| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/naming/event/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/naming/spi/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/ldap/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.naming.ldap

Provides support for LDAPv3 extended operations and controls.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**Control**](http://docs.google.com/javax/naming/ldap/Control.html) | This interface represents an LDAPv3 control as defined in [RFC 2251](ftp://ftp.isi.edu/in-notes/rfc2251.txt). |
| [**ExtendedRequest**](http://docs.google.com/javax/naming/ldap/ExtendedRequest.html) | This interface represents an LDAPv3 extended operation request as defined in [RFC 2251](ftp://ftp.isi.edu/in-notes/rfc2251.txt). |
| [**ExtendedResponse**](http://docs.google.com/javax/naming/ldap/ExtendedResponse.html) | This interface represents an LDAP extended operation response as defined in [RFC 2251](ftp://ftp.isi.edu/in-notes/rfc2251.txt). |
| [**HasControls**](http://docs.google.com/javax/naming/ldap/HasControls.html) | This interface is for returning controls with objects returned in NamingEnumerations. |
| [**LdapContext**](http://docs.google.com/javax/naming/ldap/LdapContext.html) | This interface represents a context in which you can perform operations with LDAPv3-style controls and perform LDAPv3-style extended operations. |
| [**UnsolicitedNotification**](http://docs.google.com/javax/naming/ldap/UnsolicitedNotification.html) | This interface represents an unsolicited notification as defined in [RFC 2251](ftp://ftp.isi.edu/in-notes/rfc2251.txt). |
| [**UnsolicitedNotificationListener**](http://docs.google.com/javax/naming/ldap/UnsolicitedNotificationListener.html) | This interface is for handling UnsolicitedNotificationEvent. |

| **Class Summary** | |
| --- | --- |
| [**BasicControl**](http://docs.google.com/javax/naming/ldap/BasicControl.html) | This class provides a basic implementation of the Control interface. |
| [**ControlFactory**](http://docs.google.com/javax/naming/ldap/ControlFactory.html) | This abstract class represents a factory for creating LDAPv3 controls. |
| [**InitialLdapContext**](http://docs.google.com/javax/naming/ldap/InitialLdapContext.html) | This class is the starting context for performing LDAPv3-style extended operations and controls. |
| [**LdapName**](http://docs.google.com/javax/naming/ldap/LdapName.html) | This class represents a distinguished name as specified by [RFC 2253](http://ietf.org//rfc/rfc2253.txt). |
| [**ManageReferralControl**](http://docs.google.com/javax/naming/ldap/ManageReferralControl.html) | Requests that referral and other special LDAP objects be manipulated as normal LDAP objects. |
| [**PagedResultsControl**](http://docs.google.com/javax/naming/ldap/PagedResultsControl.html) | Requests that the results of a search operation be returned by the LDAP server in batches of a specified size. |
| [**PagedResultsResponseControl**](http://docs.google.com/javax/naming/ldap/PagedResultsResponseControl.html) | Indicates the end of a batch of search results. |
| [**Rdn**](http://docs.google.com/javax/naming/ldap/Rdn.html) | This class represents a relative distinguished name, or RDN, which is a component of a distinguished name as specified by [RFC 2253](http://ietf.org/rfc/rfc2253.txt). |
| [**SortControl**](http://docs.google.com/javax/naming/ldap/SortControl.html) | Requests that the results of a search operation be sorted by the LDAP server before being returned. |
| [**SortKey**](http://docs.google.com/javax/naming/ldap/SortKey.html) | A sort key and its associated sort parameters. |
| [**SortResponseControl**](http://docs.google.com/javax/naming/ldap/SortResponseControl.html) | Indicates whether the requested sort of search results was successful or not. |
| [**StartTlsRequest**](http://docs.google.com/javax/naming/ldap/StartTlsRequest.html) | This class implements the LDAPv3 Extended Request for StartTLS as defined in [Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security](http://www.ietf.org/rfc/rfc2830.txt) The object identifier for StartTLS is 1.3.6.1.4.1.1466.20037 and no extended request value is defined. |
| [**StartTlsResponse**](http://docs.google.com/javax/naming/ldap/StartTlsResponse.html) | This class implements the LDAPv3 Extended Response for StartTLS as defined in [Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security](http://www.ietf.org/rfc/rfc2830.txt) The object identifier for StartTLS is 1.3.6.1.4.1.1466.20037 and no extended response value is defined. |
| [**UnsolicitedNotificationEvent**](http://docs.google.com/javax/naming/ldap/UnsolicitedNotificationEvent.html) | This class represents an event fired in response to an unsolicited notification sent by the LDAP server. |

