The Java LDAP Application Program Interface

draft-ietf-ldapext-alt-ldap-java-api-00.txt

STATUS OF THIS MEMO

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Distribution of this document is unlimited. Please send comments to the authors or the LDAPEXT mailing list, ietf-ldapext@netscape.com.

Copyright Notice: Copyright (C) The Internet Society (1999). All Rights Reserved.

ABSTRACT

This document defines a java language application program interface to the lightweight directory access protocol (LDAP), in the form of a class library. It provides an alternative to the draft by R. Weltman, T. Howes, M. Smith, C. Ho in describing significant additions and differences in design.
1. Overview

1.1 Preface

This document describes a Java(tm) API to the Light Weight Directory Access Protocol (LDAP). It specifies an API for LDAPv2 as specified in Internet RFC 1777, and LDAPv3 as specified in Internet RFCs 2251, 2252, 2253, 2254, 2255, and 2256.

1.2 Introduction

The Java API offers users a simple, robust way of creating applications that interact with InnosoftÂ’s and other vendor’s LDAP servers (protocol versions 2 or 3). In addition to properties inherent in the Java language itself (object-oriented, robust, secure, architecture neutral, portable, multi-threaded, and dynamic), the Java API uses a simple object hierarchy that makes it easy for application designers with at least basic knowledge of LDAP to get started right away. In addition, the Java API closely implements the LDAPv3 definitions outlined in RFCs 2251 through 2256.

1.2.1 Summary of Packages

org.ietf.ldap

Provides classes that model LDAP directory entries and their components, as well as components of the protocol that apply to both clients and servers.

org.ietf.ldap.client

Provides classes that model LDAP protocol components of that are particularly relevant to client-side applications.

org.ietf.ldap.ldif

Provides classes that implement reading and writing LDAP Interchange Format streams.

org.ietf.ldap.schema

Provides classes that implement the components of the schema on LDAP servers representing Attributes, Objectclasses, and MatchingRules.

2. Package org.ietf.ldap

2.0.1 Description

Provides interfaces and classes that model LDAP directory entries and their components, as well as components of the protocol that apply to both clients and servers. The org.ietf.ldap package provides directory entry, schema, filter, and exception abstractions.

Most of the classes will be familiar to designers that have worked with LDAP before. There are several classes that are intended to be extended.
that model protocol elements that support extension of the protocol:

```
Control
ExtendedResponse
Enum
```

### 2.1 Interface org.ietf.ldap.Controls

```java
public abstract interface Controls
```

Defines a collection of Control objects to be used to transfer ldap controls in requests as well as to receive controls in responses.

#### 2.1.1 size

```java
public int size()
```

Return the number of Control objects in the collection.

#### 2.1.2 get

```java
public Control get(java.lang.String ctrlId)
```

Return the Control with the corresponding control type OID in String form. If no such Control exists then null is returned.

#### 2.1.3 get

```java
public Control get(OID ctrlId)
```

Return the Control with the corresponding control type OID. If no such Control exists then null is returned.

#### 2.1.4 getAll

```java
public java.util.Enumeration getAll()
```

Return an Enumeration of the Control objects in the collection.

#### 2.1.5 toArray

```java
public Control[] toArray()
```

Return an array of the Controls in the collection.

#### 2.1.6 getIds

```java
public java.util.Enumeration getIds()
```

Returns an Enumeration of the control type OIDs for the Controls in the collection.

#### 2.1.7 put

```java
public Control put(Control ctrl)
```
Add a Control to the collection. If there is a Control of the same type in the collection it is replaced by the given Control and the previous Control is returned as the result of the method; otherwise null is returned.

2.1.8 remove

public Control remove(java.lang.String ctrlId)

Removes the Control with the given control type OID represented as a String. The removed Control is returned as the result of the method. If no such Control exists then null is returned and no other action is taken.

2.1.9 remove

public Control remove(OID ctrlId)

Removes the Control with the given control type OID. The removed Control is returned as the result of the method. If no such Control exists then null is returned and no other action is taken.

2.2 Interface org.ietf.ldap.EntityEnumeration

public abstract interface EntityEnumeration
   extends java.util.Enumeration

This interface defines a means of enumerating the entries returned by LDAP operations such as search(...). It extends the Enumeration interface so that Exceptions specific to the LDAP protocol and its interface components may be thrown from the operations: next(), nextEntry() and hasMore().

In some cases an application may perform processing directly on Referrals as well as the Entries that are returned from a search or other operation. In this case the next() method is used. If the nextEntry() method is used and a Referral is next among the objects of the EntityEnumeration then a ReferralException is thrown. The operations of the interface Enumeration are also available so that components that expect an Enumeration may be used with as well.

In some places in the LDAP protocol an exceptional resultCode may be returned, for example, sizeLimitExceeded. In these cases the result entries may be consumed long after the result has been received. In this situation a SizeLimitExceeded Exception will be thrown. If the nextElement () method is used and the next() method would throw an LDAPException then the NoSuchElementException is thrown instead.

2.2.1 hasMore

public boolean hasMore()

Expires 8/99

JAVA LDAP API February 1999

throws LDAPException

Returns true if more entries are available, false otherwise. This operation may signal an LDAPException in the event that all entries have been returned and an exceptional condition was returned from one
or more servers during processing of the search.

Returns: true if more results are available, false otherwise.

2.2.2 nextEntry

public Entry nextEntry() throws LDAPException

Returns the next Entry or throws an LDAPException if all entries have been processed and an exceptional condition occurred during processing of the search.

Returns: the next Entry in the EntityEnumeration.

Throws:
   LDAPException - if an exceptional result was returned during processing.

2.2.3 next

public Entity next() throws LDAPException

Returns the next Entity (Entry or Referral) or throws an LDAPException if all entries have been processed and an exceptional condition occurred during processing of the search. This method is provided for those cases in which both Entry and Referral objects are handled directly by an application. In the case that referral handling is performed implicitly by the interface, nextEntry() is the preferred method to use to access the EntityEnumeration.

Returns: the next Entry in the EntityEnumeration.

Throws:
   LDAPException - if an exceptional result was returned during processing.

2.3 Interface org.ietf.ldap.SocketHandler

public abstract interface SocketHandler

The SocketHandler interface is typically implemented by classes that provide TLS support. The LDAP protocol requires the use of the Start TLS method of establishing a TLS session. This implies that a connection from the client to the server has already been established before it is known and agreed that TLS will be used.

This interface serves to separate the implementation of the Connection mechanism from the details of a given implementation of TLS support.

2.3.1 connectSocket

public java.net.Socket
   connectSocket(java.lang.String host,
               int port,
This method is called from the Connection to establish a Socket over which the client will interact with the LDAP server.

2.4 Class org.ietf.ldap.AliasDeref

class AliasDeref

extends Enum

Enumerates the possible choices for controlling the de-referencing of aliases during search operations: NEVER, SEARCHING, FINDING, and ALWAYS.

2.4.1 Fields

public static final AliasDeref NEVER

   Never follow aliases.

public static final AliasDeref SEARCHING

   Follow aliases only when searching, once the base entry has been located.

public static final AliasDeref FINDING

   Follow aliases when finding the base entry.

public static final AliasDeref ALWAYS

   Always follow aliases.

