| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EventDirContext.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/javax/naming/event/EventContext.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/event/NamespaceChangeListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/event/EventDirContext.html)    [**NO FRAMES**](http://docs.google.com/EventDirContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#1t3h5sf) | DETAIL: FIELD | CONSTR | [METHOD](#3rdcrjn) |

## **javax.naming.event**

Interface EventDirContext

**All Superinterfaces:** [Context](http://docs.google.com/javax/naming/Context.html), [DirContext](http://docs.google.com/javax/naming/directory/DirContext.html), [EventContext](http://docs.google.com/javax/naming/event/EventContext.html)

public interface **EventDirContext**extends [EventContext](http://docs.google.com/javax/naming/event/EventContext.html), [DirContext](http://docs.google.com/javax/naming/directory/DirContext.html)

Contains methods for registering listeners to be notified of events fired when objects named in a directory context changes.

The methods in this interface support identification of objects by [RFC 2254](ftp://ftp.isi.edu/in-notes/rfc2254.txt) search filters.

Using the search filter, it is possible to register interest in objects that do not exist at the time of registration but later come into existence and satisfy the filter. However, there might be limitations in the extent to which this can be supported by the service provider and underlying protocol/service. If the caller submits a filter that cannot be supported in this way, addNamingListener() throws an InvalidSearchFilterException.

See EventContext for a description of event source and target, and information about listener registration/deregistration that are also applicable to methods in this interface. See the [package description](http://docs.google.com/package-summary.html#THREADING) for information on threading issues.

A SearchControls or array object passed as a parameter to any method is owned by the caller. The service provider will not modify the object or keep a reference to it.

**Since:** 1.3

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface javax.naming.event.**[**EventContext**](http://docs.google.com/javax/naming/event/EventContext.html) |
| --- |
| [OBJECT\_SCOPE](http://docs.google.com/javax/naming/event/EventContext.html#OBJECT_SCOPE), [ONELEVEL\_SCOPE](http://docs.google.com/javax/naming/event/EventContext.html#ONELEVEL_SCOPE), [SUBTREE\_SCOPE](http://docs.google.com/javax/naming/event/EventContext.html#SUBTREE_SCOPE) |

| **Fields inherited from interface javax.naming.directory.**[**DirContext**](http://docs.google.com/javax/naming/directory/DirContext.html) |
| --- |
| [ADD\_ATTRIBUTE](http://docs.google.com/javax/naming/directory/DirContext.html#ADD_ATTRIBUTE), [REMOVE\_ATTRIBUTE](http://docs.google.com/javax/naming/directory/DirContext.html#REMOVE_ATTRIBUTE), [REPLACE\_ATTRIBUTE](http://docs.google.com/javax/naming/directory/DirContext.html#REPLACE_ATTRIBUTE) |

| **Fields inherited from interface javax.naming.**[**Context**](http://docs.google.com/javax/naming/Context.html) |
| --- |
| [APPLET](http://docs.google.com/javax/naming/Context.html#APPLET), [AUTHORITATIVE](http://docs.google.com/javax/naming/Context.html#AUTHORITATIVE), [BATCHSIZE](http://docs.google.com/javax/naming/Context.html#BATCHSIZE), [DNS\_URL](http://docs.google.com/javax/naming/Context.html#DNS_URL), [INITIAL\_CONTEXT\_FACTORY](http://docs.google.com/javax/naming/Context.html#INITIAL_CONTEXT_FACTORY), [LANGUAGE](http://docs.google.com/javax/naming/Context.html#LANGUAGE), [OBJECT\_FACTORIES](http://docs.google.com/javax/naming/Context.html#OBJECT_FACTORIES), [PROVIDER\_URL](http://docs.google.com/javax/naming/Context.html#PROVIDER_URL), [REFERRAL](http://docs.google.com/javax/naming/Context.html#REFERRAL), [SECURITY\_AUTHENTICATION](http://docs.google.com/javax/naming/Context.html#SECURITY_AUTHENTICATION), [SECURITY\_CREDENTIALS](http://docs.google.com/javax/naming/Context.html#SECURITY_CREDENTIALS), [SECURITY\_PRINCIPAL](http://docs.google.com/javax/naming/Context.html#SECURITY_PRINCIPAL), [SECURITY\_PROTOCOL](http://docs.google.com/javax/naming/Context.html#SECURITY_PROTOCOL), [STATE\_FACTORIES](http://docs.google.com/javax/naming/Context.html#STATE_FACTORIES), [URL\_PKG\_PREFIXES](http://docs.google.com/javax/naming/Context.html#URL_PKG_PREFIXES) |

