray*](http://docs.google.com/org/omg/DynamicAny/DynArray.html) *instead* |
| [org.omg.CORBA.DynEnum](http://docs.google.com/org/omg/CORBA/DynEnum.html)  *Use the new* [*DynEnum*](http://docs.google.com/org/omg/DynamicAny/DynEnum.html) *instead* |
| [org.omg.CORBA.DynFixed](http://docs.google.com/org/omg/CORBA/DynFixed.html)  *Use the new* [*DynFixed*](http://docs.google.com/org/omg/DynamicAny/DynFixed.html) *instead* |
| [org.omg.CORBA.DynSequence](http://docs.google.com/org/omg/CORBA/DynSequence.html)  *Use the new* [*DynSequence*](http://docs.google.com/org/omg/DynamicAny/DynSequence.html) *instead* |
| [org.omg.CORBA.DynStruct](http://docs.google.com/org/omg/CORBA/DynStruct.html)  *Use the new* [*DynStruct*](http://docs.google.com/org/omg/DynamicAny/DynStruct.html) *instead* |
| [org.omg.CORBA.DynUnion](http://docs.google.com/org/omg/CORBA/DynUnion.html)  *Use the new* [*DynUnion*](http://docs.google.com/org/omg/DynamicAny/DynUnion.html) *instead* |
| [org.omg.CORBA.DynValue](http://docs.google.com/org/omg/CORBA/DynValue.html)  *Use the new* [*DynValue*](http://docs.google.com/org/omg/DynamicAny/DynValue.html) *instead* |
| [java.rmi.server.LoaderHandler](http://docs.google.com/java/rmi/server/LoaderHandler.html)  *no replacement* |
| [org.xml.sax.Parser](http://docs.google.com/org/xml/sax/Parser.html)  *This interface has been replaced by the SAX2* [*XMLReader*](http://docs.google.com/org/xml/sax/XMLReader.html) *interface, which includes Namespace support.* |
| [java.rmi.registry.RegistryHandler](http://docs.google.com/java/rmi/registry/RegistryHandler.html)  *no replacement* |
| [java.rmi.server.RemoteCall](http://docs.google.com/java/rmi/server/RemoteCall.html)  *no replacement.* |
| [java.rmi.server.Skeleton](http://docs.google.com/java/rmi/server/Skeleton.html)  *no replacement. Skeletons are no longer required for remote method calls in the Java 2 platform v1.2 and greater.* |
| [javax.xml.bind.Validator](http://docs.google.com/javax/xml/bind/Validator.html)  *since JAXB 2.0* |

| **Deprecated Classes** | |
| --- | --- |
| [javax.accessibility.AccessibleResourceBundle](http://docs.google.com/javax/accessibility/AccessibleResourceBundle.html)  *This class is deprecated as of version 1.3 of the Java Platform.* |
| [org.xml.sax.helpers.AttributeListImpl](http://docs.google.com/org/xml/sax/helpers/AttributeListImpl.html)  *This class implements a deprecated interface,* [*AttributeList*](http://docs.google.com/org/xml/sax/AttributeList.html)*; that interface has been replaced by* [*Attributes*](http://docs.google.com/org/xml/sax/Attributes.html)*, which is implemented in the* [*AttributesImpl*](http://docs.google.com/org/xml/sax/helpers/AttributesImpl.html) *helper class.* |
| [javax.management.DefaultLoaderRepository](http://docs.google.com/javax/management/DefaultLoaderRepository.html)  *Use* [*MBeanServer.getClassLoaderRepository()*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository()) *instead.* |
| [javax.management.loading.DefaultLoaderRepository](http://docs.google.com/javax/management/loading/DefaultLoaderRepository.html)  *Use* [*MBeanServer.getClassLoaderRepository()*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository())*} instead.* |
| [javax.swing.text.DefaultTextUI](http://docs.google.com/javax/swing/text/DefaultTextUI.html) |
| [org.omg.CORBA.DynamicImplementation](http://docs.google.com/org/omg/CORBA/DynamicImplementation.html)  *org.omg.CORBA.DynamicImplementation* |
| [org.xml.sax.HandlerBase](http://docs.google.com/org/xml/sax/HandlerBase.html)  *This class works with the deprecated* [*DocumentHandler*](http://docs.google.com/org/xml/sax/DocumentHandler.html) *interface. It has been replaced by the SAX2* [*DefaultHandler*](http://docs.google.com/org/xml/sax/helpers/DefaultHandler.html) *class.* |
| [java.security.Identity](http://docs.google.com/java/security/Identity.html)  *This class is no longer used. Its functionality has been replaced by java.security.KeyStore, the java.security.cert package, and java.security.Principal.* |
| [java.security.IdentityScope](http://docs.google.com/java/security/IdentityScope.html)  *This class is no longer used. Its functionality has been replaced by java.security.KeyStore, the java.security.cert package, and java.security.Principal.* |
| [java.io.LineNumberInputStream](http://docs.google.com/java/io/LineNumberInputStream.html)  *This class incorrectly assumes that bytes adequately represent characters. As of JDK 1.1, the preferred way to operate on character streams is via the new character-stream classes, which include a class for counting line numbers.* |
| [java.rmi.server.LogStream](http://docs.google.com/java/rmi/server/LogStream.html)  *no replacement* |
| [javax.swing.plaf.metal.MetalComboBoxUI.MetalComboPopup](http://docs.google.com/javax/swing/plaf/metal/MetalComboBoxUI.MetalComboPopup.html)  *As of Java 2 platform v1.4.* |
| [java.rmi.server.Operation](http://docs.google.com/java/rmi/server/Operation.html)  *no replacement* |
| [org.xml.sax.helpers.ParserFactory](http://docs.google.com/org/xml/sax/helpers/ParserFactory.html)  *This class works with the deprecated* [*Parser*](http://docs.google.com/org/xml/sax/Parser.html) *interface.* |
| [javax.security.auth.Policy](http://docs.google.com/javax/security/auth/Policy.html)  *as of JDK version 1.4 -- Replaced by java.security.Policy. java.security.Policy has a method:*  *public PermissionCollection getPermissions  (java.security.ProtectionDomain pd)*  *and ProtectionDomain has a constructor:*  *public ProtectionDomain  (CodeSource cs,  PermissionCollection permissions,  ClassLoader loader,  Principal[] principals)*  *These two APIs provide callers the means to query the Policy for Principal-based Permission entries.* |
| [org.omg.CORBA.Principal](http://docs.google.com/org/omg/CORBA/Principal.html)  *Deprecated by CORBA 2.2.* |
| [org.omg.CORBA.PrincipalHolder](http://docs.google.com/org/omg/CORBA/PrincipalHolder.html)  *Deprecated by CORBA 2.2.* |
| [java.security.Signer](http://docs.google.com/java/security/Signer.html)  *This class is no longer used. Its functionality has been replaced by java.security.KeyStore, the java.security.cert package, and java.security.Principal.* |
| [javax.xml.soap.SOAPElementFactory](http://docs.google.com/javax/xml/soap/SOAPElementFactory.html)  *- Use javax.xml.soap.SOAPFactory for creating SOAPElements.* |
| [java.io.StringBufferInputStream](http://docs.google.com/java/io/StringBufferInputStream.html)  *This class does not properly convert characters into bytes. As of JDK 1.1, the preferred way to create a stream from a string is via the StringReader class.* |
| [javax.swing.text.TableView.TableCell](http://docs.google.com/javax/swing/text/TableView.TableCell.html)  *A table cell can now be any View implementation.* |

| **Deprecated Exceptions** | |
| --- | --- |
| [java.rmi.RMISecurityException](http://docs.google.com/java/rmi/RMISecurityException.html)  *Use* [*SecurityException*](http://docs.google.com/java/lang/SecurityException.html) *instead. Application code should never directly reference this class, and RMISecurityManager no longer throws this subclass of java.lang.SecurityException.* |
| [java.rmi.ServerRuntimeException](http://docs.google.com/java/rmi/ServerRuntimeException.html)  *no replacement* |
| [java.rmi.server.SkeletonMismatchException](http://docs.google.com/java/rmi/server/SkeletonMismatchException.html)  *no replacement. Skeletons are no longer required for remote method calls in the Java 2 platform v1.2 and greater.* |
| [java.rmi.server.SkeletonNotFoundException](http://docs.google.com/java/rmi/server/SkeletonNotFoundException.html)  *no replacement. Skeletons are no longer required for remote method calls in the Java 2 platform v1.2 and greater.* |

| **Deprecated Annotation Types** | |
| --- | --- |
| [javax.jws.soap.InitParam](http://docs.google.com/javax/jws/soap/InitParam.html)  *As of JSR-181 2.0 with no replacement.* |
| [javax.jws.soap.SOAPMessageHandler](http://docs.google.com/javax/jws/soap/SOAPMessageHandler.html)  *As of JSR-181 2.0 with no replacement.* |
| [javax.jws.soap.SOAPMessageHandlers](http://docs.google.com/javax/jws/soap/SOAPMessageHandlers.html)  *As of JSR-181 2.0 with no replacement.* |

| **Deprecated Fields** | |
| --- | --- |
| [javax.management.openmbean.OpenType.ALLOWED\_CLASSNAMES](http://docs.google.com/javax/management/openmbean/OpenType.html#ALLOWED_CLASSNAMES)  *Use* [*ALLOWED\_CLASSNAMES\_LIST*](http://docs.google.com/javax/management/openmbean/OpenType.html#ALLOWED_CLASSNAMES_LIST) *instead.* |
| [javax.management.monitor.Monitor.alreadyNotified](http://docs.google.com/javax/management/monitor/Monitor.html#alreadyNotified)  *equivalent to* [*Monitor.alreadyNotifieds*](http://docs.google.com/javax/management/monitor/Monitor.html#alreadyNotifieds)*[0].* |
| [javax.swing.JViewport.backingStore](http://docs.google.com/javax/swing/JViewport.html#backingStore)  *As of Java 2 platform v1.3* |
| [javax.swing.plaf.basic.BasicDesktopPaneUI.closeKey](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.html#closeKey)  *As of 1.3.* |
| [java.awt.Frame.CROSSHAIR\_CURSOR](http://docs.google.com/java/awt/Frame.html#CROSSHAIR_CURSOR)  *replaced by Cursor.CROSSHAIR\_CURSOR.* |
| [javax.management.monitor.Monitor.dbgTag](http://docs.google.com/javax/management/monitor/Monitor.html#dbgTag)  *No replacement.* |
| [java.awt.Frame.DEFAULT\_CURSOR](http://docs.google.com/java/awt/Frame.html#DEFAULT_CURSOR)  *replaced by Cursor.DEFAULT\_CURSOR.* |
| [javax.swing.JRootPane.defaultPressAction](http://docs.google.com/javax/swing/JRootPane.html#defaultPressAction)  *As of Java 2 platform v1.3.* |
| [javax.swing.JRootPane.defaultReleaseAction](http://docs.google.com/javax/swing/JRootPane.html#defaultReleaseAction)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.dividerResizeToggleKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#dividerResizeToggleKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicToolBarUI.downKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#downKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicTabbedPaneUI.downKey](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#downKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.downKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#downKey)  *As of Java 2 platform v1.3.* |
| [java.awt.Frame.E\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#E_RESIZE_CURSOR)  *replaced by Cursor.E\_RESIZE\_CURSOR.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.endKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#endKey)  *As of Java 2 platform v1.3.* |
| [java.util.logging.Logger.global](http://docs.google.com/java/util/logging/Logger.html#global)  *Initialization of this field is prone to deadlocks. The field must be initialized by the Logger class initialization which may cause deadlocks with the LogManager class initialization. In such cases two class initialization wait for each other to complete. As of JDK version 1.6, the preferred way to get the global logger object is via the call Logger.getLogger(Logger.GLOBAL\_LOGGER\_NAME).* |
| [java.awt.Frame.HAND\_CURSOR](http://docs.google.com/java/awt/Frame.html#HAND_CURSOR)  *replaced by Cursor.HAND\_CURSOR.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.homeKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#homeKey)  *As of Java 2 platform v1.3.* |
| [java.net.HttpURLConnection.HTTP\_SERVER\_ERROR](http://docs.google.com/java/net/HttpURLConnection.html#HTTP_SERVER_ERROR)  *it is misplaced and shouldn't have existed.* |