2.4.2 toAliasDeref

public static AliasDeref toAliasDeref(int code)

Given an integer code for a value of AliasDeref, returns the unique instance of AliasDeref that corresponds to that code or null.

Parameters:
   code - encoded value of an instance of AliasDeref.

Returns: corresponding instance of AliasDeref or null.

2.4.3 toName

public static java.lang.String toName(int code)

Given an integer code returns the name of the corresponding value of AliasDeref

Expires 8/99

JAVA LDAP API February 1999

2.5 Class org.ietf.ldap.AssertionValue
public abstract class AssertionValue

An AssertionValue is used in both AttributeValueAssertions and as the matchValue of a MatchingRuleAssertion (although this latter is not currently reified as a class).

An AssertionValue may be encoded as either a string or a binary encoding represented via a byte[].

The syntax of the AssertionValue is determined by the MatchingRuleAssertion in which it occurs.

2.5.1 make

public static AssertionValue make(byte[] val)

Returns an AssertionValue represented by the byte[].

Parameters:
  val - the byte[] representation

Returns: the AssertionValue

2.5.2 make

public static AssertionValue make(java.lang.String val)

Returns an AssertionValue represented by the String.

Parameters:
  val - the String representation

Returns: the AssertionValue

2.5.3 numBytes

public abstract int numBytes()

The size of the AssertionValue in bytes.

If the value was constructed using a String then the size is reported in terms of the bytes that occur in a UTF-8 encoding of the String value.

Returns: the size of the AssertionValue in bytes

2.5.4 numChars

public abstract int numChars()

The size of the AssertionValue in chars.

If the value was constructed using a byte[] then it is assumed to have been a UTF-8 encoding of a UCS-2 String.

Returns: the size of the AssertionValue in chars
2.5.5 toString

public abstract java.lang.String toString()

Returns the String representation of the AssertionValue.

If the value used to construct the AssertionValue was a byte[] then it is converted to a String via UTF-8. If the byte[] does not encode a String via UTF-8 then null is returned.

Returns: the String representation of this AssertionValue.

2.5.6 toBytes

public abstract byte[] toBytes()

Returns the byte[] representation of the AssertionValue.

If the value used to construct the AssertionValue was a String then it is converted to a byte[] via UTF-8.

Returns: the byte[] representation of this AssertionValue.

2.6 Class org.ietf.ldap.Attribute

public class Attribute

Attribute implements the association of an AttributeType or more generally an AttributeDescription with one or more AttributeValues.

There are a wide variety of constructors available as a convenience to the application designer.

2.6.1 Constructors

public Attribute(AttributeType type)

Constructs an Attribute from an AttributeType. The resulting Attribute has no values.

These may be assigned later via addValue or addValues.

public Attribute(AttributeDescription desc)

Constructs an Attribute from an AttributeDescription. The resulting Attribute has no values.

These may be assigned later via addValue or addValues.

public Attribute(AttributeType type, AttributeValue value)

Constructs an Attribute from an AttributeType and a single AttributeValue.

Additional values may be assigned later via addValue or addValues.
public Attribute(AttributeType type, AttributeValue[] values)

Constructs an Attribute from an AttributeType and an array AttributeValues.

Additional values may be assigned later via addValue or addValues.

public Attribute(AttributeDescription desc, AttributeValue[] values)

Constructs an Attribute from an AttributeDescription and an array AttributeValues.

Additional values may be assigned later via addValue or addValues.

public Attribute(AttributeType type, java.util.Vector v)

Constructs an Attribute from an AttributeType and a Vector of AttributeValues.

Additional values may be assigned later via addValue or addValues.

public Attribute(AttributeDescription desc, java.util.Vector v)

Constructs an Attribute from an AttributeDescription and a Vector of AttributeValues.

Additional values may be assigned later via addValue or addValues.

public Attribute(java.lang.String desc, java.lang.String value)

Constructs an Attribute from an AttributeType and a String representing a single attribute value.

Additional values may be assigned later via addValue or addValues.

public Attribute(java.lang.String desc, byte[] value)

Constructs an Attribute from an AttributeType and a byte[] representing a single attribute value. Additional values may be assigned later via addValue or addValues.

Expires 8/99                                                [Page   9]
JAVA LDAP API                                            February 1999

2.6.2 toString

public java.lang.String toString()

Returns a String listing the values of this Attribute in LDIF format. Useful for debugging or simple applications that need to display the contents of an Attribute.

Returns: a String listing of the values

2.6.3 addValue

public void addValue(java.lang.String val)
Adds a single String value to the Attribute.

Parameters:
   val - the String value to add

2.6.4 addValue

public void addValue(byte[] val)

Adds a single byte[] value to the Attribute.

Parameters:
   val - the byte[] value to add

2.6.5 addValue

public void addValue(AttributeValue val)

Add a single AttributeValue to the Attribute.

Parameters:
   val - the AttributeValue to add

2.6.6 addValues

public void addValues(AttributeValue[] vals)

Merges the array of AttributeValues with the values already associated with this Attribute.

Parameters:
   vals - the array of AttributeValues to merge

2.6.7 size

public int size()

Returns the number of AttributeValues for this Attribute.

Expires 8/99

JAVA LDAP API February 1999

2.6.8 getType

public AttributeType getType()

Returns the AttributeType of this Attribute

2.6.9 getDescription

public AttributeDescription getDescription()

Returns the AttributeDescription of this Attribute

2.6.10 getValues

public AttributeValue[] getValues()

Returns an array of the AttributeValues for this Attribute
2.6.11 getStringValues

public java.lang.String[] getStringValues()

Returns an array of Strings representing the values of this Attribute. The form of each of the strings depends on the syntax of the values for this Attribute. If the value was originally presented as a byte[] then a conversion is performed from the byte[] via UTF-8 to a string value; otherwise the AttributeValue was constructed from a String and no conversion is performed.

Returns: the array of AttributeValues

2.6.12 getByteValues

public byte[][] getByteValues()

Returns a byte[] representing the values of this Attribute.

The form of each byte[] depends on the syntax of the values for this Attribute. if a value was originally presented as a String then it is converted to a byte[] representation of the UTF-8 encoding of the string; otherwise, the original values are simply returned.

Returns: the byte[] value.

2.7 Class org.ietf.ldap.AttributeDescription

public class AttributeDescription
    extends AttributeType

The AttributeDescription extends the AttributeType with any options associated with the coding of values of an Attribute. These descriptions are in accordance with RFC 2251 section 4.1.5. Accessors are provided so that subsets of options of a given description may be formed based on features of the description such as the occurrence of a given prefix, e.g., "lang-en".

2.7.1 Constructors

public AttributeDescription(OID oid)

Constructs an AttributeDescription from the given OID. The description has no name and no options.

Parameters:
    oid - of the new attribute description.

public AttributeDescription(java.lang.String desc)

Constructs an AttributeDescription from a string using the syntax from RFC 2251 section 4.1.5.

Parameters:
    desc - the attribute type and any options.
public AttributeDescription(OID oid, java.lang.String name)

Constructs an AttributeDescription from an (dotted numeric string) and
a name. This constructor essentially builds a AttributeType that
associates the name to the given OID with no options.