| **Method Summary** | |
| --- | --- |
| void | [**addNamingListener**](http://docs.google.com/javax/naming/event/EventDirContext.html#addNamingListener(javax.naming.Name,%20java.lang.String,%20java.lang.Object%5B%5D,%20javax.naming.directory.SearchControls,%20javax.naming.event.NamingListener))([Name](http://docs.google.com/javax/naming/Name.html) target, [String](http://docs.google.com/java/lang/String.html) filter, [Object](http://docs.google.com/java/lang/Object.html)[] filterArgs, [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls, [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)            Adds a listener for receiving naming events fired when objects identified by the search filter filter and filter arguments at the object named by the target are modified. |
| void | [**addNamingListener**](http://docs.google.com/javax/naming/event/EventDirContext.html#addNamingListener(javax.naming.Name,%20java.lang.String,%20javax.naming.directory.SearchControls,%20javax.naming.event.NamingListener))([Name](http://docs.google.com/javax/naming/Name.html) target, [String](http://docs.google.com/java/lang/String.html) filter, [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls, [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)            Adds a listener for receiving naming events fired when objects identified by the search filter filter at the object named by target are modified. |
| void | [**addNamingListener**](http://docs.google.com/javax/naming/event/EventDirContext.html#addNamingListener(java.lang.String,%20java.lang.String,%20java.lang.Object%5B%5D,%20javax.naming.directory.SearchControls,%20javax.naming.event.NamingListener))([String](http://docs.google.com/java/lang/String.html) target, [String](http://docs.google.com/java/lang/String.html) filter, [Object](http://docs.google.com/java/lang/Object.html)[] filterArgs, [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls, [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)            Adds a listener for receiving naming events fired when objects identified by the search filter filter and filter arguments at the object named by the string target name are modified. |
| void | [**addNamingListener**](http://docs.google.com/javax/naming/event/EventDirContext.html#addNamingListener(java.lang.String,%20java.lang.String,%20javax.naming.directory.SearchControls,%20javax.naming.event.NamingListener))([String](http://docs.google.com/java/lang/String.html) target, [String](http://docs.google.com/java/lang/String.html) filter, [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls, [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)            Adds a listener for receiving naming events fired when objects identified by the search filter filter at the object named by the string target name are modified. |

| **Methods inherited from interface javax.naming.event.**[**EventContext**](http://docs.google.com/javax/naming/event/EventContext.html) |
| --- |
| [addNamingListener](http://docs.google.com/javax/naming/event/EventContext.html#addNamingListener(javax.naming.Name,%20int,%20javax.naming.event.NamingListener)), [addNamingListener](http://docs.google.com/javax/naming/event/EventContext.html#addNamingListener(java.lang.String,%20int,%20javax.naming.event.NamingListener)), [removeNamingListener](http://docs.google.com/javax/naming/event/EventContext.html#removeNamingListener(javax.naming.event.NamingListener)), [targetMustExist](http://docs.google.com/javax/naming/event/EventContext.html#targetMustExist()) |

