| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EventQueue.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Event.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/FileDialog.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/EventQueue.html)    [**NO FRAMES**](http://docs.google.com/EventQueue.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.awt**

Class EventQueue

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.EventQueue**

public class **EventQueue**extends [Object](http://docs.google.com/java/lang/Object.html)

EventQueue is a platform-independent class that queues events, both from the underlying peer classes and from trusted application classes.

It encapsulates asynchronous event dispatch machinery which extracts events from the queue and dispatches them by calling [dispatchEvent(AWTEvent)](http://docs.google.com/java/awt/EventQueue.html#dispatchEvent(java.awt.AWTEvent)) method on this EventQueue with the event to be dispatched as an argument. The particular behavior of this machinery is implementation-dependent. The only requirements are that events which were actually enqueued to this queue (note that events being posted to the EventQueue can be coalesced) are dispatched:

Sequentially. That is, it is not permitted that several events from this queue are dispatched simultaneously. In the same order as they are enqueued. That is, if AWTEvent A is enqueued to the EventQueue before AWTEvent B then event B will not be dispatched before event A.

Some browsers partition applets in different code bases into separate contexts, and establish walls between these contexts. In such a scenario, there will be one EventQueue per context. Other browsers place all applets into the same context, implying that there will be only a single, global EventQueue for all applets. This behavior is implementation-dependent. Consult your browser's documentation for more information.

For information on the threading issues of the event dispatch machinery, see [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#Autoshutdown).

**Since:** 1.1

| **Constructor Summary** | |
| --- | --- |
| [**EventQueue**](http://docs.google.com/java/awt/EventQueue.html#EventQueue())() |

| **Method Summary** | |
| --- | --- |
| protected  void | [**dispatchEvent**](http://docs.google.com/java/awt/EventQueue.html#dispatchEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) event)            Dispatches an event. |
| static [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) | [**getCurrentEvent**](http://docs.google.com/java/awt/EventQueue.html#getCurrentEvent())()            Returns the the event currently being dispatched by the EventQueue associated with the calling thread. |
| static long | [**getMostRecentEventTime**](http://docs.google.com/java/awt/EventQueue.html#getMostRecentEventTime())()            Returns the timestamp of the most recent event that had a timestamp, and that was dispatched from the EventQueue associated with the calling thread. |
| [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) | [**getNextEvent**](http://docs.google.com/java/awt/EventQueue.html#getNextEvent())()            Removes an event from the EventQueue and returns it. |
| static void | [**invokeAndWait**](http://docs.google.com/java/awt/EventQueue.html#invokeAndWait(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) runnable)            Causes runnable to have its run method called in the dispatch thread of [the system EventQueue](http://docs.google.com/java/awt/Toolkit.html#getSystemEventQueue()). |
| static void | [**invokeLater**](http://docs.google.com/java/awt/EventQueue.html#invokeLater(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) runnable)            Causes runnable to have its run method called in the dispatch thread of [the system EventQueue](http://docs.google.com/java/awt/Toolkit.html#getSystemEventQueue()). |
| static boolean | [**isDispatchThread**](http://docs.google.com/java/awt/EventQueue.html#isDispatchThread())()            Returns true if the calling thread is the current AWT EventQueue's dispatch thread. |
| [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) | [**peekEvent**](http://docs.google.com/java/awt/EventQueue.html#peekEvent())()            Returns the first event on the EventQueue without removing it. |
| [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) | [**peekEvent**](http://docs.google.com/java/awt/EventQueue.html#peekEvent(int))(int id)            Returns the first event with the specified id, if any. |
| protected  void | [**pop**](http://docs.google.com/java/awt/EventQueue.html#pop())()            Stops dispatching events using this EventQueue. |
| void | [**postEvent**](http://docs.google.com/java/awt/EventQueue.html#postEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) theEvent)            Posts a 1.1-style event to the EventQueue. |
| void | [**push**](http://docs.google.com/java/awt/EventQueue.html#push(java.awt.EventQueue))([EventQueue](http://docs.google.com/java/awt/EventQueue.html) newEventQueue)            Replaces the existing EventQueue with the specified one. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### EventQueue

public **EventQueue**()

| **Method Detail** |
| --- |

### postEvent

public void **postEvent**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) theEvent)

Posts a 1.1-style event to the EventQueue. If there is an existing event on the queue with the same ID and event source, the source Component's coalesceEvents method will be called.

**Parameters:**theEvent - an instance of java.awt.AWTEvent, or a subclass of it **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if theEvent is null

### getNextEvent

public [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) **getNextEvent**()  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html)

Removes an event from the EventQueue and returns it. This method will block until an event has been posted by another thread.

**Returns:**the next AWTEvent **Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if any thread has interrupted this thread

### peekEvent

public [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) **peekEvent**()

Returns the first event on the EventQueue without removing it.

**Returns:**the first event

### peekEvent

public [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) **peekEvent**(int id)

Returns the first event with the specified id, if any.