Parameters:
  oid - the dotted numeric string for an OID.
  name - of attributes with the type oid.

public AttributeDescription(OID oid, java.lang.String[] options)

Constructs an AttributeDescription from an OID (dotted numeric string)
and a list of options. There is no name associated with the given OID,
so the type is known only via the oid.

Parameters:
  oid - the dotted numeric string for an OID.
  options - list of options for this attribute description,

public AttributeDescription(java.lang.String name,
                          java.lang.String[] options)

Constructs an AttributeDescription from a name and a list of options
that further qualify the attribute type and its value.

Parameters:
  name - of attributes with this type.
  options - list of options for this attribute description

public AttributeDescription(OID oid,
                          java.lang.String name,
                          java.lang.String[] options)

2.7.2 toDescriptions

public static AttributeDescription[] toDescriptions(java.lang.String[] strings)

Generates an array of AttributeDescriptions from a String[].

2.7.3 toStrings

public static java.lang.String[] toStrings(AttributeDescription[] atts)

A utility that generates a list of String descriptions from a list of
AttributeDescriptions.

Parameters:
  atts - a list of AttributeDescriptions

Returns: a list of corresponding string descriptions

2.7.4 toString

public java.lang.String toString()

Returns the full description in string form using the syntax of RFC 2251 section 4.1.5.

2.7.5 getType

public AttributeType getType()

Returns just the type portion of this attributes description.

2.7.6 getOptions

public java.lang.String[] getOptions()

Return a list of the options for this description.

2.7.7 includes

public boolean includes(java.lang.String lang)

Return true if at least one option starts with lang. This method may be used to test whether this description pertains to a given language by providing for example "lang-en". It could also be used to test whether the description describes an attribute value encoded as binary via "binary". The test is case insensitive.

2.7.8 includes

public boolean includes(java.lang.String[] s)

Returns true if every element of the given list of Strings is a prefix of at least one option of the description. The test is case insensitive.

2.7.9 getSubOptions

public java.lang.String[] getSubOptions(java.lang.String lang)

Returns all the options that start with lang. This essentially captures the idea that lang subsumes all of the returned options. For example, "lang-en" subsumes both "lang-en-US" and "lang-EN-gb".

2.7.10 isSubtype

public boolean isSubtype(AttributeDescription other)
Compares the options of this with the options of other and returns true if the options of other are a strict subset of the options of this.

Parameters:
other - is the candidate supertype of this.

2.7.11 isSubtypeEq

public boolean isSubtypeEq(AttributeDescription other)

Compares the options of this with the options of other and returns true if the options of other are a (possibly equal) subset of the this.

Parameters:
other - is the candidate super type of this.

Returns: true if other’s options are a subset of this.options.

2.7.12 addOption

public void addOption(java.lang.String option)

Adds a new option to the description ensuring that the options are in sorted order. This is the order that will be returned by a toString() or getOptions().

Parameters:
option - to be added to the description.

2.7.13 addOptions

Expires 8/99

JAVA LDAP API February 1999

public void addOptions(java.lang.String[] opts)

Adds a list of options to the description ensuring that the options are in sorted order. This is the order that will be returned by a toString() or getOptions().

Parameters:
opts - array of options to be added to the description.

2.8 Class org.ietf.ldap.AttributeSet

public class AttributeSet
    implements java.lang.Cloneable

An AttributeSet is simply a collection of Attributes. It provides a convenient way of manipulating a set of Attributes to be sent in a message or retrieved perhaps from an Entry. One notable use is to collect together all the Attributes from an Entry that have a common AttributeType but possibly differing AttributeDescriptions, such as different languages. This class provides methods for extracting a subset of a set that have a common type or description and supports the language subtyping rules in draft-ietf-ldapext-lang-01.txt.

2.8.1 Constructors
public AttributeSet()

public AttributeSet(Attribute[] a)

2.8.2 toString

public java.lang.String toString()

Returns a String representation conforming to LDIF for the elements of
the set.

2.8.3 elementAt

public Attribute elementAt(int i)

2.8.4 elements

public java.util.Enumeration elements()

Enumerates the attributes in the set. That is, nextElement() will
return instances of Attribute.

2.8.5 size

public int size()

Returns: the number of Attributes in the set.

2.8.6 addElement

Expires 8/99 [Page 15]

JAVA LDAP API February 1999

public AttributeSet addElement(Attribute elem)

Adds an Attribute to the set. No check is performed to verify that the
added element is unique.

Parameters:
   elem - the Attribute to add to the set.

2.8.7 getAttributes

public AttributeSet getAttributes(AttributeType type)

There can be many Attributes with the same AttributeType but different
options hence different AttributeDescriptions. This method returns all
of these Attributes.

2.8.8 getAttributes

public AttributeSet getAttributes(AttributeDescription desc)

Returns the subset of attributes that have at least as much detail as
desc.

2.8.9 getAttributes

public AttributeSet getAttributes(AttributeType type,
getAttributes

public AttributeSet getAttributes(AttributeDescription desc,
java.lang.String optionPrefix)

Returns the subset of attributes that have at least as much detail as
desc and an option at least as specific as optionPrefix. While intended
for "lang-" options any options with a common prefix will do, e.g. "x-
image", etc.

2.8.10 getAttributes

public AttributeSet getAttributes(AttributeDescription desc,
java.lang.String optionPrefix)

Returns the subset of attributes with the given type and an option at
least as specific as optionPrefix. While intended for "lang-" options
any options with a common prefix will do, e.g. "x-image", etc.

2.8.11 getAttributes

public AttributeSet getAttributes(java.lang.String optionPrefix)

Returns the subset of attributes that have an option at least as
specific as optionPrefix. While motivated by "lang-" options, any
options with a common prefix will do, e.g. "x-image", etc.

2.9 Class org.ietf.ldap.AttributeType

Subclasses:

Expires 8/99

JAVA LDAP API

AttributeDescription

public class AttributeType
   extends SchemaElementId

The AttributeType represents the OID and name(s) by which a given
Attribute is known. If both a name and OID are present in an instance
of Attribute then the instance can considered to assign the name to the
OID.

2.9.1 Constructors

public AttributeType(OID oid)

public AttributeType(java.lang.String name)

public AttributeType(OID oid, java.lang.String name)

2.9.2 equals

public boolean equals(AttributeDescription desc)

Tests whether given AttributeDescription is of this AttributeType.
Returns: whether the description is of this type

2.9.3 toDescription

public AttributeDescription toDescription()
Converts an AttributeType to an AttributeDescription.

2.10 Class org.ietf.ldap.AttributeValueAssertion

public class AttributeValueAssertion
        extends java.lang.Object

The AttributeValueAssertion implements an assertion about the value of
an attribute of the given description as specified in RFC 2251.

If the "binary" option is present in attribute description, this
signals to the server that the assertion is a binary encoding of the
assertion value. See ByteArrayAssertion.

For all the string-valued user attributes described in RFC 2252, the
assertion value syntax is the same as the value syntax. Clients may use
attribute values as assertion values in compare requests and search
filters.

Note however that the assertion syntax may be different from the value
syntax for other attributes or for non-equality matching rules. These
may have an assertion syntax which contains only part of the value. See
section 20.2.1.8 of X.501 for examples.