| **Methods inherited from interface javax.naming.directory.**[**DirContext**](http://docs.google.com/javax/naming/directory/DirContext.html) |
| --- |
| [bind](http://docs.google.com/javax/naming/directory/DirContext.html#bind(javax.naming.Name,%20java.lang.Object,%20javax.naming.directory.Attributes)), [bind](http://docs.google.com/javax/naming/directory/DirContext.html#bind(java.lang.String,%20java.lang.Object,%20javax.naming.directory.Attributes)), [createSubcontext](http://docs.google.com/javax/naming/directory/DirContext.html#createSubcontext(javax.naming.Name,%20javax.naming.directory.Attributes)), [createSubcontext](http://docs.google.com/javax/naming/directory/DirContext.html#createSubcontext(java.lang.String,%20javax.naming.directory.Attributes)), [getAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#getAttributes(javax.naming.Name)), [getAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#getAttributes(javax.naming.Name,%20java.lang.String%5B%5D)), [getAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#getAttributes(java.lang.String)), [getAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#getAttributes(java.lang.String,%20java.lang.String%5B%5D)), [getSchema](http://docs.google.com/javax/naming/directory/DirContext.html#getSchema(javax.naming.Name)), [getSchema](http://docs.google.com/javax/naming/directory/DirContext.html#getSchema(java.lang.String)), [getSchemaClassDefinition](http://docs.google.com/javax/naming/directory/DirContext.html#getSchemaClassDefinition(javax.naming.Name)), [getSchemaClassDefinition](http://docs.google.com/javax/naming/directory/DirContext.html#getSchemaClassDefinition(java.lang.String)), [modifyAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#modifyAttributes(javax.naming.Name,%20int,%20javax.naming.directory.Attributes)), [modifyAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#modifyAttributes(javax.naming.Name,%20javax.naming.directory.ModificationItem%5B%5D)), [modifyAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#modifyAttributes(java.lang.String,%20int,%20javax.naming.directory.Attributes)), [modifyAttributes](http://docs.google.com/javax/naming/directory/DirContext.html#modifyAttributes(java.lang.String,%20javax.naming.directory.ModificationItem%5B%5D)), [rebind](http://docs.google.com/javax/naming/directory/DirContext.html#rebind(javax.naming.Name,%20java.lang.Object,%20javax.naming.directory.Attributes)), [rebind](http://docs.google.com/javax/naming/directory/DirContext.html#rebind(java.lang.String,%20java.lang.Object,%20javax.naming.directory.Attributes)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20javax.naming.directory.Attributes)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20javax.naming.directory.Attributes,%20java.lang.String%5B%5D)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20java.lang.String,%20java.lang.Object%5B%5D,%20javax.naming.directory.SearchControls)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20java.lang.String,%20javax.naming.directory.SearchControls)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(java.lang.String,%20javax.naming.directory.Attributes)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(java.lang.String,%20javax.naming.directory.Attributes,%20java.lang.String%5B%5D)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(java.lang.String,%20java.lang.String,%20java.lang.Object%5B%5D,%20javax.naming.directory.SearchControls)), [search](http://docs.google.com/javax/naming/directory/DirContext.html#search(java.lang.String,%20java.lang.String,%20javax.naming.directory.SearchControls)) |