| **Exception Summary** | |
| --- | --- |
| [**LdapReferralException**](http://docs.google.com/javax/naming/ldap/LdapReferralException.html) | This abstract class is used to represent an LDAP referral exception. |

## Package javax.naming.ldap Description

Provides support for LDAPv3 extended operations and controls.

This package extends the directory operations of the Java Naming and Directory InterfaceTM (JNDI).   JNDI provides naming and directory functionality to applications written in the Java programming language. It is designed to be independent of any specific naming or directory service implementation. Thus a variety of services--new, emerging, and already deployed ones--can be accessed in a common way.

This package is for applications and service providers that deal with LDAPv3 extended operations and controls, as defined by [RFC 2251](http://www.ietf.org/rfc/rfc2251.txt). The core interface in this package is LdapContext, which defines methods on a context for performing extended operations and handling controls.

#### Extended Operations

This package defines the interface ExtendedRequest to represent the argument to an extended operation, and the interface ExtendedResponse to represent the result of the extended operation. An extended response is always paired with an extended request but not necessarily vice versa. That is, you can have an extended request that has no corresponding extended response.

An application typically does not deal directly with these interfaces. Instead, it deals with classes that *implement* these interfaces. The application gets these classes either as part of a repertoire of extended operations standardized through the IETF, or from directory vendors for vendor-specific extended operations. The request classes should have constructors that accept arguments in a type-safe and user-friendly manner, while the response classes should have access methods for getting the data of the response in a type-safe and user-friendly manner. Internally, the request/response classes deal with encoding and decoding BER values.

For example, suppose an LDAP server supports a "get time" extended operation. It would supply classes such as GetTimeRequest and GetTimeResponse, so that applications can use this feature. An application would use these classes as follows:

GetTimeResponse resp =  
 (GetTimeResponse) ectx.extendedOperation(new GetTimeRequest());  
long time = resp.getTime();

The GetTimeRequest and GetTimeResponse classes might be defined as follows:

public class GetTimeRequest implements ExtendedRequest {  
 // User-friendly constructor   
 public GetTimeRequest() {  
 };  
  
 // Methods used by service providers  
 public String getID() {  
 return GETTIME\_REQ\_OID;  
 }  
 public byte[] getEncodedValue() {  
 return null; // no value needed for get time request  
 }  
 public ExtendedResponse createExtendedResponse(  
 String id, byte[] berValue, int offset, int length) throws NamingException {  
 return new GetTimeResponse(id, berValue, offset, length);  
 }  
}  
public class GetTimeResponse() implements ExtendedResponse {  
 long time;  
 // called by GetTimeRequest.createExtendedResponse()  
 public GetTimeResponse(String id, byte[] berValue, int offset, int length)   
 throws NamingException {  
 // check validity of id  
 long time = ... // decode berValue to get time  
 }  
  
 // Type-safe and User-friendly methods  
 public java.util.Date getDate() { return new java.util.Date(time); }  
 public long getTime() { return time; }  
  
 // Low level methods  
 public byte[] getEncodedValue() {  
 return // berValue saved;  
 }  
 public String getID() {  
 return GETTIME\_RESP\_OID;  
 }  
}

#### Controls

This package defines the interface Control to represent an LDAPv3 control. It can be a control that is sent to an LDAP server (*request control*) or a control returned by an LDAP server (*response control*). Unlike extended requests and responses, there is not necessarily any pairing between request controls and response controls. You can send request controls and expect no response controls back, or receive response controls without sending any request controls.