2.10.1 Constructors

public AttributeValueAssertion(AttributeDescription desc,
                               AssertionValue assertion)

public AttributeValueAssertion(java.lang.String desc,
                                java.lang.String assertion)

public AttributeValueAssertion(java.lang.String desc,
                                byte[] assertion)

2.10.2 toString

public java.lang.String toString()

2.10.3 getType

public AttributeType getType()

Returns the AttributeType of this assertion.

2.10.4 getDescription

public AttributeDescription getDescription()

Returns the AttributeDescription of this assertion.

2.10.5 getAssertion

public AssertionValue getAssertion()
Returns the value asserted as an AssertionValue.

2.10.6 getStringAssertion

public java.lang.String getStringAssertion()

Returns a String representing the asserted value.

The form of the String depends on the syntax of the asserted value for this AttributeValueAssertion. If the value was originally presented as a byte[] then it is converted to a String via a UTF-8 decoding of the string if possible; otherwise, the original value are simply returned.

2.10.7 getByteArrayAssertion

public byte[] getByteArrayAssertion()

Returns a byte[] representing the asserted value.

The form of the byte[] depends on the syntax of the asserted value for this AttributeValueAssertion. If the value was originally presented as a String then it is converted to a byte[] representation of the UTF-8 encoding of the string; otherwise, the original value are simply returned.

2.11 Class org.ietf.ldap.AttributeValue

public abstract class AttributeValue
extends java.lang.Object

The AttributeValue encapsulates the different possible representations such as String and byte[], for an attribute value.

2.11.1 make

public static AttributeValue make(byte[] val)

Returns an AttributeValue represented by the byte[].

Parameters:
val - the byte[] representation

2.11.2 make

public static AttributeValue make(java.lang.String val)

Returns an AttributeValue represented by the String.

Parameters:
val - the String representation

2.11.3 size

public abstract int size()

2.11.4 toString
public abstract java.lang.String toString()

Returns a String representation of the AttributeValue.

If the value used to construct the AttributeValue was a byte[] then it is converted to a String via UTF-8. If the byte[] does not encode a String via UTF-8 then null is returned.

Returns: the String representation of this AttributeValue.

2.11.5 isBinary

public abstract boolean isBinary()

Returns true if the AttributeValue must be encoded as binary in an LDIF representation.

2.11.6 toLDIFString

public abstract java.lang.String toLDIFString()

2.11.7 toBytes

public abstract byte[] toBytes()

Returns the byte[] representation of the AttributeValue.

If the value used to construct the AttributeValue was a String then it is converted to a byte[] via UTF-8.

2.12 Class org.ietf.ldap.BasicControls

public class BasicControls
    extends java.lang.Object
    implements Controls

Provides the default implementation of the Controls interface. It is a collection of individual Control objects.

2.12.1 Constructors

public BasicControls()

public BasicControls(Control[] ca)

Constructs an instance from the given array of Control objects. Primarily used when receiving a collection of Control objects in a message.

public BasicControls(java.util.Vector cv)

2.12.2 size

public int size()
Return the number of Control objects in the collection

2.12.3 get

public Control get(java.lang.String ctrlId)

Return the Control with the corresponding control type OID in String form.

2.12.4 get

public Control get(OID ctrlId)

Return the Control with the corresponding control type OID

2.12.5 getAll

public java.util.Enumeration getAll()

Returns an Enumeration of the control type OIDs for the Controls in the collection.

2.12.6 toArray

public Control[] toArray()

Return an array of the Controls in the collection.

2.12.7 getIds

public java.util.Enumeration getIds()

Returns an Enumeration of the Control objects in the collection.

2.12.8 put

public Control put(Control ctrl)

Add a Control to the collection. If there is a Control of the same type in the collection it is replaced by the given Control and the previous Control is returned as the result of the method; otherwise null is returned.

2.12.9 remove

public Control remove(java.lang.String ctrlId)

Removes the Control with the given control type OID represented as a String. The removed Control is returned as the result of the method.

2.12.10 remove

public Control remove(OID ctrlId)

Removes the Control with the given control type OID represented as a String. The removed Control is returned as the result of the method.
2.13 Class org.ietf.ldap.Continuation

public class Continuation
    extends URLList

A Continuation is a list of urls to other servers that should be contacted to continue a search. A Continuation is ONLY returned via an LDAP SearchResultReference, and is otherwise, the same as a URLList. It is useful when an application is performing referral handling on its own. This distinguishes the case of a SearchResultDone with a referral that indicates that the contacted server was not able to locate the base entry of the search.

2.13.1 Constructors

public Continuation(java.util.Vector urls)

public Continuation(java.lang.String[] s)

2.13.2 toString

public java.lang.String toString()

Returns the String representation of the first url on the list of urls.

2.14 Class org.ietf.ldap.Control

Subclasses:
    AttributeSizeLimitControl, ChainServerControl, ManageDsaITControl,
    MatchedValuesOnlyControl, NoChainingControl, NoCopyControl,
    SimplePagedControl, SortRequestControl, SortResponseControl,
    TriggerControl

public abstract class Control
    extends java.lang.Object

Control is an abstract class that is the base class for all the LDAP controls that can be sent to an LDAP server or received in a response from an LDAP server. A new controlType is implemented by extending Control with the fields that represent the controlValue for the Control, defining the static NAME and OID for the controlType, registering the Control, and defining the methods:

    controlValue
    controlValueString

and any accessors appropriate to the controlValue components of the Control.

2.14.1 Constructors

public Control(java.lang.String name, OID oid, boolean criticality)

Constructs a Control with no controlValue for the specified OID with the given criticality.
Parameters:
oid - the type of the control
criticality - true = operation should be discarded if server does not support this control

public Control(java.lang.String name, OID oid)

Constructs a non-critical ControlSeq with no controlValue for the specified OID.

Parameters:
oid - the id of the control

2.14.2 fromAsn

Expires 8/99

JAVA LDAP API February 1999

public static Control
fromAsn(java.lang.String controlType,
boolean criticality,
org.ietf.ldap.asn.AsnOctets controlValue)

fromAsn uses the oid in the received control to locate the Class for corresponding controlType. The Class is instantiated via newInstance() then requested via its init method to initialize the instance with the received criticality and controlValue.

2.14.3 init

public void init(boolean crit,
org.ietf.ldap.asn.AsnOctets cv)

Performs initialization of the criticality and controlValue fields of an instance built from ASN.1. This method typically needs to be overridden in the implementation of specific controls since the specific control is the locus of the information about how the control value is encoded.

Parameters:
criticality - the criticality encoded in the message
controlValue - the OCTET STRING encoding of the controlValue

2.14.4 toAsn

public org.ietf.ldap.apdu.ControlSeq toAsn()

This method converts any Control to its corresponding ASN.1 SEQUENCE for writing on the wire. This method will typically be overridden in each specific Control's class to implement the control specific encoding of the controlValue for the specific control.

This method is not intended to be called by users, rather it is called from the machinery that is responsible for de-serializing objects from the BER.

Returns: the ControlSeq PDU component.
2.14.5 getControlName

public java.lang.String getControlName()

Return a local name (if any) for this type of Control.

2.14.6 getControlType

public OID getControlType()

Return the OID by which this type of Control is known to both clients and servers.

2.14.7 sameType

public boolean sameType(Control c)

Returns true if the given Control has the same type as this Control.

Parameters:
   c - the Control to test for type equality

2.14.8 getCriticality

public boolean getCriticality()

Return the current criticality for this Control.