| [java.lang.SecurityManager.inCheck](http://docs.google.com/java/lang/SecurityManager.html#inCheck)  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.keyboardDownRightListener](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#keyboardDownRightListener)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.keyboardEndListener](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#keyboardEndListener)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.keyboardHomeListener](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#keyboardHomeListener)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.keyboardResizeToggleListener](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#keyboardResizeToggleListener)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.keyboardUpLeftListener](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#keyboardUpLeftListener)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicToolBarUI.leftKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#leftKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicTabbedPaneUI.leftKey](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#leftKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.leftKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#leftKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicDesktopPaneUI.maximizeKey](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.html#maximizeKey)  *As of 1.3.* |
| [javax.swing.plaf.basic.BasicDesktopPaneUI.minimizeKey](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.html#minimizeKey)  *As of 1.3.* |
| [java.awt.Frame.MOVE\_CURSOR](http://docs.google.com/java/awt/Frame.html#MOVE_CURSOR)  *replaced by Cursor.MOVE\_CURSOR.* |
| [java.awt.Frame.N\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#N_RESIZE_CURSOR)  *replaced by Cursor.N\_RESIZE\_CURSOR.* |
| [javax.swing.plaf.basic.BasicDesktopPaneUI.navigateKey](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.html#navigateKey)  *As of 1.3.* |
| [javax.swing.plaf.basic.BasicDesktopPaneUI.navigateKey2](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.html#navigateKey2)  *As of 1.3.* |
| [java.awt.Frame.NE\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#NE_RESIZE_CURSOR)  *replaced by Cursor.NE\_RESIZE\_CURSOR.* |
| [java.awt.Frame.NW\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#NW_RESIZE_CURSOR)  *replaced by Cursor.NW\_RESIZE\_CURSOR.* |
| [javax.swing.plaf.basic.BasicInternalFrameUI.openMenuKey](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameUI.html#openMenuKey)  *As of Java 2 platform v1.3.* |
| [java.awt.datatransfer.DataFlavor.plainTextFlavor](http://docs.google.com/java/awt/datatransfer/DataFlavor.html#plainTextFlavor)  *as of 1.3. Use DataFlavor.getReaderForText(Transferable) instead of Transferable.getTransferData(DataFlavor.plainTextFlavor).* |
| [javax.swing.text.html.FormView.RESET](http://docs.google.com/javax/swing/text/html/FormView.html#RESET)  *As of 1.3, value comes from UIManager UIManager property FormView.resetButtonText* |
| [javax.swing.table.TableColumn.resizedPostingDisableCount](http://docs.google.com/javax/swing/table/TableColumn.html#resizedPostingDisableCount)  *as of Java 2 platform v1.3* |
| [javax.swing.plaf.basic.BasicToolBarUI.rightKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#rightKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicTabbedPaneUI.rightKey](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#rightKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.rightKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#rightKey)  *As of Java 2 platform v1.3.* |
| [java.awt.Frame.S\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#S_RESIZE_CURSOR)  *replaced by Cursor.S\_RESIZE\_CURSOR.* |
| [java.awt.Frame.SE\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#SE_RESIZE_CURSOR)  *replaced by Cursor.SE\_RESIZE\_CURSOR.* |
| [javax.swing.text.html.FormView.SUBMIT](http://docs.google.com/javax/swing/text/html/FormView.html#SUBMIT)  *As of 1.3, value now comes from UIManager property FormView.submitButtonText* |
| [java.lang.Character.UnicodeBlock.SURROGATES\_AREA](http://docs.google.com/java/lang/Character.UnicodeBlock.html#SURROGATES_AREA)  *As of J2SE 5, use* [*Character.UnicodeBlock.HIGH\_SURROGATES*](http://docs.google.com/java/lang/Character.UnicodeBlock.html#HIGH_SURROGATES)*,* [*Character.UnicodeBlock.HIGH\_PRIVATE\_USE\_SURROGATES*](http://docs.google.com/java/lang/Character.UnicodeBlock.html#HIGH_PRIVATE_USE_SURROGATES)*, and* [*Character.UnicodeBlock.LOW\_SURROGATES*](http://docs.google.com/java/lang/Character.UnicodeBlock.html#LOW_SURROGATES)*. These new constants match the block definitions of the Unicode Standard. The* [*Character.UnicodeBlock.of(char)*](http://docs.google.com/java/lang/Character.UnicodeBlock.html#of(char)) *and* [*Character.UnicodeBlock.of(int)*](http://docs.google.com/java/lang/Character.UnicodeBlock.html#of(int)) *methods return the new constants, not SURROGATES\_AREA.* |
| [java.awt.Frame.SW\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#SW_RESIZE_CURSOR)  *replaced by Cursor.SW\_RESIZE\_CURSOR.* |
| [java.awt.Frame.TEXT\_CURSOR](http://docs.google.com/java/awt/Frame.html#TEXT_CURSOR)  *replaced by Cursor.TEXT\_CURSOR.* |
| [javax.swing.plaf.basic.BasicToolBarUI.upKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#upKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicTabbedPaneUI.upKey](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#upKey)  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.upKey](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#upKey)  *As of Java 2 platform v1.3.* |
| [java.awt.Frame.W\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#W_RESIZE_CURSOR)  *replaced by Cursor.W\_RESIZE\_CURSOR.* |
| [java.awt.Frame.WAIT\_CURSOR](http://docs.google.com/java/awt/Frame.html#WAIT_CURSOR)  *replaced by Cursor.WAIT\_CURSOR.* |

| **Deprecated Methods** | |
| --- | --- |
| [java.awt.Component.action(Event, Object)](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object))  *As of JDK version 1.1, should register this component as ActionListener on component which fires action events.* |
| [java.awt.List.addItem(String)](http://docs.google.com/java/awt/List.html#addItem(java.lang.String))  *replaced by add(String).* |
| [java.awt.List.addItem(String, int)](http://docs.google.com/java/awt/List.html#addItem(java.lang.String,%20int))  *replaced by add(String, int).* |
| [java.awt.CardLayout.addLayoutComponent(String, Component)](http://docs.google.com/java/awt/CardLayout.html#addLayoutComponent(java.lang.String,%20java.awt.Component))  *replaced by addLayoutComponent(Component, Object).* |
| [java.awt.BorderLayout.addLayoutComponent(String, Component)](http://docs.google.com/java/awt/BorderLayout.html#addLayoutComponent(java.lang.String,%20java.awt.Component))  *replaced by addLayoutComponent(Component, Object).* |
| [java.awt.List.allowsMultipleSelections()](http://docs.google.com/java/awt/List.html#allowsMultipleSelections())  *As of JDK version 1.1, replaced by isMultipleMode().* |
| [java.lang.ThreadGroup.allowThreadSuspension(boolean)](http://docs.google.com/java/lang/ThreadGroup.html#allowThreadSuspension(boolean))  *The definition of this call depends on* [*ThreadGroup.suspend()*](http://docs.google.com/java/lang/ThreadGroup.html#suspend())*, which is deprecated. Further, the behavior of this call was never specified.* |
| [java.awt.TextArea.appendText(String)](http://docs.google.com/java/awt/TextArea.html#appendText(java.lang.String))  *As of JDK version 1.1, replaced by append(String).* |
| [java.awt.Window.applyResourceBundle(ResourceBundle)](http://docs.google.com/java/awt/Window.html#applyResourceBundle(java.util.ResourceBundle))  *As of J2SE 1.4, replaced by* [*Component.applyComponentOrientation*](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation))*.* |
| [java.awt.Window.applyResourceBundle(String)](http://docs.google.com/java/awt/Window.html#applyResourceBundle(java.lang.String))  *As of J2SE 1.4, replaced by* [*Component.applyComponentOrientation*](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation))*.* |
| [java.awt.Component.bounds()](http://docs.google.com/java/awt/Component.html#bounds())  *As of JDK version 1.1, replaced by getBounds().* |
| [java.lang.SecurityManager.checkMulticast(InetAddress, byte)](http://docs.google.com/java/lang/SecurityManager.html#checkMulticast(java.net.InetAddress,%20byte))  *Use #checkPermission(java.security.Permission) instead* |
| [java.lang.SecurityManager.classDepth(String)](http://docs.google.com/java/lang/SecurityManager.html#classDepth(java.lang.String))  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [java.lang.SecurityManager.classLoaderDepth()](http://docs.google.com/java/lang/SecurityManager.html#classLoaderDepth())  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [java.awt.List.clear()](http://docs.google.com/java/awt/List.html#clear())  *As of JDK version 1.1, replaced by removeAll().* |
| [java.awt.image.renderable.RenderContext.concetenateTransform(AffineTransform)](http://docs.google.com/java/awt/image/renderable/RenderContext.html#concetenateTransform(java.awt.geom.AffineTransform))  *replaced by concatenateTransform(AffineTransform).* |
| [java.awt.Container.countComponents()](http://docs.google.com/java/awt/Container.html#countComponents())  *As of JDK version 1.1, replaced by getComponentCount().* |
| [java.awt.Menu.countItems()](http://docs.google.com/java/awt/Menu.html#countItems())  *As of JDK version 1.1, replaced by getItemCount().* |
| [java.awt.Choice.countItems()](http://docs.google.com/java/awt/Choice.html#countItems())  *As of JDK version 1.1, replaced by getItemCount().* |
| [java.awt.List.countItems()](http://docs.google.com/java/awt/List.html#countItems())  *As of JDK version 1.1, replaced by getItemCount().* |
| [java.awt.MenuBar.countMenus()](http://docs.google.com/java/awt/MenuBar.html#countMenus())  *As of JDK version 1.1, replaced by getMenuCount().* |
| [java.lang.Thread.countStackFrames()](http://docs.google.com/java/lang/Thread.html#countStackFrames())  *The definition of this call depends on* [*Thread.suspend()*](http://docs.google.com/java/lang/Thread.html#suspend())*, which is deprecated. Further, the results of this call were never well-defined.* |
| [org.omg.CORBA.ORB.create\_basic\_dyn\_any(TypeCode)](http://docs.google.com/org/omg/CORBA/ORB.html#create_basic_dyn_any(org.omg.CORBA.TypeCode))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_dyn\_any(Any)](http://docs.google.com/org/omg/CORBA/ORB.html#create_dyn_any(org.omg.CORBA.Any))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_dyn\_array(TypeCode)](http://docs.google.com/org/omg/CORBA/ORB.html#create_dyn_array(org.omg.CORBA.TypeCode))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_dyn\_enum(TypeCode)](http://docs.google.com/org/omg/CORBA/ORB.html#create_dyn_enum(org.omg.CORBA.TypeCode))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_dyn\_sequence(TypeCode)](http://docs.google.com/org/omg/CORBA/ORB.html#create_dyn_sequence(org.omg.CORBA.TypeCode))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_dyn\_struct(TypeCode)](http://docs.google.com/org/omg/CORBA/ORB.html#create_dyn_struct(org.omg.CORBA.TypeCode))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_dyn\_union(TypeCode)](http://docs.google.com/org/omg/CORBA/ORB.html#create_dyn_union(org.omg.CORBA.TypeCode))  *Use the new* [*DynAnyFactory*](http://docs.google.com/org/omg/DynamicAny/DynAnyFactory.html) *API instead* |