| **Methods inherited from interface javax.naming.**[**Context**](http://docs.google.com/javax/naming/Context.html) |
| --- |
| [addToEnvironment](http://docs.google.com/javax/naming/Context.html#addToEnvironment(java.lang.String,%20java.lang.Object)), [bind](http://docs.google.com/javax/naming/Context.html#bind(javax.naming.Name,%20java.lang.Object)), [bind](http://docs.google.com/javax/naming/Context.html#bind(java.lang.String,%20java.lang.Object)), [close](http://docs.google.com/javax/naming/Context.html#close()), [composeName](http://docs.google.com/javax/naming/Context.html#composeName(javax.naming.Name,%20javax.naming.Name)), [composeName](http://docs.google.com/javax/naming/Context.html#composeName(java.lang.String,%20java.lang.String)), [createSubcontext](http://docs.google.com/javax/naming/Context.html#createSubcontext(javax.naming.Name)), [createSubcontext](http://docs.google.com/javax/naming/Context.html#createSubcontext(java.lang.String)), [destroySubcontext](http://docs.google.com/javax/naming/Context.html#destroySubcontext(javax.naming.Name)), [destroySubcontext](http://docs.google.com/javax/naming/Context.html#destroySubcontext(java.lang.String)), [getEnvironment](http://docs.google.com/javax/naming/Context.html#getEnvironment()), [getNameInNamespace](http://docs.google.com/javax/naming/Context.html#getNameInNamespace()), [getNameParser](http://docs.google.com/javax/naming/Context.html#getNameParser(javax.naming.Name)), [getNameParser](http://docs.google.com/javax/naming/Context.html#getNameParser(java.lang.String)), [list](http://docs.google.com/javax/naming/Context.html#list(javax.naming.Name)), [list](http://docs.google.com/javax/naming/Context.html#list(java.lang.String)), [listBindings](http://docs.google.com/javax/naming/Context.html#listBindings(javax.naming.Name)), [listBindings](http://docs.google.com/javax/naming/Context.html#listBindings(java.lang.String)), [lookup](http://docs.google.com/javax/naming/Context.html#lookup(javax.naming.Name)), [lookup](http://docs.google.com/javax/naming/Context.html#lookup(java.lang.String)), [lookupLink](http://docs.google.com/javax/naming/Context.html#lookupLink(javax.naming.Name)), [lookupLink](http://docs.google.com/javax/naming/Context.html#lookupLink(java.lang.String)), [rebind](http://docs.google.com/javax/naming/Context.html#rebind(javax.naming.Name,%20java.lang.Object)), [rebind](http://docs.google.com/javax/naming/Context.html#rebind(java.lang.String,%20java.lang.Object)), [removeFromEnvironment](http://docs.google.com/javax/naming/Context.html#removeFromEnvironment(java.lang.String)), [rename](http://docs.google.com/javax/naming/Context.html#rename(javax.naming.Name,%20javax.naming.Name)), [rename](http://docs.google.com/javax/naming/Context.html#rename(java.lang.String,%20java.lang.String)), [unbind](http://docs.google.com/javax/naming/Context.html#unbind(javax.naming.Name)), [unbind](http://docs.google.com/javax/naming/Context.html#unbind(java.lang.String)) |

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

### addNamingListener

void **addNamingListener**([Name](http://docs.google.com/javax/naming/Name.html) target,  
 [String](http://docs.google.com/java/lang/String.html) filter,  
 [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls,  
 [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Adds a listener for receiving naming events fired when objects identified by the search filter filter at the object named by target are modified.

The scope, returningObj flag, and returningAttributes flag from the search controls ctls are used to control the selection of objects that the listener is interested in, and determines what information is returned in the eventual NamingEvent object. Note that the requested information to be returned might not be present in the NamingEvent object if they are unavailable or could not be obtained by the service provider or service.

**Parameters:**target - The nonnull name of the object resolved relative to this context.filter - The nonnull string filter (see RFC2254).ctls - The possibly null search controls. If null, the default search controls are used.l - The nonnull listener. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a problem was encountered while adding the listener.**See Also:**[EventContext.removeNamingListener(javax.naming.event.NamingListener)](http://docs.google.com/javax/naming/event/EventContext.html#removeNamingListener(javax.naming.event.NamingListener)), [DirContext.search(javax.naming.Name, java.lang.String, javax.naming.directory.SearchControls)](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20java.lang.String,%20javax.naming.directory.SearchControls))

### addNamingListener

void **addNamingListener**([String](http://docs.google.com/java/lang/String.html) target,  
 [String](http://docs.google.com/java/lang/String.html) filter,  
 [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls,  
 [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Adds a listener for receiving naming events fired when objects identified by the search filter filter at the object named by the string target name are modified. See the overload that accepts a Name for details of how this method behaves.