2.14.9 toString

public java.lang.String toString()

Generate a String representation for this Control.

2.15 Class org.ietf.ldap.CramMD5SaslCredentials

class CramMD5SaslCredentials
    extends SaslCredentials

CramMD5SaslCredentials implements the use of CRAM-MD5 in the simple authentication and security layer. The credentials are computed in accordance with RFCs 2095 and 2104.

2.15.1 Constructors

public CramMD5SaslCredentials()

Creates an instance that has null credentials which translates to absent credentials over protocol. An instance of this type is used to signal the start (or re-start) of a Sasl bind using the CRAM-MD5 mechanism.

public CramMD5SaslCredentials(byte[] authzId,
                               java.lang.String password,
                               byte[] challenge)
Creates credentials initialized from the given authorization id, password, and challenge. The credentials are computed in accordance with RFCs 2095 and 2104.

Parameters:
- authzId - authorization id, may be null
- password - the secret information shared between the user and the server
- challenge - from the server

```java
public CramMD5SaslCredentials(byte[] authzId,
                                        byte[] password,
                                        byte[] challenge)
```

2.15.2 `getMethod`

```java
public java.lang.String getMethod()
```

Returns the SASL method name "CRAM-MD5"

2.16 Class `org.ietf.ldap.DNAttributeTypeAndValue`

```java
public class DNAttributeTypeAndValue
    extends java.lang.Object
```

Instances of this class represent an attribute type and corresponding value as a component of an RDN.

2.16.1 Constructors

```java
public DNAttributeTypeAndValue(java.lang.String type,
                                        java.lang.String val)
```

Construct an `attributeTypeAndValue` as one of possibly several values of an RDN from a given type and value. The value is assumed to not be escaped and escape processing will be applied.

Parameters:
- type - of the attribute
- val - the unescaped value of the attribute.

```java
public DNAttributeTypeAndValue(java.lang.String type,
                                        byte[] val)
```

Construct an `attributeTypeAndValue` as one of possibly several values of an RDN from a given type and value. The value is assumed to not be escaped.
escaped and escape processing will be applied.

Parameters:
  type - of the attribute
  val - the unescaped value of the attribute.

2.16.2 getType

public AttributeType getType()

Expires 8/99

JAVA LDAP API

Returns the AttributeType of this RDN component.

2.16.3 getValue

public AttributeValue getValue()

Returns the unescaped form of the value for this attributeTypeAndValue

2.16.4 toAttribute

public Attribute toAttribute()

Returns an Attribute consisting of the given AttributeType and the single given value.

2.16.5 toString

public java.lang.String toString()

Returns the attributeTypeAndValue as a String conforming to RFC 2253.
The return value is suitable for use in the string representation of a distinguished name.

2.17 Class org.ietf.ldap.DN

public class DN
  extends java.lang.Object

This class manages the construction of distinguished names and access to their component parts. The syntax for a distinguished name may be found in RFC 2253 section 3, which discusses the UTF-8 string representation of distinguished names. A distinguished name is a sequence of relative distinguished names (RDN) each of which expresses one or more attribute type and value equalities, e.g. ("cn=Bill Jones"). An attribute type may also be presented as an OID (a dotted numeric string, e.g. 1.3.6.1.4.1.1466.0).

2.17.1 Constructors

public DN()

Constructs an empty DN

public DN(java.lang.String dn)

Constructs a DN representation of the distinguished name written in the
parameter String. The string is presumed to conform to RFC 2253 section 3.

2.17.2 getLeaf

public RDN getLeaf()

Expires 8/99 [Page 26]

JAVA LDAP API February 1999

Returns the leftmost RDN of this DN. If this is the root DN then an empty RDN is returned.

2.17.3 getParent

public DN getParent()

Returns the parent DN for this DN. If this is the root DN then the current instance is simply returned.

2.17.4 getRDNs

public RDN[] getRDNs()

Returns the list of RDNs for this DN.

2.17.5 toString

public java.lang.String toString()

Return a standard String representation for this distinguished name.

2.17.6 toURLstring

public java.lang.String toURLstring()

Return a URL encoded String representation for this distinguished name.

2.18 Class org.ietf.ldap.Entity

Subclasses:
  Entry, URLList

public class Entity
  extends java.lang.Object

An Entity is either an Entry or a URLList. See EntityEnumeration and SearchResults. Essentially, an application that is performing referral following will expect an Entity as a search result. An application that relies on the SearchResults to perform referral following will expect only an Entry as a search result.

2.18.1 Constructors

public Entity()

2.19 Class org.ietf.ldap.Entry

Subclasses:
  RootDSEntry
public class Entry
    extends Entity

Expires 8/99

JAVA LDAP API
February 1999

An Entry models the basic unit of an LDAP directory.

2.19.1 Constructors

public Entry()

public Entry(DN dn, AttributeSet attrs)

2.19.2 toString

public java.lang.String toString()

Returns a String representation of the Entry in accordance with LDIF syntax.

2.19.3 getDN

public DN getDN()

Returns the distinguished name of the Entry.

2.19.4 getAttributes

public AttributeSet getAttributes()

Returns an AttributeSet of all of the Attributes in the Entry.

2.19.5 getAttributes

public AttributeSet getAttributes(AttributeType type)

There can be many Attributes with the same AttributeType but different options hence different AttributeDescriptions. This method returns all of these Attributes.

2.19.6 getAttributes

public AttributeSet getAttributes(AttributeDescription desc)

Return the subset of attributes that are subtypes of desc

2.19.7 getAttributes

public AttributeSet getAttributes(AttributeType type, java.lang.String lang)

Return the subset of attributes of type, type and with at least one option in the description that has a prefix of lang. The test is case insensitive.

2.19.8 getAttributes
public AttributeSet
  getAttributes(AttributeDescription desc, java.lang.String lang)

Expires 8/99

JAVA LDAP API

Return the subset of attributes that are subtypes of desc with at least
one option in the description that has a prefix of lang. The test is
case insensitive.

2.19.9 getAttributes

public AttributeSet getAttributes(java.lang.String lang)

Return the subset of attributes that have at least one option with lang
as a case insensitive prefix.

2.20 Class org.ietf.ldap.Enum

Subclasses:
  AliasDeref, ModifyOp, Scope, SortResult

public abstract class Enum
  extends java.lang.Object

Enum is the base class for all enumerated types. An enumerated type has
a fixed set of values. Each value has a (meaningful) name and a code
that is used in the ASN.1 representation for the value. Enumerated
types are used in the api to support well typed use of LDAP. They are
similar to the C API’s constant definitions with the additional benefit
that they are type checked where they occur. An enumerated type is
defined by extending Enum and defining a private constructor that is
invoked from within static constructors for each of the instances of
the type. For example:

    public class Foo extends Enum {
        public static Foo bar = new Foo("bar", 1);
        public static Foo baz = new Foo("baz", 2);
        private Foo(String name, int code) {
            this.name = name;
            this.code = code;
        }
    }

defines an enumerated type Foo with values Foo.bar and Foo.baz.

2.20.1 toString

public java.lang.String toString()

Returns the name of the value.

2.20.2 toCode

public int toCode()

Returns the ASN.1 code assigned to the value.