| [org.omg.CORBA.ORB.create\_recursive\_sequence\_tc(int, int)](http://docs.google.com/org/omg/CORBA/ORB.html#create_recursive_sequence_tc(int,%20int))  *Use a combination of create\_recursive\_tc and create\_sequence\_tc instead* |
| [javax.xml.soap.SOAPElementFactory.create(Name)](http://docs.google.com/javax/xml/soap/SOAPElementFactory.html#create(javax.xml.soap.Name))  *Use javax.xml.soap.SOAPFactory.createElement(javax.xml.soap.Name) instead* |
| [javax.xml.soap.SOAPElementFactory.create(String)](http://docs.google.com/javax/xml/soap/SOAPElementFactory.html#create(java.lang.String))  *Use javax.xml.soap.SOAPFactory.createElement(String localName) instead* |
| [javax.xml.soap.SOAPElementFactory.create(String, String, String)](http://docs.google.com/javax/xml/soap/SOAPElementFactory.html#create(java.lang.String,%20java.lang.String,%20java.lang.String))  *Use javax.xml.soap.SOAPFactory.createElement(String localName, String prefix, String uri) instead* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.createKeyboardDownRightListener()](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#createKeyboardDownRightListener())  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.createKeyboardEndListener()](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#createKeyboardEndListener())  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.createKeyboardHomeListener()](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#createKeyboardHomeListener())  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.createKeyboardResizeToggleListener()](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#createKeyboardResizeToggleListener())  *As of Java 2 platform v1.3.* |
| [javax.swing.plaf.basic.BasicSplitPaneUI.createKeyboardUpLeftListener()](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#createKeyboardUpLeftListener())  *As of Java 2 platform v1.3.* |
| [javax.swing.JTable.createScrollPaneForTable(JTable)](http://docs.google.com/javax/swing/JTable.html#createScrollPaneForTable(javax.swing.JTable))  *As of Swing version 1.0.2, replaced by new JScrollPane(aTable).* |
| [javax.swing.text.TableView.createTableCell(Element)](http://docs.google.com/javax/swing/text/TableView.html#createTableCell(javax.swing.text.Element))  *Table cells can now be any arbitrary View implementation and should be produced by the ViewFactory rather than the table.* |
| [javax.xml.bind.JAXBContext.createValidator()](http://docs.google.com/javax/xml/bind/JAXBContext.html#createValidator())  *since JAXB2.0* |
| [java.lang.SecurityManager.currentClassLoader()](http://docs.google.com/java/lang/SecurityManager.html#currentClassLoader())  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [java.lang.SecurityManager.currentLoadedClass()](http://docs.google.com/java/lang/SecurityManager.html#currentLoadedClass())  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [java.net.URLDecoder.decode(String)](http://docs.google.com/java/net/URLDecoder.html#decode(java.lang.String))  *The resulting string may vary depending on the platform's default encoding. Instead, use the decode(String,String) method to specify the encoding.* |
| [java.lang.ClassLoader.defineClass(byte[], int, int)](http://docs.google.com/java/lang/ClassLoader.html#defineClass(byte%5B%5D,%20int,%20int))  *Replaced by* [*defineClass(String, byte[], int, int)*](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)) |
| [java.awt.List.delItem(int)](http://docs.google.com/java/awt/List.html#delItem(int))  *replaced by remove(String) and remove(int).* |
| [java.awt.List.delItems(int, int)](http://docs.google.com/java/awt/List.html#delItems(int,%20int))  *As of JDK version 1.1, Not for public use in the future. This method is expected to be retained only as a package private method.* |
| [java.awt.Container.deliverEvent(Event)](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event))  *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e)* |
| [java.awt.Component.deliverEvent(Event)](http://docs.google.com/java/awt/Component.html#deliverEvent(java.awt.Event))  *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e).* |
| [javax.management.MBeanServer.deserialize(ObjectName, byte[])](http://docs.google.com/javax/management/MBeanServer.html#deserialize(javax.management.ObjectName,%20byte%5B%5D))  *Use* [*getClassLoaderFor*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderFor(javax.management.ObjectName)) *to obtain the appropriate class loader for deserialization.* |
| [javax.management.MBeanServer.deserialize(String, byte[])](http://docs.google.com/javax/management/MBeanServer.html#deserialize(java.lang.String,%20byte%5B%5D))  *Use* [*MBeanServer.getClassLoaderRepository()*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository()) *to obtain the class loader repository and use it to deserialize.* |
| [javax.management.MBeanServer.deserialize(String, ObjectName, byte[])](http://docs.google.com/javax/management/MBeanServer.html#deserialize(java.lang.String,%20javax.management.ObjectName,%20byte%5B%5D))  *Use* [*getClassLoader*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoader(javax.management.ObjectName)) *to obtain the class loader for deserialization.* |
| [java.lang.Thread.destroy()](http://docs.google.com/java/lang/Thread.html#destroy())  *This method was originally designed to destroy this thread without any cleanup. Any monitors it held would have remained locked. However, the method was never implemented. If if were to be implemented, it would be deadlock-prone in much the manner of* [*Thread.suspend()*](http://docs.google.com/java/lang/Thread.html#suspend())*. If the target thread held a lock protecting a critical system resource when it was destroyed, no thread could ever access this resource again. If another thread ever attempted to lock this resource, deadlock would result. Such deadlocks typically manifest themselves as "frozen" processes. For more information, see*  [*Why are Thread.stop, Thread.suspend and Thread.resume Deprecated?*](http://docs.google.com/technotes/guides/concurrency/threadPrimitiveDeprecation.html)*.* |
| [java.awt.MenuItem.disable()](http://docs.google.com/java/awt/MenuItem.html#disable())  *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| [java.awt.Component.disable()](http://docs.google.com/java/awt/Component.html#disable())  *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| [javax.swing.JComponent.disable()](http://docs.google.com/javax/swing/JComponent.html#disable())  *As of JDK version 1.1, replaced by java.awt.Component.setEnabled(boolean).* |
| [javax.swing.table.TableColumn.disableResizedPosting()](http://docs.google.com/javax/swing/table/TableColumn.html#disableResizedPosting())  *as of Java 2 platform v1.3* |
| [javax.swing.FocusManager.disableSwingFocusManager()](http://docs.google.com/javax/swing/FocusManager.html#disableSwingFocusManager())  *as of 1.4, replaced by KeyboardFocusManager.setDefaultFocusTraversalPolicy(FocusTraversalPolicy)* |
| [java.rmi.server.Skeleton.dispatch(Remote, RemoteCall, int, long)](http://docs.google.com/java/rmi/server/Skeleton.html#dispatch(java.rmi.Remote,%20java.rmi.server.RemoteCall,%20int,%20long))  *no replacement* |
| [java.rmi.server.RemoteCall.done()](http://docs.google.com/java/rmi/server/RemoteCall.html#done())  *no replacement* |
| [java.rmi.server.RemoteRef.done(RemoteCall)](http://docs.google.com/java/rmi/server/RemoteRef.html#done(java.rmi.server.RemoteCall))  *1.2 style stubs no longer use this method. Instead of using a sequence of method calls to the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.* |
| [javax.swing.plaf.metal.MetalComboBoxUI.editablePropertyChanged(PropertyChangeEvent)](http://docs.google.com/javax/swing/plaf/metal/MetalComboBoxUI.html#editablePropertyChanged(java.beans.PropertyChangeEvent))  *As of Java 2 platform v1.4.* |
| [java.awt.MenuItem.enable()](http://docs.google.com/java/awt/MenuItem.html#enable())  *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| [java.awt.Component.enable()](http://docs.google.com/java/awt/Component.html#enable())  *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| [javax.swing.JComponent.enable()](http://docs.google.com/javax/swing/JComponent.html#enable())  *As of JDK version 1.1, replaced by java.awt.Component.setEnabled(boolean).* |
| [java.awt.MenuItem.enable(boolean)](http://docs.google.com/java/awt/MenuItem.html#enable(boolean))  *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| [java.awt.Component.enable(boolean)](http://docs.google.com/java/awt/Component.html#enable(boolean))  *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| [javax.swing.table.TableColumn.enableResizedPosting()](http://docs.google.com/javax/swing/table/TableColumn.html#enableResizedPosting())  *as of Java 2 platform v1.3* |
| [java.net.URLEncoder.encode(String)](http://docs.google.com/java/net/URLEncoder.html#encode(java.lang.String))  *The resulting string may vary depending on the platform's default encoding. Instead, use the encode(String,String) method to specify the encoding.* |
| [java.security.SignatureSpi.engineGetParameter(String)](http://docs.google.com/java/security/SignatureSpi.html#engineGetParameter(java.lang.String)) |
| [java.security.SignatureSpi.engineSetParameter(String, Object)](http://docs.google.com/java/security/SignatureSpi.html#engineSetParameter(java.lang.String,%20java.lang.Object))  *Replaced by* [*engineSetParameter*](http://docs.google.com/java/security/SignatureSpi.html#engineSetParameter(java.security.spec.AlgorithmParameterSpec))*.* |
| [java.awt.datatransfer.DataFlavor.equals(String)](http://docs.google.com/java/awt/datatransfer/DataFlavor.html#equals(java.lang.String))  *As inconsistent with hashCode() contract, use isMimeTypeEqual(String) instead.* |
| [org.omg.CORBA.ServerRequest.except(Any)](http://docs.google.com/org/omg/CORBA/ServerRequest.html#except(org.omg.CORBA.Any))  *use set\_exception()* |
| [java.rmi.server.RemoteCall.executeCall()](http://docs.google.com/java/rmi/server/RemoteCall.html#executeCall())  *no replacement* |
| [org.omg.CORBA.Any.extract\_Principal()](http://docs.google.com/org/omg/CORBA/Any.html#extract_Principal())  *Deprecated by CORBA 2.2.* |
| [javax.swing.SwingUtilities.findFocusOwner(Component)](http://docs.google.com/javax/swing/SwingUtilities.html#findFocusOwner(java.awt.Component))  *As of 1.4, replaced by KeyboardFocusManager.getFocusOwner().* |
| [org.omg.CORBA.ORB.get\_current()](http://docs.google.com/org/omg/CORBA/ORB.html#get_current())  *use resolve\_initial\_references.* |
| [java.security.Security.getAlgorithmProperty(String, String)](http://docs.google.com/java/security/Security.html#getAlgorithmProperty(java.lang.String,%20java.lang.String))  *This method used to return the value of a proprietary property in the master file of the "SUN" Cryptographic Service Provider in order to determine how to parse algorithm-specific parameters. Use the new provider-based and algorithm-independent AlgorithmParameters and KeyFactory engine classes (introduced in the J2SE version 1.2 platform) instead.* |
| [java.sql.CallableStatement.getBigDecimal(int, int)](http://docs.google.com/java/sql/CallableStatement.html#getBigDecimal(int,%20int))  *use getBigDecimal(int parameterIndex) or getBigDecimal(String parameterName)* |
| [java.sql.ResultSet.getBigDecimal(int, int)](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int,%20int)) |
| [java.sql.ResultSet.getBigDecimal(String, int)](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String,%20int)) |
| [java.awt.Polygon.getBoundingBox()](http://docs.google.com/java/awt/Polygon.html#getBoundingBox())  *As of JDK version 1.1, replaced by getBounds().* |