**Parameters:**id - the id of the type of event desired **Returns:**the first event of the specified id or null if there is no such event

### dispatchEvent

protected void **dispatchEvent**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) event)

Dispatches an event. The manner in which the event is dispatched depends upon the type of the event and the type of the event's source object:

| Event Type | Source Type | Dispatched To |
| --- | --- | --- |
| ActiveEvent | Any | event.dispatch() |
| Other | Component | source.dispatchEvent(AWTEvent) |
| Other | MenuComponent | source.dispatchEvent(AWTEvent) |
| Other | Other | No action (ignored) |

**Parameters:**event - an instance of java.awt.AWTEvent, or a subclass of it **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if event is null**Since:** 1.2

### getMostRecentEventTime

public static long **getMostRecentEventTime**()

Returns the timestamp of the most recent event that had a timestamp, and that was dispatched from the EventQueue associated with the calling thread. If an event with a timestamp is currently being dispatched, its timestamp will be returned. If no events have yet been dispatched, the EventQueue's initialization time will be returned instead.In the current version of the JDK, only InputEvents, ActionEvents, and InvocationEvents have timestamps; however, future versions of the JDK may add timestamps to additional event types. Note that this method should only be invoked from an application's event dispatching thread. If this method is invoked from another thread, the current system time (as reported by System.currentTimeMillis()) will be returned instead.

**Returns:**the timestamp of the last InputEvent, ActionEvent, or InvocationEvent to be dispatched, or System.currentTimeMillis() if this method is invoked on a thread other than an event dispatching thread**Since:** 1.4 **See Also:**[InputEvent.getWhen()](http://docs.google.com/java/awt/event/InputEvent.html#getWhen()), [ActionEvent.getWhen()](http://docs.google.com/java/awt/event/ActionEvent.html#getWhen()), [InvocationEvent.getWhen()](http://docs.google.com/java/awt/event/InvocationEvent.html#getWhen())

### getCurrentEvent

public static [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) **getCurrentEvent**()

Returns the the event currently being dispatched by the EventQueue associated with the calling thread. This is useful if a method needs access to the event, but was not designed to receive a reference to it as an argument. Note that this method should only be invoked from an application's event dispatching thread. If this method is invoked from another thread, null will be returned.

**Returns:**the event currently being dispatched, or null if this method is invoked on a thread other than an event dispatching thread**Since:** 1.4

### push

public void **push**([EventQueue](http://docs.google.com/java/awt/EventQueue.html) newEventQueue)

Replaces the existing EventQueue with the specified one. Any pending events are transferred to the new EventQueue for processing by it.

**Parameters:**newEventQueue - an EventQueue (or subclass thereof) instance to be use **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if newEventQueue is null**Since:** 1.2 **See Also:**[pop()](http://docs.google.com/java/awt/EventQueue.html#pop())

### pop

protected void **pop**()  
 throws [EmptyStackException](http://docs.google.com/java/util/EmptyStackException.html)

Stops dispatching events using this EventQueue. Any pending events are transferred to the previous EventQueue for processing.

Warning: To avoid deadlock, do not declare this method synchronized in a subclass.

**Throws:** [EmptyStackException](http://docs.google.com/java/util/EmptyStackException.html) - if no previous push was made on this EventQueue**Since:** 1.2 **See Also:**[push(java.awt.EventQueue)](http://docs.google.com/java/awt/EventQueue.html#push(java.awt.EventQueue))

### isDispatchThread

public static boolean **isDispatchThread**()

Returns true if the calling thread is the current AWT EventQueue's dispatch thread. Use this call the ensure that a given task is being executed (or not being) on the current AWT EventDispatchThread.

**Returns:**true if running on the current AWT EventQueue's dispatch thread**Since:** 1.2

### invokeLater

public static void **invokeLater**([Runnable](http://docs.google.com/java/lang/Runnable.html) runnable)

Causes runnable to have its run method called in the dispatch thread of [the system EventQueue](http://docs.google.com/java/awt/Toolkit.html#getSystemEventQueue()). This will happen after all pending events are processed.

**Parameters:**runnable - the Runnable whose run method should be executed synchronously on the EventQueue**Since:** 1.2 **See Also:**[invokeAndWait(java.lang.Runnable)](http://docs.google.com/java/awt/EventQueue.html#invokeAndWait(java.lang.Runnable))

### invokeAndWait

public static void **invokeAndWait**([Runnable](http://docs.google.com/java/lang/Runnable.html) runnable)  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html),  
 [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html)

Causes runnable to have its run method called in the dispatch thread of [the system EventQueue](http://docs.google.com/java/awt/Toolkit.html#getSystemEventQueue()). This will happen after all pending events are processed. The call blocks until this has happened. This method will throw an Error if called from the event dispatcher thread.

**Parameters:**runnable - the Runnable whose run method should be executed synchronously on the EventQueue **Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if any thread has interrupted this thread [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html) - if an throwable is thrown when running runnable**Since:** 1.2 **See Also:**[invokeLater(java.lang.Runnable)](http://docs.google.com/java/awt/EventQueue.html#invokeLater(java.lang.Runnable))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EventQueue.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Event.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/FileDialog.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/EventQueue.html)    [**NO FRAMES**](http://docs.google.com/EventQueue.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).