2.21 Class org.ietf.ldap.ExternalSaslCredentials
public class ExternalSaslCredentials
    extends SaslCredentials

ExternalSaslCredentials are used to convey credentials between the SDK and underlying SSL or TLS security layers.

2.21.1 Constructors

public ExternalSaslCredentials(java.lang.String authzId)

2.21.2 getMethod

public java.lang.String getMethod()

    Returns the SASL method String.

2.22 Class org.ietf.ldap.Filter

public class Filter
    extends java.lang.Object

Filter encapsulates the RFC 2254 compliant string expression for a filter and the internal FilterChoice structure used to write the ASN.1 encoding for the filter.

2.22.1 Constructors

public Filter(java.lang.String filterExpr)
    throws ParseException

Constructs a Filter with a string representing a filter as defined in RFC 2254. The representing field filterExpr is constructed from the FilterChoice object, which is in interpreted from the input filterExpr.

Parameters:
    filterExpr - string which to parse and construct itself.

Throws:
    ParseException - If there is an error parsing the filter expression string.

public Filter(org.ietf.ldap.apdu.FilterChoice fc)

Constructs a Filter from a FilterChoice object.

Parameters:
    filterChoice - FilterChoice object from which to construct itself.

2.22.2 toString

public java.lang.String toString()

Returns a String representation of itself
2.22.3 toURLstring

public java.lang.String toURLstring()

Returns a String representation of itself that is URL "safe".

2.22.4 internal

public org.ietf.ldap.apdu.FilterChoice internal()

Returns a FilterChoice representation of itself.

2.23 Class org.ietf.ldap.Interaction

public class Interaction
    extends java.lang.Object

An Interaction models the flow of messages for a single client/server interaction such as a modify or search. A request to getMessage will block the caller until there is a message available and request to putMessage places a message in the interaction. The Interaction supports completely asynchronous interaction between the using code and the connection(s) that support the interaction with the other end-point.

2.23.1 Constructors

public Interaction(org.ietf.ldap.apdu.Connection con, int msgId)

An Interaction is created by Connection or a subclass to represent the interaction between one or more connections and a user (client or server) for a single request/response instance, e.g., a modify or a search interaction between client and server. Due to referral handling there may be several connections involved with a single Interaction.

2.23.2 expect

public void expect(int messageId)

Informs the Interaction to expect messages with sequence number messageId. This is used to provide bookkeeping for the outstanding messages on an Interaction.

2.23.3 close

public void close(int messageId)

Mark this Interaction as closed. I.e., there should be no more messages put into this Interaction. An Interaction may be closed because the last message in the interaction has been received or because the connection(s) over which the Interaction is occurring has been closed. This method is intended to be called by Connection or a subclass.

Parameters:
messageId - no longer expecting messages for this id - final response received

2.23.4 isClosed

public boolean isClosed()

Returns whether the interaction has been closed or not. An Interaction is closed when either the consumer is finished using it or when the connection(s) with which it is associated are closed.

2.23.5 getMessageId

public int getMessageId()

Returns the most recent messageId that was signalled via an expect().

2.23.6 getMessage

public Message getMessage()

Get the next message in an interaction. The caller is blocked until either there is a message available or until all connections on which the interaction is participating have been closed.

2.23.7 putMessage

public void putMessage(Message message)

Puts a message into the interaction. Any threads waiting on the interaction are notified. It is erroneous to attempt to put a message into an interaction that is already closed. This method is intended to be called by Connection or a subclass.

Parameters:
  message - the message to add to the interaction

2.24 Class org.ietf.ldap.LDAPURL

public class LDAPURL
  extends java.lang.Object

LDAPURL implements LDAP URLs as defined in RFC 2255. In addition to containing the host and port information, LDAP URL’s allow search parameters to be encoded with the URL. The question mark (?) character is reserved as a delimiter for the search parameters. Other "unsafe" characters (as described in RFC 1738 section 2.2) must be encoded using the percent sign (%) escaping mechanism.

2.24.1 Constructors

public LDAPURL()

Constructs an empty, default URL

Expires 8/99                                                [Page 32]

JAVA LDAP API                                            February 1999
public LDAPURL(java.lang.String url_str)

throws ParseException

Constructs an instance from a String representation of an LDAP URL. The String must conform to RFC 2255.

Parameters:
   url_str - the String representing an LDAP URL

2.24.2 toString

public java.lang.String toString()

Returns: a String representation of itself that is URL "safe" according to RFC 2255

2.24.3 getScheme

public java.lang.String getScheme()

Returns: String representing the scheme of this URL, typically "ldap".

2.24.4 setHost

public void setHost(java.lang.String host)

Sets the host name to which this URL refers.

Parameters:
   host - String representation of a host name

2.24.5 getHost

public java.lang.String getHost()

Returns: String representation the host name to which this URL refers. Local host name is default.

2.24.6 setPort

public void setPort(int port)

Sets the port number to which this URL refers.

Parameters:
   port - int representation of a port number

2.24.7 getPort

public int getPort()

Returns: int representation of the port number to which this URL refers. LDAP port 389 is default.
public void setDN(DN dn)

Sets the DN parameter of the search part of this LDAP URL.

Parameters:
   dn - DN of base object in search

2.24.9 getDN

public DN getDN()

Returns: DN of base object in search part of this LDAP URL.

2.24.10 setAttributes

public void setAttributes(AttributeDescription[] attributes)

Sets the attributes parameter of the search part of this LDAP URL.

Parameters:
   attributes - array of AttributeDescription objects which to return in
                the specified search

2.24.11 getAttributes

public AttributeDescription[] getAttributes()

Returns: array of AttributeDescription objects which are to be returned
         in the search specified in the LDAP URL.

2.24.12 setScope

public void setScope(Scope scope)

Sets the scope parameter of the search part of this LDAP URL.

Parameters:
   scope - Scope of the search specified

2.24.13 getScope

public Scope getScope()

Returns: Scope of the search specified in this LDAP URL

2.24.14 setFilter

public void setFilter(Filter filter)

Sets the filter parameter of the search part of this LDAP URL.

Parameters:
   filter - Filter of the search specified

2.24.15 getFilter
public Filter getFilter()

Returns: Filter of the search specified in this LDAP URL

2.24.16 setExtensions

public void setExtensions(LDAPURLExtension[] extensions)

Sets the extensions part of this LDAP URL.

Parameters:
  extensions - array of LDAPURLExtension objects associated with this
  LDAP URL

2.24.17 getExtensions

public LDAPURLExtension[] getExtensions()

Returns: array of LDAPURLExtension objects specified by this LDAP URL

2.24.18 isSafe

public static boolean isSafe(char c)

Tests a character to determine whether or not it is "safe" according to
RFC 2255.

Parameters:
  c - char to test

Returns: true if c is a "safe" character; false otherwise

2.24.19 isSafe

public static boolean isSafe(java.lang.String s)

Tests a String to determine whether or not it is "safe". A String is
"safe" if it contains no "unsafe" characters.