| [java.lang.String.getBytes(int, int, byte[], int)](http://docs.google.com/java/lang/String.html#getBytes(int,%20int,%20byte%5B%5D,%20int))  *This method does not properly convert characters into bytes. As of JDK 1.1, the preferred way to do this is via the* [*String.getBytes()*](http://docs.google.com/java/lang/String.html#getBytes()) *method, which uses the platform's default charset.* |
| [java.awt.Graphics.getClipRect()](http://docs.google.com/java/awt/Graphics.html#getClipRect())  *As of JDK version 1.1, replaced by getClipBounds().* |
| [javax.swing.JPopupMenu.getComponentAtIndex(int)](http://docs.google.com/javax/swing/JPopupMenu.html#getComponentAtIndex(int))  *replaced by* [*Container.getComponent(int)*](http://docs.google.com/java/awt/Container.html#getComponent(int)) |
| [javax.swing.JMenuBar.getComponentAtIndex(int)](http://docs.google.com/javax/swing/JMenuBar.html#getComponentAtIndex(int))  *replaced by getComponent(int i)* |
| [java.awt.CheckboxGroup.getCurrent()](http://docs.google.com/java/awt/CheckboxGroup.html#getCurrent())  *As of JDK version 1.1, replaced by getSelectedCheckbox().* |
| [java.awt.Frame.getCursorType()](http://docs.google.com/java/awt/Frame.html#getCursorType())  *As of JDK version 1.1, replaced by Component.getCursor().* |
| [java.sql.Time.getDate()](http://docs.google.com/java/sql/Time.html#getDate()) |
| [java.util.Date.getDate()](http://docs.google.com/java/util/Date.html#getDate())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.DAY\_OF\_MONTH).* |
| [java.sql.Time.getDay()](http://docs.google.com/java/sql/Time.html#getDay()) |
| [java.util.Date.getDay()](http://docs.google.com/java/util/Date.html#getDay())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.DAY\_OF\_WEEK).* |
| [java.net.URLConnection.getDefaultRequestProperty(String)](http://docs.google.com/java/net/URLConnection.html#getDefaultRequestProperty(java.lang.String))  *The instance specific getRequestProperty method should be used after an appropriate instance of URLConnection is obtained.* |
| [java.rmi.server.LogStream.getDefaultStream()](http://docs.google.com/java/rmi/server/LogStream.html#getDefaultStream())  *no replacement* |
| [javax.management.monitor.StringMonitorMBean.getDerivedGauge()](http://docs.google.com/javax/management/monitor/StringMonitorMBean.html#getDerivedGauge())  *As of JMX 1.2, replaced by* [*StringMonitorMBean.getDerivedGauge(ObjectName)*](http://docs.google.com/javax/management/monitor/StringMonitorMBean.html#getDerivedGauge(javax.management.ObjectName)) |
| [javax.management.monitor.StringMonitor.getDerivedGauge()](http://docs.google.com/javax/management/monitor/StringMonitor.html#getDerivedGauge())  *As of JMX 1.2, replaced by* [*StringMonitor.getDerivedGauge(ObjectName)*](http://docs.google.com/javax/management/monitor/StringMonitor.html#getDerivedGauge(javax.management.ObjectName)) |
| [javax.management.monitor.GaugeMonitorMBean.getDerivedGauge()](http://docs.google.com/javax/management/monitor/GaugeMonitorMBean.html#getDerivedGauge())  *As of JMX 1.2, replaced by* [*GaugeMonitorMBean.getDerivedGauge(ObjectName)*](http://docs.google.com/javax/management/monitor/GaugeMonitorMBean.html#getDerivedGauge(javax.management.ObjectName)) |
| [javax.management.monitor.GaugeMonitor.getDerivedGauge()](http://docs.google.com/javax/management/monitor/GaugeMonitor.html#getDerivedGauge())  *As of JMX 1.2, replaced by* [*GaugeMonitor.getDerivedGauge(ObjectName)*](http://docs.google.com/javax/management/monitor/GaugeMonitor.html#getDerivedGauge(javax.management.ObjectName)) |
| [javax.management.monitor.CounterMonitorMBean.getDerivedGauge()](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#getDerivedGauge())  *As of JMX 1.2, replaced by* [*CounterMonitorMBean.getDerivedGauge(ObjectName)*](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#getDerivedGauge(javax.management.ObjectName)) |
| [javax.management.monitor.CounterMonitor.getDerivedGauge()](http://docs.google.com/javax/management/monitor/CounterMonitor.html#getDerivedGauge())  *As of JMX 1.2, replaced by* [*CounterMonitor.getDerivedGauge(ObjectName)*](http://docs.google.com/javax/management/monitor/CounterMonitor.html#getDerivedGauge(javax.management.ObjectName)) |
| [javax.management.monitor.StringMonitorMBean.getDerivedGaugeTimeStamp()](http://docs.google.com/javax/management/monitor/StringMonitorMBean.html#getDerivedGaugeTimeStamp())  *As of JMX 1.2, replaced by* [*StringMonitorMBean.getDerivedGaugeTimeStamp(ObjectName)*](http://docs.google.com/javax/management/monitor/StringMonitorMBean.html#getDerivedGaugeTimeStamp(javax.management.ObjectName)) |
| [javax.management.monitor.StringMonitor.getDerivedGaugeTimeStamp()](http://docs.google.com/javax/management/monitor/StringMonitor.html#getDerivedGaugeTimeStamp())  *As of JMX 1.2, replaced by* [*StringMonitor.getDerivedGaugeTimeStamp(ObjectName)*](http://docs.google.com/javax/management/monitor/StringMonitor.html#getDerivedGaugeTimeStamp(javax.management.ObjectName)) |
| [javax.management.monitor.GaugeMonitorMBean.getDerivedGaugeTimeStamp()](http://docs.google.com/javax/management/monitor/GaugeMonitorMBean.html#getDerivedGaugeTimeStamp())  *As of JMX 1.2, replaced by* [*GaugeMonitorMBean.getDerivedGaugeTimeStamp(ObjectName)*](http://docs.google.com/javax/management/monitor/GaugeMonitorMBean.html#getDerivedGaugeTimeStamp(javax.management.ObjectName)) |
| [javax.management.monitor.GaugeMonitor.getDerivedGaugeTimeStamp()](http://docs.google.com/javax/management/monitor/GaugeMonitor.html#getDerivedGaugeTimeStamp())  *As of JMX 1.2, replaced by* [*GaugeMonitor.getDerivedGaugeTimeStamp(ObjectName)*](http://docs.google.com/javax/management/monitor/GaugeMonitor.html#getDerivedGaugeTimeStamp(javax.management.ObjectName)) |
| [javax.management.monitor.CounterMonitorMBean.getDerivedGaugeTimeStamp()](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#getDerivedGaugeTimeStamp())  *As of JMX 1.2, replaced by* [*CounterMonitorMBean.getDerivedGaugeTimeStamp(ObjectName)*](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#getDerivedGaugeTimeStamp(javax.management.ObjectName)) |
| [javax.management.monitor.CounterMonitor.getDerivedGaugeTimeStamp()](http://docs.google.com/javax/management/monitor/CounterMonitor.html#getDerivedGaugeTimeStamp())  *As of JMX 1.2, replaced by* [*CounterMonitor.getDerivedGaugeTimeStamp(ObjectName)*](http://docs.google.com/javax/management/monitor/CounterMonitor.html#getDerivedGaugeTimeStamp(javax.management.ObjectName)) |
| [javax.swing.plaf.basic.BasicSplitPaneUI.getDividerBorderSize()](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html#getDividerBorderSize())  *As of Java 2 platform v1.3, instead set the border on the divider.* |
| [javax.xml.bind.Validator.getEventHandler()](http://docs.google.com/javax/xml/bind/Validator.html#getEventHandler())  *since JAXB2.0* |
| [java.awt.Toolkit.getFontList()](http://docs.google.com/java/awt/Toolkit.html#getFontList())  *see* [*GraphicsEnvironment.getAvailableFontFamilyNames()*](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAvailableFontFamilyNames()) |
| [javax.swing.text.LabelView.getFontMetrics()](http://docs.google.com/javax/swing/text/LabelView.html#getFontMetrics())  *FontMetrics are not used for glyph rendering when running in the JDK.* |
| [java.awt.Toolkit.getFontMetrics(Font)](http://docs.google.com/java/awt/Toolkit.html#getFontMetrics(java.awt.Font))  *As of JDK version 1.2, replaced by the Font method getLineMetrics.* |
| [java.awt.Toolkit.getFontPeer(String, int)](http://docs.google.com/java/awt/Toolkit.html#getFontPeer(java.lang.String,%20int))  *see java.awt.GraphicsEnvironment#getAllFonts* |
| [java.sql.Date.getHours()](http://docs.google.com/java/sql/Date.html#getHours()) |
| [java.util.Date.getHours()](http://docs.google.com/java/util/Date.html#getHours())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.HOUR\_OF\_DAY).* |
| [java.lang.SecurityManager.getInCheck()](http://docs.google.com/java/lang/SecurityManager.html#getInCheck())  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [java.rmi.server.RemoteCall.getInputStream()](http://docs.google.com/java/rmi/server/RemoteCall.html#getInputStream())  *no replacement* |
| [javax.swing.KeyStroke.getKeyStroke(char, boolean)](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(char,%20boolean))  *use getKeyStroke(char)* |
| [javax.swing.AbstractButton.getLabel()](http://docs.google.com/javax/swing/AbstractButton.html#getLabel())  *- Replaced by getText* |
| [java.awt.Scrollbar.getLineIncrement()](http://docs.google.com/java/awt/Scrollbar.html#getLineIncrement())  *As of JDK version 1.1, replaced by getUnitIncrement().* |
| [java.lang.Runtime.getLocalizedInputStream(InputStream)](http://docs.google.com/java/lang/Runtime.html#getLocalizedInputStream(java.io.InputStream))  *As of JDK 1.1, the preferred way to translate a byte stream in the local encoding into a character stream in Unicode is via the InputStreamReader and BufferedReader classes.* |
| [java.lang.Runtime.getLocalizedOutputStream(OutputStream)](http://docs.google.com/java/lang/Runtime.html#getLocalizedOutputStream(java.io.OutputStream))  *As of JDK 1.1, the preferred way to translate a Unicode character stream into a byte stream in the local encoding is via the OutputStreamWriter, BufferedWriter, and PrintWriter classes.* |
| [java.sql.DriverManager.getLogStream()](http://docs.google.com/java/sql/DriverManager.html#getLogStream()) |
| [java.awt.FontMetrics.getMaxDecent()](http://docs.google.com/java/awt/FontMetrics.html#getMaxDecent())  *As of JDK version 1.1.1, replaced by getMaxDescent().* |
| [javax.swing.JInternalFrame.getMenuBar()](http://docs.google.com/javax/swing/JInternalFrame.html#getMenuBar())  *As of Swing version 1.0.3, replaced by getJMenuBar().* |
| [javax.swing.JRootPane.getMenuBar()](http://docs.google.com/javax/swing/JRootPane.html#getMenuBar())  *As of Swing version 1.0.3 replaced by getJMenuBar().* |
| [java.sql.Date.getMinutes()](http://docs.google.com/java/sql/Date.html#getMinutes()) |
| [java.util.Date.getMinutes()](http://docs.google.com/java/util/Date.html#getMinutes())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.MINUTE).* |
| [java.sql.Time.getMonth()](http://docs.google.com/java/sql/Time.html#getMonth()) |
| [java.util.Date.getMonth()](http://docs.google.com/java/util/Date.html#getMonth())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.MONTH).* |
| [javax.swing.JComponent.getNextFocusableComponent()](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent())  *As of 1.4, replaced by FocusTraversalPolicy.* |
| [javax.management.monitor.MonitorMBean.getObservedObject()](http://docs.google.com/javax/management/monitor/MonitorMBean.html#getObservedObject())  *As of JMX 1.2, replaced by* [*MonitorMBean.getObservedObjects()*](http://docs.google.com/javax/management/monitor/MonitorMBean.html#getObservedObjects()) |
| [javax.management.monitor.Monitor.getObservedObject()](http://docs.google.com/javax/management/monitor/Monitor.html#getObservedObject())  *As of JMX 1.2, replaced by* [*Monitor.getObservedObjects()*](http://docs.google.com/javax/management/monitor/Monitor.html#getObservedObjects()) |
| [java.rmi.server.Operation.getOperation()](http://docs.google.com/java/rmi/server/Operation.html#getOperation())  *no replacement* |
| [java.rmi.server.Skeleton.getOperations()](http://docs.google.com/java/rmi/server/Skeleton.html#getOperations())  *no replacement* |
| [java.awt.ComponentOrientation.getOrientation(ResourceBundle)](http://docs.google.com/java/awt/ComponentOrientation.html#getOrientation(java.util.ResourceBundle))  *As of J2SE 1.4, use* [*ComponentOrientation.getOrientation(java.util.Locale)*](http://docs.google.com/java/awt/ComponentOrientation.html#getOrientation(java.util.Locale))*.* |