An application typically does not deal directly with this interface. Instead, it deals with classes that *implement* this interface. The application gets control classes either as part of a repertoire of controls standardized through the IETF, or from directory vendors for vendor-specific controls. The request control classes should have constructors that accept arguments in a type-safe and user-friendly manner, while the response control classes should have access methods for getting the data of the response in a type-safe and user-friendly manner. Internally, the request/response control classes deal with encoding and decoding BER values.

For example, suppose an LDAP server supports a "signed results" request control, which when sent with a request, asks the server to digitally sign the results of an operation. It would supply a class SignedResultsControl so that applications can use this feature. An application would use this class as follows:

Control[] reqCtls = new Control[] {new SignedResultsControl(Control.CRITICAL)};  
ectx.setRequestControls(reqCtls);  
NamingEnumeration enum = ectx.search(...);

The SignedResultsControl class might be defined as follows:

public class SignedResultsControl implements Control {  
 // User-friendly constructor   
 public SignedResultsControl(boolean criticality) {  
 // assemble the components of the request control  
 };  
  
 // Methods used by service providers  
 public String getID() {  
 return // control's object identifier  
 }  
 public byte[] getEncodedValue() {  
 return // ASN.1 BER encoded control value  
 }  
 ...  
}

When a service provider receives response controls, it uses the ControlFactory class to produce specific classes that implement the Control interface.

An LDAP server can send back response controls with an LDAP operation and also with enumeration results, such as those returned by a list or search operation. The LdapContext provides a method (getResponseControls()) for getting the response controls sent with an LDAP operation, while the HasControls interface is used to retrieve response controls associated with enumeration results.

For example, suppose an LDAP server sends back a "change ID" control in response to a successful modification. It would supply a class ChangeIDControl so that the application can use this feature. An application would perform an update, and then try to get the change ID.

// Perform update  
Context ctx = ectx.createSubsubcontext("cn=newobj");  
  
// Get response controls  
Control[] respCtls = ectx.getResponseControls();  
if (respCtls != null) {  
 // Find the one we want  
 for (int i = 0; i < respCtls; i++) {  
 if(respCtls[i] instanceof ChangeIDControl) {  
 ChangeIDControl cctl = (ChangeIDControl)respCtls[i];  
 System.out.println(cctl.getChangeID());  
 }  
 }  
}

The vendor might supply the following ChangeIDControl and VendorXControlFactory classes. The VendorXControlFactory will be used by the service provider when the provider receives response controls from the LDAP server.

public class ChangeIDControl implements Control {  
 long id;  
  
 // Constructor used by ControlFactory  
 public ChangeIDControl(String OID, byte[] berVal) throws NamingException {  
 // check validity of OID  
 id = // extract change ID from berVal  
 };  
  
 // Type-safe and User-friendly method  
 public long getChangeID() {  
 return id;  
 }  
  
 // Low-level methods  
 public String getID() {  
 return CHANGEID\_OID;  
 }  
 public byte[] getEncodedValue() {  
 return // original berVal  
 }  
 ...  
}  
public class VendorXControlFactory extends ControlFactory {  
 public VendorXControlFactory () {  
 }  
  
 public Control getControlInstance(Control orig) throws NamingException {  
 if (isOneOfMyControls(orig.getID())) {  
 ...   
  
 // determine which of ours it is and call its constructor  
 return (new ChangeIDControl(orig.getID(), orig.getEncodedValue()));  
 }  
 return null; // not one of ours  
 }  
}

## Package Specification

The JNDI API Specification and related documents can be found in the [JNDI documentation](http://docs.google.com/technotes/guides/jndi/index.html).

**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/naming/event/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/naming/spi/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/ldap/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.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).