Parameters:
  s - String to test

Returns: true if the s contains no "unsafe" characters; false otherwise

2.25 Class org.ietf.ldap.LDAPURLExtension

public class LDAPURLExtension
    extends java.lang.Object

Expires 8/99

JAVA LDAP API

February 1999

LDAPURLExtension implements the extension construct for LDAP URLs as
defined in RFC 2255. This construct provides the LDAP URL with an
extensibility mechanism, allowing the capabilities of the URL to be
extended in the future. Extensions are a simple comma-separated list of
type=value pairs, where the =value portion MAY be omitted for options
not requiring it.
Each type=value pair is a separate extension. These LDAP URL extensions are not necessarily related to any of the LDAPv3 extension mechanisms. Extensions may be supported or unsupported by the client resolving the URL. An extension prefixed with a ‘!’ character (ASCII 33) is critical. An extension not prefixed with a ‘!’ character is non-critical. If an extension is supported by the client, the client MUST obey the extension if the extension is critical. The client SHOULD obey supported extensions that are non-critical. If an extension is unsupported by the client, the client MUST NOT process the URL if the extension is critical. If an unsupported extension is non-critical, the client MUST ignore the extension. If a critical extension cannot be processed successfully by the client, the client MUST NOT process the URL. If a non-critical extension cannot be processed successfully by the client, the client SHOULD ignore the extension. Extension types prefixed by "X-" or "x-" are reserved for use in bilateral agreements between communicating parties. Other extension types MUST be defined in this document, or in other standards-track documents.

2.25.1 Constructors

```java
public LDAPURLExtension(java.lang.String extension_string)

Constructs an instance from the specified parameters.

Parameters:
  extension_string - String object assumed to consist of an oid, and optionally an equal sign followed by a value, prepended by an exclamation point if this extension is critical.
```

```java
public LDAPURLExtension(OID type, java.lang.String value, boolean critical)

Constructs an instance from specified parameters.

Parameters:
  type - oid of the extension
  value - String representing the extension value
  critical - true if this extension is critical
```

2.25.2 toURLstring

```java
public java.lang.String toURLstring()

Returns: a String representation of itself which it LDAP URL "safe" as defined in RFC 2255
```

2.25.3 setDescription

```java
public void setDescription(OID type)

Sets the OID of this extension

Parameters:
  type - OID of the attribute
```
2.25.4 getDescription

public OID getDescription()

Returns: OID or this extension

2.25.5 setValue

public void setValue(java.lang.String value)

Sets the value of this extension

Parameters:
   value - value of the extension

2.25.6 getValue

public java.lang.String getValue()

Returns: String representation of the value of this extension

2.25.7 setCriticality

public void setCriticality(boolean critical)

Sets the criticality of this extension

Parameters:
   critical - true if this extension is critical

2.25.8 getCriticality

public boolean getCriticality()

Returns: true if this extension is critical

2.26 Class org.ietf.ldap.MatchingRuleId

public class MatchingRuleId
   extends SchemaElementId

The MatchingRuleId represents the OID and/or name by which a given MatchingRule is known. If both a name and OID are present in an instance of MatchingRuleId then the instance can considered to assign the name to the OID in the context of a specific type of server.

Expires 8/99

JAVA LDAP API February 1999

2.26.1 Constructors

public MatchingRuleId(OID oid)

public MatchingRuleId(java.lang.String name)

public MatchingRuleId(OID oid, java.lang.String name)

2.27 Class org.ietf.ldap.Message
Subclasses:
  Response, SearchEntry, SearchReference

public class Message
  extends java.lang.Object

  A Message is either an LDAP request or a result. It has a messageId and optional Controls. Any other fields are defined in subclasses according to the type from RFC 2251. Messages are defined completely in terms of types internal to the api and are independent of the coding in ASN.1. This allows a clean separation between the functions of encoding/decoding ASN.1 and the application oriented functions of the abstract LDAP Message types.

  2.27.1 Constructors

  public Message()

  2.27.2 setMessageId

  public void setMessageId(int messageId)

  Sets the messageId. This is typically used in the implementation of connection objects and should not be of interest to the api user.

  Parameters:
    messageId - the message sequence number for this message

  2.27.3 getMessageId

  public int getMessageId()

  Returns the message sequence number for this message.

  2.27.4 setControls

  public void setControls(Control[] ca)

  Sets the controls of this message. Normally only used in the implementation of connection objects. This method appears here due to the order in which ASN.1 objects are decoded and the desire to construct message specific classes on-the-fly.

  Parameters:
    controls - the Control[] for this message

  2.27.5 getControls

  public Controls getControls()

  Returns the Controls for this message or null if they are not present.

  2.28 Class org.ietf.ldap.Modification

  public class Modification
    extends java.lang.Object
A Modification is an operation, e.g. ADD or DELETE, and an Attribute that will be added or deleted from some entry.

### 2.28.1 Constructors

```java
public Modification(ModifyOp op, Attribute attribute)
```

Constructs a Modification from the given operation and attribute.

**Parameters:**
- `op` - the operation
- `attribute` - the attribute

### 2.28.2 setToModifications

```java
public static Modification[] setToModifications(ModifyOp op, AttributeSet set)
```

Accepts an operation and a set of Attributes and returns an array of Modifications formed by associating the operation with each of the attributes in the set.

**Parameters:**
- `op` - the operation
- `AttributeSet set` - the set of attributes

**Returns:** the array of Modifications

### 2.28.3 getOp

```java
public ModifyOp getOp()
```

Return the operation of the Modification.

### 2.28.4 setOp

```java
public void setOp(ModifyOp op)
```

Sets the operation to be performed.

### 2.28.5 getAttribute

```java
public Attribute getAttribute()
```

Return the attribute to be modified.

### 2.29 Class org.ietf.ldap.ModifyOp

```java
public final class ModifyOp extends Enum
```

ModifyOp is an enumerated type consisting of three constant values:
ADD, DELETE, and REPLACE that correspond to the operations that may be requested of an LDAP directory on an attribute of an entry.

2.29.1 Fields

public static final ModifyOp ADD
public static final ModifyOp DELETE
public static final ModifyOp REPLACE

2.29.2 toModifyOp

public static ModifyOp toModifyOp(int code)

Used by ASN.1 decode routines to obtain the Scope constant that corresponds to an ASN.1 value.

Parameters:
   code - the ASN.1 value to map to a ModifyOp

Returns: the ModifyOp corresponding to code or null

2.29.3 toName

public static java.lang.String toName(int code)

2.30 Class org.ietf.ldap.NullSaslCredentials

public class NullSaslCredentials
   extends SaslCredentials

NullSaslCredentials are used to abort a Sasl bind in progress. The application may then initiate a new Sasl bind sequence with the same or different mechanism.

2.30.1 Constructors

public NullSaslCredentials()

2.30.2 getMethod

public java.lang.String getMethod()

Returns the SASL method name "NULL".

2.31 Class org.ietf.ldap.OID

public class OID
   extends java.lang.Object

The OID represents the globally unique object identifier for some LDAP element such as an objectClass or attributeType.

2.31.1 Constructors
public OID(java.lang.String oid)

2.31.2 equals

public boolean equals(OID other)

Returns true if both OIDs are the same, false otherwise.

2.31.3 equals

public boolean equals(java.lang.String other)

Returns true if the given string represents the same OID.

2.31.4 toString

public java.lang.String toString()

Returns the conventional dotted numeric string representation of the OID.

2.32 Class org.ietf.ldap.RDN

public class RDN
  extends java.lang.Object

RDN represents a relative distinguished name as specified in RFC 2253 where its syntax is given as a name-component. The components of an RDN may be obtained via toAttributes() which will return a list of Attributes representing each of the attributeTypeAndValues that comprise the RDN.