| [java.rmi.server.LogStream.getOutputStream()](http://docs.google.com/java/rmi/server/LogStream.html#getOutputStream())  *no replacement* |
| [java.rmi.server.RemoteCall.getOutputStream()](http://docs.google.com/java/rmi/server/RemoteCall.html#getOutputStream())  *no replacement* |
| [java.awt.Scrollbar.getPageIncrement()](http://docs.google.com/java/awt/Scrollbar.html#getPageIncrement())  *As of JDK version 1.1, replaced by getBlockIncrement().* |
| [java.security.Signature.getParameter(String)](http://docs.google.com/java/security/Signature.html#getParameter(java.lang.String)) |
| [java.awt.MenuComponent.getPeer()](http://docs.google.com/java/awt/MenuComponent.html#getPeer())  *As of JDK version 1.1, programs should not directly manipulate peers.* |
| [java.awt.Component.getPeer()](http://docs.google.com/java/awt/Component.html#getPeer())  *As of JDK version 1.1, programs should not directly manipulate peers; replaced by boolean isDisplayable().* |
| [java.awt.Font.getPeer()](http://docs.google.com/java/awt/Font.html#getPeer())  *Font rendering is now platform independent.* |
| [javax.xml.bind.Validator.getProperty(String)](http://docs.google.com/javax/xml/bind/Validator.html#getProperty(java.lang.String))  *since JAXB2.0* |
| [java.rmi.server.RemoteCall.getResultStream(boolean)](http://docs.google.com/java/rmi/server/RemoteCall.html#getResultStream(boolean))  *no replacement* |
| [java.sql.Date.getSeconds()](http://docs.google.com/java/sql/Date.html#getSeconds()) |
| [java.util.Date.getSeconds()](http://docs.google.com/java/util/Date.html#getSeconds())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.SECOND).* |
| [java.rmi.server.LoaderHandler.getSecurityContext(ClassLoader)](http://docs.google.com/java/rmi/server/LoaderHandler.html#getSecurityContext(java.lang.ClassLoader))  *no replacement* |
| [java.rmi.server.RMIClassLoader.getSecurityContext(ClassLoader)](http://docs.google.com/java/rmi/server/RMIClassLoader.html#getSecurityContext(java.lang.ClassLoader))  *no replacement. As of the Java 2 platform v1.2, RMI no longer uses this method to obtain a class loader's security context.* |
| [javax.swing.JPasswordField.getText()](http://docs.google.com/javax/swing/JPasswordField.html#getText())  *As of Java 2 platform v1.2, replaced by getPassword.* |
| [javax.swing.JPasswordField.getText(int, int)](http://docs.google.com/javax/swing/JPasswordField.html#getText(int,%20int))  *As of Java 2 platform v1.2, replaced by getPassword.* |
| [javax.management.monitor.CounterMonitorMBean.getThreshold()](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#getThreshold())  *As of JMX 1.2, replaced by* [*CounterMonitorMBean.getThreshold(ObjectName)*](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#getThreshold(javax.management.ObjectName)) |
| [javax.management.monitor.CounterMonitor.getThreshold()](http://docs.google.com/javax/management/monitor/CounterMonitor.html#getThreshold())  *As of JMX 1.2, replaced by* [*CounterMonitor.getThreshold(ObjectName)*](http://docs.google.com/javax/management/monitor/CounterMonitor.html#getThreshold(javax.management.ObjectName)) |
| [java.util.Date.getTimezoneOffset()](http://docs.google.com/java/util/Date.html#getTimezoneOffset())  *As of JDK version 1.1, replaced by -(Calendar.get(Calendar.ZONE\_OFFSET) + Calendar.get(Calendar.DST\_OFFSET)) / (60 \* 1000).* |
| [java.net.MulticastSocket.getTTL()](http://docs.google.com/java/net/MulticastSocket.html#getTTL())  *use the getTimeToLive method instead, which returns an* ***int*** *instead of a* ***byte****.* |
| [java.net.DatagramSocketImpl.getTTL()](http://docs.google.com/java/net/DatagramSocketImpl.html#getTTL())  *use getTimeToLive instead.* |
| [java.sql.ResultSet.getUnicodeStream(int)](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(int))  *use getCharacterStream in place of getUnicodeStream* |
| [java.sql.ResultSet.getUnicodeStream(String)](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(java.lang.String))  *use getCharacterStream instead* |
| [javax.swing.ScrollPaneLayout.getViewportBorderBounds(JScrollPane)](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getViewportBorderBounds(javax.swing.JScrollPane))  *As of JDK version Swing1.1 replaced by JScrollPane.getViewportBorderBounds().* |
| [java.awt.Scrollbar.getVisible()](http://docs.google.com/java/awt/Scrollbar.html#getVisible())  *As of JDK version 1.1, replaced by getVisibleAmount().* |
| [java.sql.Time.getYear()](http://docs.google.com/java/sql/Time.html#getYear()) |
| [java.util.Date.getYear()](http://docs.google.com/java/util/Date.html#getYear())  *As of JDK version 1.1, replaced by Calendar.get(Calendar.YEAR) - 1900.* |
| [java.awt.Component.gotFocus(Event, Object)](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object))  *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).* |
| [java.awt.Component.handleEvent(Event)](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event))  *As of JDK version 1.1 replaced by processEvent(AWTEvent).* |
| [java.awt.Dialog.hide()](http://docs.google.com/java/awt/Dialog.html#hide())  *As of JDK version 1.5, replaced by* [*setVisible(boolean)*](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))*.* |
| [java.awt.Window.hide()](http://docs.google.com/java/awt/Window.html#hide())  *As of JDK version 1.5, replaced by* [*Window.setVisible(boolean)*](http://docs.google.com/java/awt/Window.html#setVisible(boolean))*.* |
| [java.awt.Component.hide()](http://docs.google.com/java/awt/Component.html#hide())  *As of JDK version 1.1, replaced by setVisible(boolean).* |
| [java.lang.SecurityManager.inClass(String)](http://docs.google.com/java/lang/SecurityManager.html#inClass(java.lang.String))  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [java.lang.SecurityManager.inClassLoader()](http://docs.google.com/java/lang/SecurityManager.html#inClassLoader())  *This type of security checking is not recommended. It is recommended that the checkPermission call be used instead.* |
| [org.omg.CORBA.Any.insert\_Principal(Principal)](http://docs.google.com/org/omg/CORBA/Any.html#insert_Principal(org.omg.CORBA.Principal))  *Deprecated by CORBA 2.2.* |
| [javax.swing.text.html.HTMLEditorKit.InsertHTMLTextAction.insertAtBoundry(JEditorPane, HTMLDocument, int, Element, String, HTML.Tag, HTML.Tag)](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.InsertHTMLTextAction.html#insertAtBoundry(javax.swing.JEditorPane,%20javax.swing.text.html.HTMLDocument,%20int,%20javax.swing.text.Element,%20java.lang.String,%20javax.swing.text.html.HTML.Tag,%20javax.swing.text.html.HTML.Tag))  *As of Java 2 platform v1.3, use insertAtBoundary* |
| [java.awt.TextArea.insertText(String, int)](http://docs.google.com/java/awt/TextArea.html#insertText(java.lang.String,%20int))  *As of JDK version 1.1, replaced by insert(String, int).* |
| [java.awt.Container.insets()](http://docs.google.com/java/awt/Container.html#insets())  *As of JDK version 1.1, replaced by getInsets().* |
| [java.awt.Rectangle.inside(int, int)](http://docs.google.com/java/awt/Rectangle.html#inside(int,%20int))  *As of JDK version 1.1, replaced by contains(int, int).* |
| [java.awt.Component.inside(int, int)](http://docs.google.com/java/awt/Component.html#inside(int,%20int))  *As of JDK version 1.1, replaced by contains(int, int).* |
| [java.awt.Polygon.inside(int, int)](http://docs.google.com/java/awt/Polygon.html#inside(int,%20int))  *As of JDK version 1.1, replaced by contains(int, int).* |
| [java.rmi.server.RemoteRef.invoke(RemoteCall)](http://docs.google.com/java/rmi/server/RemoteRef.html#invoke(java.rmi.server.RemoteCall))  *1.2 style stubs no longer use this method. Instead of using a sequence of method calls to the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.* |
| [org.omg.CORBA.DynamicImplementation.invoke(ServerRequest)](http://docs.google.com/org/omg/CORBA/DynamicImplementation.html#invoke(org.omg.CORBA.ServerRequest))  *Deprecated by Portable Object Adapter* |
| [javax.swing.JViewport.isBackingStoreEnabled()](http://docs.google.com/javax/swing/JViewport.html#isBackingStoreEnabled())  *As of Java 2 platform v1.3, replaced by getScrollMode().* |
| [javax.swing.FocusManager.isFocusManagerEnabled()](http://docs.google.com/javax/swing/FocusManager.html#isFocusManagerEnabled())  *As of 1.4, replaced by KeyboardFocusManager.getDefaultFocusTraversalPolicy()* |
| [java.awt.Component.isFocusTraversable()](http://docs.google.com/java/awt/Component.html#isFocusTraversable())  *As of 1.4, replaced by isFocusable().* |
| [java.lang.Character.isJavaLetter(char)](http://docs.google.com/java/lang/Character.html#isJavaLetter(char))  *Replaced by isJavaIdentifierStart(char).* |
| [java.lang.Character.isJavaLetterOrDigit(char)](http://docs.google.com/java/lang/Character.html#isJavaLetterOrDigit(char))  *Replaced by isJavaIdentifierPart(char).* |
| [javax.swing.JComponent.isManagingFocus()](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus())  *As of 1.4, replaced by Component.setFocusTraversalKeys(int, Set) and Container.setFocusCycleRoot(boolean).* |
| [java.awt.List.isSelected(int)](http://docs.google.com/java/awt/List.html#isSelected(int))  *As of JDK version 1.1, replaced by isIndexSelected(int).* |
| [java.lang.Character.isSpace(char)](http://docs.google.com/java/lang/Character.html#isSpace(char))  *Replaced by isWhitespace(char).* |
| [java.rmi.dgc.VMID.isUnique()](http://docs.google.com/java/rmi/dgc/VMID.html#isUnique()) |
| [javax.xml.bind.Unmarshaller.isValidating()](http://docs.google.com/javax/xml/bind/Unmarshaller.html#isValidating())  *since JAXB2.0, please see* [*Unmarshaller.getSchema()*](http://docs.google.com/javax/xml/bind/Unmarshaller.html#getSchema()) |
| [java.awt.Component.keyDown(Event, int)](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int))  *As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).* |
| [java.awt.Component.keyUp(Event, int)](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int))  *As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).* |
| [java.awt.Container.layout()](http://docs.google.com/java/awt/Container.html#layout())  *As of JDK version 1.1, replaced by doLayout().* |
| [java.awt.ScrollPane.layout()](http://docs.google.com/java/awt/ScrollPane.html#layout())  *As of JDK version 1.1, replaced by doLayout().* |
| [java.awt.Component.layout()](http://docs.google.com/java/awt/Component.html#layout())  *As of JDK version 1.1, replaced by doLayout().* |
| [java.rmi.server.LoaderHandler.loadClass(String)](http://docs.google.com/java/rmi/server/LoaderHandler.html#loadClass(java.lang.String))  *no replacement* |
| [java.rmi.server.RMIClassLoader.loadClass(String)](http://docs.google.com/java/rmi/server/RMIClassLoader.html#loadClass(java.lang.String))  *replaced by loadClass(String,String) method* |
| [java.rmi.server.LoaderHandler.loadClass(URL, String)](http://docs.google.com/java/rmi/server/LoaderHandler.html#loadClass(java.net.URL,%20java.lang.String))  *no replacement* |
| [java.awt.Container.locate(int, int)](http://docs.google.com/java/awt/Container.html#locate(int,%20int))  *As of JDK version 1.1, replaced by getComponentAt(int, int).* |
| [java.awt.Component.locate(int, int)](http://docs.google.com/java/awt/Component.html#locate(int,%20int))  *As of JDK version 1.1, replaced by getComponentAt(int, int).* |
| [java.awt.Component.location()](http://docs.google.com/java/awt/Component.html#location())  *As of JDK version 1.1, replaced by getLocation().* |