**Parameters:**target - The nonnull string name of the object resolved relative to this context.filter - The nonnull string filter (see RFC2254).ctls - The possibly null search controls. If null, the default search controls is used.l - The nonnull listener. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a problem was encountered while adding the listener.**See Also:**[EventContext.removeNamingListener(javax.naming.event.NamingListener)](http://docs.google.com/javax/naming/event/EventContext.html#removeNamingListener(javax.naming.event.NamingListener)), [DirContext.search(java.lang.String, java.lang.String, javax.naming.directory.SearchControls)](http://docs.google.com/javax/naming/directory/DirContext.html#search(java.lang.String,%20java.lang.String,%20javax.naming.directory.SearchControls))

### addNamingListener

void **addNamingListener**([Name](http://docs.google.com/javax/naming/Name.html) target,  
 [String](http://docs.google.com/java/lang/String.html) filter,  
 [Object](http://docs.google.com/java/lang/Object.html)[] filterArgs,  
 [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls,  
 [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Adds a listener for receiving naming events fired when objects identified by the search filter filter and filter arguments at the object named by the target are modified. The scope, returningObj flag, and returningAttributes flag from the search controls ctls are used to control the selection of objects that the listener is interested in, and determines what information is returned in the eventual NamingEvent object. Note that the requested information to be returned might not be present in the NamingEvent object if they are unavailable or could not be obtained by the service provider or service.

**Parameters:**target - The nonnull name of the object resolved relative to this context.filter - The nonnull string filter (see RFC2254).filterArgs - The possibly null array of arguments for the filter.ctls - The possibly null search controls. If null, the default search controls are used.l - The nonnull listener. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a problem was encountered while adding the listener.**See Also:**[EventContext.removeNamingListener(javax.naming.event.NamingListener)](http://docs.google.com/javax/naming/event/EventContext.html#removeNamingListener(javax.naming.event.NamingListener)), [DirContext.search(javax.naming.Name, java.lang.String, java.lang.Object[], javax.naming.directory.SearchControls)](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20java.lang.String,%20java.lang.Object%5B%5D,%20javax.naming.directory.SearchControls))

### addNamingListener

void **addNamingListener**([String](http://docs.google.com/java/lang/String.html) target,  
 [String](http://docs.google.com/java/lang/String.html) filter,  
 [Object](http://docs.google.com/java/lang/Object.html)[] filterArgs,  
 [SearchControls](http://docs.google.com/javax/naming/directory/SearchControls.html) ctls,  
 [NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) l)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Adds a listener for receiving naming events fired when objects identified by the search filter filter and filter arguments at the object named by the string target name are modified. See the overload that accepts a Name for details of how this method behaves.

**Parameters:**target - The nonnull string name of the object resolved relative to this context.filter - The nonnull string filter (see RFC2254).filterArgs - The possibly null array of arguments for the filter.ctls - The possibly null search controls. If null, the default search controls is used.l - The nonnull listener. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a problem was encountered while adding the listener.**See Also:**[EventContext.removeNamingListener(javax.naming.event.NamingListener)](http://docs.google.com/javax/naming/event/EventContext.html#removeNamingListener(javax.naming.event.NamingListener)), [DirContext.search(java.lang.String, java.lang.String, java.lang.Object[], javax.naming.directory.SearchControls)](http://docs.google.com/javax/naming/directory/DirContext.html#search(java.lang.String,%20java.lang.String,%20java.lang.Object%5B%5D,%20javax.naming.directory.SearchControls))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EventDirContext.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/javax/naming/event/EventContext.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/event/NamespaceChangeListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/event/EventDirContext.html)    [**NO FRAMES**](http://docs.google.com/EventDirContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#1t3h5sf) | DETAIL: FIELD | CONSTR | [METHOD](#3rdcrjn) |

[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).