| [java.rmi.server.LogStream.log(String)](http://docs.google.com/java/rmi/server/LogStream.html#log(java.lang.String))  *no replacement* |
| [java.awt.Component.lostFocus(Event, Object)](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object))  *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).* |
| [java.awt.TextArea.minimumSize()](http://docs.google.com/java/awt/TextArea.html#minimumSize())  *As of JDK version 1.1, replaced by getMinimumSize().* |
| [java.awt.Container.minimumSize()](http://docs.google.com/java/awt/Container.html#minimumSize())  *As of JDK version 1.1, replaced by getMinimumSize().* |
| [java.awt.List.minimumSize()](http://docs.google.com/java/awt/List.html#minimumSize())  *As of JDK version 1.1, replaced by getMinimumSize().* |
| [java.awt.TextField.minimumSize()](http://docs.google.com/java/awt/TextField.html#minimumSize())  *As of JDK version 1.1, replaced by getMinimumSize().* |
| [java.awt.Component.minimumSize()](http://docs.google.com/java/awt/Component.html#minimumSize())  *As of JDK version 1.1, replaced by getMinimumSize().* |
| [java.awt.List.minimumSize(int)](http://docs.google.com/java/awt/List.html#minimumSize(int))  *As of JDK version 1.1, replaced by getMinimumSize(int).* |
| [java.awt.TextField.minimumSize(int)](http://docs.google.com/java/awt/TextField.html#minimumSize(int))  *As of JDK version 1.1, replaced by getMinimumSize(int).* |
| [java.awt.TextArea.minimumSize(int, int)](http://docs.google.com/java/awt/TextArea.html#minimumSize(int,%20int))  *As of JDK version 1.1, replaced by getMinimumSize(int, int).* |
| [javax.swing.text.View.modelToView(int, Shape)](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape)) |
| [java.awt.Component.mouseDown(Event, int, int)](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int))  *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| [java.awt.Component.mouseDrag(Event, int, int)](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int))  *As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).* |
| [java.awt.Component.mouseEnter(Event, int, int)](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int))  *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| [java.awt.Component.mouseExit(Event, int, int)](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int))  *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| [java.awt.Component.mouseMove(Event, int, int)](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int))  *As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).* |
| [java.awt.Component.mouseUp(Event, int, int)](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int))  *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| [java.awt.Rectangle.move(int, int)](http://docs.google.com/java/awt/Rectangle.html#move(int,%20int))  *As of JDK version 1.1, replaced by setLocation(int, int).* |
| [java.awt.Component.move(int, int)](http://docs.google.com/java/awt/Component.html#move(int,%20int))  *As of JDK version 1.1, replaced by setLocation(int, int).* |
| [org.omg.CORBA.Principal.name()](http://docs.google.com/org/omg/CORBA/Principal.html#name())  *Deprecated by CORBA 2.2.* |
| [org.omg.CORBA.Principal.name(byte[])](http://docs.google.com/org/omg/CORBA/Principal.html#name(byte%5B%5D))  *Deprecated by CORBA 2.2.* |
| [java.rmi.server.RemoteRef.newCall(RemoteObject, Operation[], int, long)](http://docs.google.com/java/rmi/server/RemoteRef.html#newCall(java.rmi.server.RemoteObject,%20java.rmi.server.Operation%5B%5D,%20int,%20long))  *1.2 style stubs no longer use this method. Instead of using a sequence of method calls on the stub's the remote reference (newCall, invoke, and done), a stub uses a single method, invoke(Remote, Method, Object[], int), on the remote reference to carry out parameter marshalling, remote method executing and unmarshalling of the return value.* |
| [java.awt.Component.nextFocus()](http://docs.google.com/java/awt/Component.html#nextFocus())  *As of JDK version 1.1, replaced by transferFocus().* |
| [java.awt.datatransfer.DataFlavor.normalizeMimeType(String)](http://docs.google.com/java/awt/datatransfer/DataFlavor.html#normalizeMimeType(java.lang.String)) |
| [javax.activation.ActivationDataFlavor.normalizeMimeType(String)](http://docs.google.com/javax/activation/ActivationDataFlavor.html#normalizeMimeType(java.lang.String)) |
| [java.awt.datatransfer.DataFlavor.normalizeMimeTypeParameter(String, String)](http://docs.google.com/java/awt/datatransfer/DataFlavor.html#normalizeMimeTypeParameter(java.lang.String,%20java.lang.String)) |
| [javax.activation.ActivationDataFlavor.normalizeMimeTypeParameter(String, String)](http://docs.google.com/javax/activation/ActivationDataFlavor.html#normalizeMimeTypeParameter(java.lang.String,%20java.lang.String)) |
| [org.omg.CORBA.ServerRequest.op\_name()](http://docs.google.com/org/omg/CORBA/ServerRequest.html#op_name())  *use operation()* |
| [org.omg.CORBA.ServerRequest.params(NVList)](http://docs.google.com/org/omg/CORBA/ServerRequest.html#params(org.omg.CORBA.NVList))  *use the method arguments* |
| [java.util.Date.parse(String)](http://docs.google.com/java/util/Date.html#parse(java.lang.String))  *As of JDK version 1.1, replaced by DateFormat.parse(String s).* |
| [java.rmi.server.LogStream.parseLevel(String)](http://docs.google.com/java/rmi/server/LogStream.html#parseLevel(java.lang.String))  *no replacement* |
| [java.awt.MenuComponent.postEvent(Event)](http://docs.google.com/java/awt/MenuComponent.html#postEvent(java.awt.Event))  *As of JDK version 1.1, replaced by* [*dispatchEvent*](http://docs.google.com/java/awt/MenuComponent.html#dispatchEvent(java.awt.AWTEvent))*.* |
| [java.awt.Window.postEvent(Event)](http://docs.google.com/java/awt/Window.html#postEvent(java.awt.Event))  *As of JDK version 1.1 replaced by dispatchEvent(AWTEvent).* |
| [java.awt.Component.postEvent(Event)](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event))  *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent).* |
| [java.awt.MenuContainer.postEvent(Event)](http://docs.google.com/java/awt/MenuContainer.html#postEvent(java.awt.Event))  *As of JDK version 1.1 replaced by dispatchEvent(AWTEvent).* |
| [java.awt.image.renderable.RenderContext.preConcetenateTransform(AffineTransform)](http://docs.google.com/java/awt/image/renderable/RenderContext.html#preConcetenateTransform(java.awt.geom.AffineTransform))  *replaced by preConcatenateTransform(AffineTransform).* |
| [java.awt.TextArea.preferredSize()](http://docs.google.com/java/awt/TextArea.html#preferredSize())  *As of JDK version 1.1, replaced by getPreferredSize().* |
| [java.awt.Container.preferredSize()](http://docs.google.com/java/awt/Container.html#preferredSize())  *As of JDK version 1.1, replaced by getPreferredSize().* |
| [java.awt.List.preferredSize()](http://docs.google.com/java/awt/List.html#preferredSize())  *As of JDK version 1.1, replaced by getPreferredSize().* |
| [java.awt.TextField.preferredSize()](http://docs.google.com/java/awt/TextField.html#preferredSize())  *As of JDK version 1.1, replaced by getPreferredSize().* |
| [java.awt.Component.preferredSize()](http://docs.google.com/java/awt/Component.html#preferredSize())  *As of JDK version 1.1, replaced by getPreferredSize().* |
| [java.awt.List.preferredSize(int)](http://docs.google.com/java/awt/List.html#preferredSize(int))  *As of JDK version 1.1, replaced by getPreferredSize(int).* |
| [java.awt.TextField.preferredSize(int)](http://docs.google.com/java/awt/TextField.html#preferredSize(int))  *As of JDK version 1.1, replaced by getPreferredSize(int).* |
| [java.awt.TextArea.preferredSize(int, int)](http://docs.google.com/java/awt/TextArea.html#preferredSize(int,%20int))  *As of JDK version 1.1, replaced by getPreferredSize(int, int).* |
| [org.omg.CORBA.portable.InputStream.read\_Principal()](http://docs.google.com/org/omg/CORBA/portable/InputStream.html#read_Principal())  *Deprecated by CORBA 2.2.* |
| [java.io.DataInputStream.readLine()](http://docs.google.com/java/io/DataInputStream.html#readLine())  *This method does not properly convert bytes to characters. As of JDK 1.1, the preferred way to read lines of text is via the BufferedReader.readLine() method. Programs that use the DataInputStream class to read lines can be converted to use the BufferedReader class by replacing code of the form:*  *DataInputStream d = new DataInputStream(in);*  *with:*  *BufferedReader d  = new BufferedReader(new InputStreamReader(in));* |
| [java.io.ObjectInputStream.readLine()](http://docs.google.com/java/io/ObjectInputStream.html#readLine())  *This method does not properly convert bytes to characters. see DataInputStream for the details and alternatives.* |
| [java.rmi.registry.RegistryHandler.registryImpl(int)](http://docs.google.com/java/rmi/registry/RegistryHandler.html#registryImpl(int))  *no replacement. As of the Java 2 platform v1.2, RMI no longer uses the RegistryHandler to obtain the registry's implementation.* |
| [java.rmi.registry.RegistryHandler.registryStub(String, int)](http://docs.google.com/java/rmi/registry/RegistryHandler.html#registryStub(java.lang.String,%20int))  *no replacement. As of the Java 2 platform v1.2, RMI no longer uses the RegistryHandler to obtain the registry's stub.* |
| [java.rmi.server.RemoteCall.releaseInputStream()](http://docs.google.com/java/rmi/server/RemoteCall.html#releaseInputStream())  *no replacement* |
| [java.rmi.server.RemoteCall.releaseOutputStream()](http://docs.google.com/java/rmi/server/RemoteCall.html#releaseOutputStream())  *no replacement* |
| [javax.swing.plaf.metal.MetalComboBoxUI.removeListeners()](http://docs.google.com/javax/swing/plaf/metal/MetalComboBoxUI.html#removeListeners())  *As of Java 2 platform v1.4.* |
| [java.awt.TextArea.replaceText(String, int, int)](http://docs.google.com/java/awt/TextArea.html#replaceText(java.lang.String,%20int,%20int))  *As of JDK version 1.1, replaced by replaceRange(String, int, int).* |
| [javax.swing.JComponent.requestDefaultFocus()](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus())  *As of 1.4, replaced by FocusTraversalPolicy.getDefaultComponent(Container).requestFocus()* |
| [java.awt.Window.reshape(int, int, int, int)](http://docs.google.com/java/awt/Window.html#reshape(int,%20int,%20int,%20int))  *As of JDK version 1.1, replaced by setBounds(int, int, int, int).* |
| [java.awt.Rectangle.reshape(int, int, int, int)](http://docs.google.com/java/awt/Rectangle.html#reshape(int,%20int,%20int,%20int))  *As of JDK version 1.1, replaced by setBounds(int, int, int, int).* |
| [java.awt.Component.reshape(int, int, int, int)](http://docs.google.com/java/awt/Component.html#reshape(int,%20int,%20int,%20int))  *As of JDK version 1.1, replaced by setBounds(int, int, int, int).* |
| [javax.swing.JComponent.reshape(int, int, int, int)](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int))  *As of JDK 5, replaced by Component.setBounds(int, int, int, int).*  *Moves and resizes this component.* |
| [java.awt.Component.resize(Dimension)](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension))  *As of JDK version 1.1, replaced by setSize(Dimension).* |
| [java.awt.Rectangle.resize(int, int)](http://docs.google.com/java/awt/Rectangle.html#resize(int,%20int))  *As of JDK version 1.1, replaced by setSize(int, int).* |
| [java.awt.Component.resize(int, int)](http://docs.google.com/java/awt/Component.html#resize(int,%20int))  *As of JDK version 1.1, replaced by setSize(int, int).* |
| [org.omg.CORBA.ServerRequest.result(Any)](http://docs.google.com/org/omg/CORBA/ServerRequest.html#result(org.omg.CORBA.Any))  *use the method set\_result* |
| [java.lang.Thread.resume()](http://docs.google.com/java/lang/Thread.html#resume())  *This method exists solely for use with* [*Thread.suspend()*](http://docs.google.com/java/lang/Thread.html#suspend())*, which has been deprecated because it is deadlock-prone. For more information, see* [*Why are Thread.stop, Thread.suspend and Thread.resume Deprecated?*](http://docs.google.com/technotes/guides/concurrency/threadPrimitiveDeprecation.html)*.* |
| [java.lang.ThreadGroup.resume()](http://docs.google.com/java/lang/ThreadGroup.html#resume())  *This method is used solely in conjunction with Thread.suspend and ThreadGroup.suspend, both of which have been deprecated, as they are inherently deadlock-prone. See* [*Thread.suspend()*](http://docs.google.com/java/lang/Thread.html#suspend()) *for details.* |
| [java.lang.System.runFinalizersOnExit(boolean)](http://docs.google.com/java/lang/System.html#runFinalizersOnExit(boolean))  *This method is inherently unsafe. It may result in finalizers being called on live objects while other threads are concurrently manipulating those objects, resulting in erratic behavior or deadlock.* |
| [java.lang.Runtime.runFinalizersOnExit(boolean)](http://docs.google.com/java/lang/Runtime.html#runFinalizersOnExit(boolean))  *This method is inherently unsafe. It may result in finalizers being called on live objects while other threads are concurrently manipulating those objects, resulting in erratic behavior or deadlock.* |
| [java.util.Properties.save(OutputStream, String)](http://docs.google.com/java/util/Properties.html#save(java.io.OutputStream,%20java.lang.String))  *This method does not throw an IOException if an I/O error occurs while saving the property list. The preferred way to save a properties list is via the store(OutputStream out, String comments) method or the storeToXML(OutputStream os, String comment) method.* |
| [java.net.MulticastSocket.send(DatagramPacket, byte)](http://docs.google.com/java/net/MulticastSocket.html#send(java.net.DatagramPacket,%20byte))  *Use the following code or its equivalent instead: ...... int ttl = mcastSocket.getTimeToLive(); mcastSocket.setTimeToLive(newttl); mcastSocket.send(p); mcastSocket.setTimeToLive(ttl); ......* |
| [javax.swing.JViewport.setBackingStoreEnabled(boolean)](http://docs.google.com/javax/swing/JViewport.html#setBackingStoreEnabled(boolean))  *As of Java 2 platform v1.3, replaced by setScrollMode().* |
| [java.awt.CheckboxGroup.setCurrent(Checkbox)](http://docs.google.com/java/awt/CheckboxGroup.html#setCurrent(java.awt.Checkbox))  *As of JDK version 1.1, replaced by setSelectedCheckbox(Checkbox).* |
| [java.awt.Frame.setCursor(int)](http://docs.google.com/java/awt/Frame.html#setCursor(int))  *As of JDK version 1.1, replaced by Component.setCursor(Cursor).* |
| [java.sql.Time.setDate(int)](http://docs.google.com/java/sql/Time.html#setDate(int)) |
| [java.util.Date.setDate(int)](http://docs.google.com/java/util/Date.html#setDate(int))  *As of JDK version 1.1, replaced by Calendar.set(Calendar.DAY\_OF\_MONTH, int date).* |
| [java.net.URLConnection.setDefaultRequestProperty(String, String)](http://docs.google.com/java/net/URLConnection.html#setDefaultRequestProperty(java.lang.String,%20java.lang.String))  *The instance specific setRequestProperty method should be used after an appropriate instance of URLConnection is obtained. Invoking this method will have no effect.* |
| [java.rmi.server.LogStream.setDefaultStream(PrintStream)](http://docs.google.com/java/rmi/server/LogStream.html#setDefaultStream(java.io.PrintStream))  *no replacement* |
| [java.awt.TextField.setEchoCharacter(char)](http://docs.google.com/java/awt/TextField.html#setEchoCharacter(char))  *As of JDK version 1.1, replaced by setEchoChar(char).* |
| [javax.xml.bind.Validator.setEventHandler(ValidationEventHandler)](http://docs.google.com/javax/xml/bind/Validator.html#setEventHandler(javax.xml.bind.ValidationEventHandler))  *since JAXB2.0* |
| [java.sql.Date.setHours(int)](http://docs.google.com/java/sql/Date.html#setHours(int)) |
| [java.util.Date.setHours(int)](http://docs.google.com/java/util/Date.html#setHours(int))  *As of JDK version 1.1, replaced by Calendar.set(Calendar.HOUR\_OF\_DAY, int hours).* |
| [javax.swing.AbstractButton.setLabel(String)](http://docs.google.com/javax/swing/AbstractButton.html#setLabel(java.lang.String))  *- Replaced by setText(text)* |
| [java.awt.Scrollbar.setLineIncrement(int)](http://docs.google.com/java/awt/Scrollbar.html#setLineIncrement(int))  *As of JDK version 1.1, replaced by setUnitIncrement(int).* |
| [java.sql.DriverManager.setLogStream(PrintStream)](http://docs.google.com/java/sql/DriverManager.html#setLogStream(java.io.PrintStream)) |
| [javax.management.ValueExp.setMBeanServer(MBeanServer)](http://docs.google.com/javax/management/ValueExp.html#setMBeanServer(javax.management.MBeanServer))  *This method is not needed because a ValueExp can access the MBean server in which it is being evaluated by using* [*QueryEval.getMBeanServer()*](http://docs.google.com/javax/management/QueryEval.html#getMBeanServer())*.* |
| [javax.swing.JInternalFrame.setMenuBar(JMenuBar)](http://docs.google.com/javax/swing/JInternalFrame.html#setMenuBar(javax.swing.JMenuBar))  *As of Swing version 1.0.3 replaced by setJMenuBar(JMenuBar m).* |
| [javax.swing.JRootPane.setMenuBar(JMenuBar)](http://docs.google.com/javax/swing/JRootPane.html#setMenuBar(javax.swing.JMenuBar))  *As of Swing version 1.0.3 replaced by setJMenuBar(JMenuBar menu).* |
| [java.sql.Date.setMinutes(int)](http://docs.google.com/java/sql/Date.html#setMinutes(int)) |
| [java.util.Date.setMinutes(int)](http://docs.google.com/java/util/Date.html#setMinutes(int))  *As of JDK version 1.1, replaced by Calendar.set(Calendar.MINUTE, int minutes).* |
| [java.awt.event.KeyEvent.setModifiers(int)](http://docs.google.com/java/awt/event/KeyEvent.html#setModifiers(int))  *as of JDK1.1.4* |
| [java.sql.Time.setMonth(int)](http://docs.google.com/java/sql/Time.html#setMonth(int)) |
| [java.util.Date.setMonth(int)](http://docs.google.com/java/util/Date.html#setMonth(int))  *As of JDK version 1.1, replaced by Calendar.set(Calendar.MONTH, int month).* |
| [java.awt.List.setMultipleSelections(boolean)](http://docs.google.com/java/awt/List.html#setMultipleSelections(boolean))  *As of JDK version 1.1, replaced by setMultipleMode(boolean).* |
| [javax.swing.JComponent.setNextFocusableComponent(Component)](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component))  *As of 1.4, replaced by FocusTraversalPolicy* |
| [javax.management.monitor.MonitorMBean.setObservedObject(ObjectName)](http://docs.google.com/javax/management/monitor/MonitorMBean.html#setObservedObject(javax.management.ObjectName))  *As of JMX 1.2, replaced by* [*MonitorMBean.addObservedObject(javax.management.ObjectName)*](http://docs.google.com/javax/management/monitor/MonitorMBean.html#addObservedObject(javax.management.ObjectName)) |
| [javax.management.monitor.Monitor.setObservedObject(ObjectName)](http://docs.google.com/javax/management/monitor/Monitor.html#setObservedObject(javax.management.ObjectName))  *As of JMX 1.2, replaced by* [*Monitor.addObservedObject(javax.management.ObjectName)*](http://docs.google.com/javax/management/monitor/Monitor.html#addObservedObject(javax.management.ObjectName)) |
| [java.rmi.server.LogStream.setOutputStream(OutputStream)](http://docs.google.com/java/rmi/server/LogStream.html#setOutputStream(java.io.OutputStream))  *no replacement* |
| [java.awt.Scrollbar.setPageIncrement(int)](http://docs.google.com/java/awt/Scrollbar.html#setPageIncrement(int))  *As of JDK version 1.1, replaced by setBlockIncrement().* |
| [java.security.Signature.setParameter(String, Object)](http://docs.google.com/java/security/Signature.html#setParameter(java.lang.String,%20java.lang.Object))  *Use* [*setParameter*](http://docs.google.com/java/security/Signature.html#setParameter(java.security.spec.AlgorithmParameterSpec))*.* |
| [javax.xml.bind.Validator.setProperty(String, Object)](http://docs.google.com/javax/xml/bind/Validator.html#setProperty(java.lang.String,%20java.lang.Object))  *since JAXB2.0* |
| [java.rmi.server.RemoteStub.setRef(RemoteStub, RemoteRef)](http://docs.google.com/java/rmi/server/RemoteStub.html#setRef(java.rmi.server.RemoteStub,%20java.rmi.server.RemoteRef))  *no replacement. The setRef method is not needed since RemoteStubs can be created with the RemoteStub(RemoteRef) constructor.* |
| [java.sql.Date.setSeconds(int)](http://docs.google.com/java/sql/Date.html#setSeconds(int)) |
| [java.util.Date.setSeconds(int)](http://docs.google.com/java/util/Date.html#setSeconds(int))  *As of JDK version 1.1, replaced by Calendar.set(Calendar.SECOND, int seconds).* |
| [javax.management.monitor.CounterMonitorMBean.setThreshold(Number)](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#setThreshold(java.lang.Number))  *As of JMX 1.2, replaced by* [*CounterMonitorMBean.setInitThreshold(java.lang.Number)*](http://docs.google.com/javax/management/monitor/CounterMonitorMBean.html#setInitThreshold(java.lang.Number)) |
| [javax.management.monitor.CounterMonitor.setThreshold(Number)](http://docs.google.com/javax/management/monitor/CounterMonitor.html#setThreshold(java.lang.Number))  *As of JMX 1.2, replaced by* [*CounterMonitor.setInitThreshold(java.lang.Number)*](http://docs.google.com/javax/management/monitor/CounterMonitor.html#setInitThreshold(java.lang.Number)) |
| [java.net.MulticastSocket.setTTL(byte)](http://docs.google.com/java/net/MulticastSocket.html#setTTL(byte))  *use the setTimeToLive method instead, which uses* ***int*** *instead of* ***byte*** *as the type for ttl.* |
| [java.net.DatagramSocketImpl.setTTL(byte)](http://docs.google.com/java/net/DatagramSocketImpl.html#setTTL(byte))  *use setTimeToLive instead.* |
| [java.sql.PreparedStatement.setUnicodeStream(int, InputStream, int)](http://docs.google.com/java/sql/PreparedStatement.html#setUnicodeStream(int,%20java.io.InputStream,%20int)) |
| [javax.sql.rowset.BaseRowSet.setUnicodeStream(int, InputStream, int)](http://docs.google.com/javax/sql/rowset/BaseRowSet.html#setUnicodeStream(int,%20java.io.InputStream,%20int))  *getCharacterStream should be used in its place* |
| [java.net.URLStreamHandler.setURL(URL, String, String, int, String, String)](http://docs.google.com/java/net/URLStreamHandler.html#setURL(java.net.URL,%20java.lang.String,%20java.lang.String,%20int,%20java.lang.String,%20java.lang.String))  *Use setURL(URL, String, String, int, String, String, String, String);* |
| [javax.xml.bind.Unmarshaller.setValidating(boolean)](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setValidating(boolean))  *since JAXB2.0, please see* [*Unmarshaller.setSchema(javax.xml.validation.Schema)*](http://docs.google.com/javax/xml/bind/Unmarshaller.html#setSchema(javax.xml.validation.Schema)) |
| [java.sql.Time.setYear(int)](http://docs.google.com/java/sql/Time.html#setYear(int)) |
| [java.util.Date.setYear(int)](http://docs.google.com/java/util/Date.html#setYear(int))  *As of JDK version 1.1, replaced by Calendar.set(Calendar.YEAR, year + 1900).* |
| [java.awt.Dialog.show()](http://docs.google.com/java/awt/Dialog.html#show())  *As of JDK version 1.5, replaced by* [*setVisible(boolean)*](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))*.* |
| [java.awt.Window.show()](http://docs.google.com/java/awt/Window.html#show())  *As of JDK version 1.5, replaced by* [*Window.setVisible(boolean)*](http://docs.google.com/java/awt/Window.html#setVisible(boolean))*.* |
| [java.awt.Component.show()](http://docs.google.com/java/awt/Component.html#show())  *As of JDK version 1.1, replaced by setVisible(boolean).* |
| [java.awt.Component.show(boolean)](http://docs.google.com/java/awt/Component.html#show(boolean))  *As of JDK version 1.1, replaced by setVisible(boolean).* |
| [java.awt.Component.size()](http://docs.google.com/java/awt/Component.html#size())  *As of JDK version 1.1, replaced by getSize().* |
| [javax.swing.JTable.sizeColumnsToFit(boolean)](http://docs.google.com/javax/swing/JTable.html#sizeColumnsToFit(boolean))  *As of Swing version 1.0.3, replaced by doLayout().* |
| [java.lang.Thread.stop()](http://docs.google.com/java/lang/Thread.html#stop())  *This method is inherently unsafe. Stopping a thread with Thread.stop causes it to unlock all of the monitors that it has locked (as a natural consequence of the unchecked ThreadDeath exception propagating up the stack). If any of the objects previously protected by these monitors were in an inconsistent state, the damaged objects become visible to other threads, potentially resulting in arbitrary behavior. Many uses of stop should be replaced by code that simply modifies some variable to indicate that the target thread should stop running. The target thread should check this variable regularly, and return from its run method in an orderly fashion if the variable indicates that it is to stop running. If the target thread waits for long periods (on a condition variable, for example), the interrupt method should be used to interrupt the wait. For more information, see* [*Why are Thread.stop, Thread.suspend and Thread.resume Deprecated?*](http://docs.google.com/technotes/guides/concurrency/threadPrimitiveDeprecation.html)*.* |
| [java.lang.ThreadGroup.stop()](http://docs.google.com/java/lang/ThreadGroup.html#stop())  *This method is inherently unsafe. See* [*Thread.stop()*](http://docs.google.com/java/lang/Thread.html#stop()) *for details.* |
| [java.lang.Thread.stop(Throwable)](http://docs.google.com/java/lang/Thread.html#stop(java.lang.Throwable))  *This method is inherently unsafe. See* [*Thread.stop()*](http://docs.google.com/java/lang/Thread.html#stop()) *for details. An additional danger of this method is that it may be used to generate exceptions that the target thread is unprepared to handle (including checked exceptions that the thread could not possibly throw, were it not for this method). For more information, see* [*Why are Thread.stop, Thread.suspend and Thread.resume Deprecated?*](http://docs.google.com/technotes/guides/concurrency/threadPrimitiveDeprecation.html)*.* |
| [java.lang.Thread.suspend()](http://docs.google.com/java/lang/Thread.html#suspend())  *This method has been deprecated, as it is inherently deadlock-prone. If the target thread holds a lock on the monitor protecting a critical system resource when it is suspended, no thread can access this resource until the target thread is resumed. If the thread that would resume the target thread attempts to lock this monitor prior to calling resume, deadlock results. Such deadlocks typically manifest themselves as "frozen" processes. For more information, see* [*Why are Thread.stop, Thread.suspend and Thread.resume Deprecated?*](http://docs.google.com/technotes/guides/concurrency/threadPrimitiveDeprecation.html)*.* |
| [java.lang.ThreadGroup.suspend()](http://docs.google.com/java/lang/ThreadGroup.html#suspend())  *This method is inherently deadlock-prone. See* [*Thread.suspend()*](http://docs.google.com/java/lang/Thread.html#suspend()) *for details.* |
| [java.util.Date.toGMTString()](http://docs.google.com/java/util/Date.html#toGMTString())  *As of JDK version 1.1, replaced by DateFormat.format(Date date), using a GMT TimeZone.* |
| [java.util.Date.toLocaleString()](http://docs.google.com/java/util/Date.html#toLocaleString())  *As of JDK version 1.1, replaced by DateFormat.format(Date date).* |
| [java.rmi.server.LogStream.toString()](http://docs.google.com/java/rmi/server/LogStream.html#toString())  *no replacement* |
| [java.rmi.server.Operation.toString()](http://docs.google.com/java/rmi/server/Operation.html#toString())  *no replacement* |
| [java.io.ByteArrayOutputStream.toString(int)](http://docs.google.com/java/io/ByteArrayOutputStream.html#toString(int))  *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the toString(String enc) method, which takes an encoding-name argument, or the toString() method, which uses the platform's default character encoding.* |
| [java.io.File.toURL()](http://docs.google.com/java/io/File.html#toURL())  *This method does not automatically escape characters that are illegal in URLs. It is recommended that new code convert an abstract pathname into a URL by first converting it into a URI, via the* [*toURI*](http://docs.google.com/java/io/File.html#toURI()) *method, and then converting the URI into a URL via the* [*URI.toURL*](http://docs.google.com/java/net/URI.html#toURL()) *method.* |
| [java.util.Date.UTC(int, int, int, int, int, int)](http://docs.google.com/java/util/Date.html#UTC(int,%20int,%20int,%20int,%20int,%20int))  *As of JDK version 1.1, replaced by Calendar.set(year + 1900, month, date, hrs, min, sec) or GregorianCalendar(year + 1900, month, date, hrs, min, sec), using a UTC TimeZone, followed by Calendar.getTime().getTime().* |
| [javax.xml.bind.Validator.validate(Object)](http://docs.google.com/javax/xml/bind/Validator.html#validate(java.lang.Object))  *since JAXB2.0* |
| [javax.xml.bind.Validator.validateRoot(Object)](http://docs.google.com/javax/xml/bind/Validator.html#validateRoot(java.lang.Object))  *since JAXB2.0* |
| [javax.swing.text.View.viewToModel(float, float, Shape)](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape)) |
| [org.omg.CORBA.portable.OutputStream.write\_Principal(Principal)](http://docs.google.com/org/omg/CORBA/portable/OutputStream.html#write_Principal(org.omg.CORBA.Principal))  *Deprecated by CORBA 2.2.* |
| [java.rmi.server.LogStream.write(byte[], int, int)](http://docs.google.com/java/rmi/server/LogStream.html#write(byte%5B%5D,%20int,%20int))  *no replacement* |
| [java.rmi.server.LogStream.write(int)](http://docs.google.com/java/rmi/server/LogStream.html#write(int))  *no replacement* |
| [java.io.ObjectOutputStream.PutField.write(ObjectOutput)](http://docs.google.com/java/io/ObjectOutputStream.PutField.html#write(java.io.ObjectOutput))  *This method does not write the values contained by this PutField object in a proper format, and may result in corruption of the serialization stream. The correct way to write PutField data is by calling the* [*ObjectOutputStream.writeFields()*](http://docs.google.com/java/io/ObjectOutputStream.html#writeFields()) *method.* |

| **Deprecated Constructors** | |
| --- | --- |
| [javax.management.AttributeValueExp()](http://docs.google.com/javax/management/AttributeValueExp.html#AttributeValueExp())  *An instance created with this constructor cannot be used in a query.* |
| [java.sql.Date(int, int, int)](http://docs.google.com/java/sql/Date.html#Date(int,%20int,%20int))  *instead use the constructor Date(long date)* |
| [java.util.Date(int, int, int)](http://docs.google.com/java/util/Date.html#Date(int,%20int,%20int))  *As of JDK version 1.1, replaced by Calendar.set(year + 1900, month, date) or GregorianCalendar(year + 1900, month, date).* |
| [java.util.Date(int, int, int, int, int)](http://docs.google.com/java/util/Date.html#Date(int,%20int,%20int,%20int,%20int))  *As of JDK version 1.1, replaced by Calendar.set(year + 1900, month, date, hrs, min) or GregorianCalendar(year + 1900, month, date, hrs, min).* |
| [java.util.Date(int, int, int, int, int, int)](http://docs.google.com/java/util/Date.html#Date(int,%20int,%20int,%20int,%20int,%20int))  *As of JDK version 1.1, replaced by Calendar.set(year + 1900, month, date, hrs, min, sec) or GregorianCalendar(year + 1900, month, date, hrs, min, sec).* |
| [java.util.Date(String)](http://docs.google.com/java/util/Date.html#Date(java.lang.String))  *As of JDK version 1.1, replaced by DateFormat.parse(String s).* |
| [java.awt.event.KeyEvent(Component, int, long, int, int)](http://docs.google.com/java/awt/event/KeyEvent.html#KeyEvent(java.awt.Component,%20int,%20long,%20int,%20int))  *as of JDK1.1* |
| [java.rmi.server.Operation(String)](http://docs.google.com/java/rmi/server/Operation.html#Operation(java.lang.String))  *no replacement* |
| [java.rmi.RMISecurityException(String)](http://docs.google.com/java/rmi/RMISecurityException.html#RMISecurityException(java.lang.String))  *no replacement* |
| [java.rmi.RMISecurityException(String, String)](http://docs.google.com/java/rmi/RMISecurityException.html#RMISecurityException(java.lang.String,%20java.lang.String))  *no replacement* |
| [java.rmi.ServerRuntimeException(String, Exception)](http://docs.google.com/java/rmi/ServerRuntimeException.html#ServerRuntimeException(java.lang.String,%20java.lang.Exception))  *no replacement* |
| [java.rmi.server.SkeletonMismatchException(String)](http://docs.google.com/java/rmi/server/SkeletonMismatchException.html#SkeletonMismatchException(java.lang.String))  *no replacement* |
| [java.net.Socket(InetAddress, int, boolean)](http://docs.google.com/java/net/Socket.html#Socket(java.net.InetAddress,%20int,%20boolean))  *Use DatagramSocket instead for UDP transport.* |
| [java.net.Socket(String, int, boolean)](http://docs.google.com/java/net/Socket.html#Socket(java.lang.String,%20int,%20boolean))  *Use DatagramSocket instead for UDP transport.* |
| [java.io.StreamTokenizer(InputStream)](http://docs.google.com/java/io/StreamTokenizer.html#StreamTokenizer(java.io.InputStream))  *As of JDK version 1.1, the preferred way to tokenize an input stream is to convert it into a character stream, for example:*  *Reader r = new BufferedReader(new InputStreamReader(is));  StreamTokenizer st = new StreamTokenizer(r);* |
| [java.lang.String(byte[], int)](http://docs.google.com/java/lang/String.html#String(byte%5B%5D,%20int))  *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a* [*Charset*](http://docs.google.com/java/nio/charset/Charset.html)*, charset name, or that use the platform's default charset.* |
| [java.lang.String(byte[], int, int, int)](http://docs.google.com/java/lang/String.html#String(byte%5B%5D,%20int,%20int,%20int))  *This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a* [*Charset*](http://docs.google.com/java/nio/charset/Charset.html)*, charset name, or that use the platform's default charset.* |
| [org.omg.CORBA.TCKind(int)](http://docs.google.com/org/omg/CORBA/TCKind.html#TCKind(int))  *Do not use this constructor as this method should be private according to the OMG specification. Use* [*TCKind.from\_int(int)*](http://docs.google.com/org/omg/CORBA/TCKind.html#from_int(int)) *instead.* |
| [java.sql.Time(int, int, int)](http://docs.google.com/java/sql/Time.html#Time(int,%20int,%20int))  *Use the constructor that takes a milliseconds value in place of this constructor* |
| [java.sql.Timestamp(int, int, int, int, int, int, int)](http://docs.google.com/java/sql/Timestamp.html#Timestamp(int,%20int,%20int,%20int,%20int,%20int,%20int))  *instead use the constructor Timestamp(long millis)* |

| **Deprecated Annotation Type Elements** | |
| --- | --- |
| [javax.jws.HandlerChain.name](http://docs.google.com/javax/jws/HandlerChain.html#name())  *As of JSR-181 2.0 with no replacement.* |

| | [**Overview**](http://docs.google.com/overview-summary.html) | Package | Class | Use | [**Tree**](http://docs.google.com/overview-tree.html) | **Deprecated** | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV   NEXT | [**FRAMES**](http://docs.google.com/index.html?deprecated-list.html)    [**NO FRAMES**](http://docs.google.com/deprecated-list.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).