2.32.1 Constructors

public RDN(java.lang.String spec)

Constructs an RDN from a String, spec,.

Parameters:
  spec - the String representation of the RDN

2.32.2 getATAVs

public DNAttributeTypeAndValue[] getATAVs()

Returns an array of the attribute type and value objects that comprise the components of this RDN

2.32.3 toAttributes

public Attribute[] toAttributes()

Returns a list of Attributes that represent the attributeTypeAndValues of this RDN.

2.32.4 getTypeAndValues
public java.lang.String[] getTypeAndValues()

Returns a list of Strings of the form ".=".

2.32.5 toString

public java.lang.String toString()

Returns a String representing this RDN according to the syntax in RFC 2253 section 3.

2.33 Class org.ietf.ldap.Referral

public class Referral
    extends URLList

A Referral is a list of urls to other servers that should be contacted to perform some operation, such as a search. A Referral is returned in via an LDAP Result with a result code of Referral.

2.33.1 Constructors

public Referral(java.util.Vector urls)
public Referral(java.lang.String[] s)

2.33.2 toString

public java.lang.String toString()

Returns a string representation of the list of URLs suitable for use in LDIF.

2.34 Class org.ietf.ldap.RootDSEntry

Expires 8/99

JAVA LDAP API February 1999

public class RootDSEntry
    extends Entry

RootDSEntry subclasses Entry for accessing server information contained in the root DS entry of a directory.

2.34.1 Constructors

public RootDSEntry(java.lang.String host, int port)
    throws LDAPException

Constructs an instance from a host and port of an LDAP server from which to retrieve the root DS entry.

Parameters:
    host - String representing the host name of an LDAP server
    port - int representing the port number of an LDAP server

public RootDSEntry(ClientConnection connection)
    throws LDAPException
Constructs an instance from a connection already established to an LDAP server from which to retrieve the root DS entry.

Parameters:
  connection - ClientConnection object representing a communication medium to an LDAP server

public RootDSEntry(DirectoryClient client) throws LDAPException

Constructs a RootDSEntry instance from the given client.

2.34.2 getDSEType

public java.lang.String getDSEType()

Returns: String representing the type of the DSE, e.g. "(root)".

2.34.3 getNamingContexts

public DN[] getNamingContexts()

Returns: an array of DNs representing naming contexts held in the server. Naming contexts are defined in section 17 of X.501

2.34.4 getServerName

public DN getServerName()

Returns: a DN representing the name of the server

2.34.5 getSupportedVersions

Expires 8/99

public int[] getSupportedVersions()

Returns: an array of ints representing LDAP versions implemented by the server

2.34.6 getAccessControlScheme

public OID getAccessControlScheme()

Returns: an OID representing the control which dictates access rights for users of this directory

2.34.7getCurrentTime

public java.lang.String getCurrentTime()

Returns: a String representing the current time kept by the server

2.34.8 getSubSchemaEntry

public DN getSubSchemaEntry()
Returns: a DN representing the distinguished name of the subschema entry (or subentry) which controls the schema for this entry.

2.34.9 getSupportedSaslMechanisms

public java.lang.String[] getSupportedSaslMechanisms()

Returns: an array of Strings representing a list of supported SASL security features

2.34.10 getSupportedControls

public OID[] getSupportedControls()

Returns: an array of OIDs representing a list of controls supported by the server

2.34.11 getChangeLog

public DN getChangeLog()

Returns: a DN representing the distinguished name of the changelog directory tree.

2.34.12 getOgSupportedProfile

public OID[] getOgSupportedProfile()

Returns: an array of OIDs for the supported profiles.

2.35 Class org.ietf.ldap.SaslCredentials

Expires 8/99

JAVA LDAP API February 1999

Subclasses:
    CramMD5SaslCredentials, ExternalSaslCredentials, NullSaslCredentials

public abstract class SaslCredentials
    extends java.lang.Object

SaslCredentials is an abstract class that is the root of all the classes that implement various simple authentication and security layer mechanisms.

Implementing a new mechanism is done by subclassing SaslCredentials. It is necessary in each subclass to declare the method string as a static and to override the getMethod() accessor in subclass.

2.35.1 Constructors

public SaslCredentials()

public SaslCredentials(byte[] b)

2.35.2 getMethod

public java.lang.String getMethod()
Returns the method name for a SaslCredentials. This method is overridden in each subclass.

2.35.3 toBytes

public byte[] toBytes()

Returns a byte[] representing the credential value to be sent via protocol or that was received over protocol.

2.36 Class org.ietf.ldap.SchemaElementId

Subclasses:
  AttributeType, MatchingRuleId, ObjectClassId, SchemaDescriptionId, SyntaxId

public class SchemaElementId
  extends java.lang.Object

The SchemaElementId represents the OID and name(s) by which a given schema element is known. If both a name and OID are present in an instance then the instance can considered to assign the name to the OID.

2.36.1 Constructors

public SchemaElementId(OID oid)

public SchemaElementId(java.lang.String name)

public SchemaElementId(OID oid, java.lang.String name)

2.36.2 getOid

public OID getOid()

Return the OID if any associated with this SchemaElementId

2.36.3 setOid

public void setOid(OID oid)

Establish the OID for this SchemaElementId.

2.36.4 getName

public java.lang.String getName()

Return the name if any associated with this SchemaElementId.

2.36.5 setName

public void setName(java.lang.String name)

Set the name of this SchemaElementId.
public boolean equals(SchemaElementId other)

Compares two SchemaElementIds. If both have an OID then the values must be identical. On the other hand, if one or both have null OIDs then the names must match.

public java.lang.String toString()

If a name is available it is returned as the string representation for this id, else the associated OID in string form is returned.

public java.lang.String toOidString()

Return the string representation of the associated OID or null if there is no OID associated.

Class org.ietf.ldap.Scope

public final class Scope
    extends Enum

Defines the constants that represent the three different scopes for a LDAP search. Each constant has a name and code that represents its value in the LDAP ASN.1 definition. An application designer need only use toName and the static constants of this class: Scope.BASE, Scope.ONELEVEL, and SCOPE.SUBTREE.

public static final Scope BASE

Limits a search to the Entry that named by the and that matches the Filter.

public static final Scope ONELEVEL

Limits the search to the Entry named by the DN and that entries children.

public static final Scope SUBTREE

Limits the search to the sub-tree rooted at the Entry named by the DN.

public static Scope toScope(int code)

Used by ASN.1 decode routines to obtain the Scope constant that corresponds to an ASN.1 value.
Parameters:
   code - the ASN.1 value to map to a Scope

Returns: the Scope corresponding to code or null

2.37.3 toName

public static java.lang.String toName(int code)

Returns the String name for an ASN.1 value.

Parameters:
   code - ASN.1 value to map to a name

2.38 Class org.ietf.ldap.SDK

public class SDK

SDK defines constants related to an implementation of the LDAP API.

2.38.1 Fields

public static final java.lang.String version

Defines the version string of the interface instance.

Expires 8/99                                                [Page 47]

JAVA LDAP API                                            February 1999

public static final java.lang.String product

Defines the product or vendor string.

public static final java.lang.String release

Defines the vendor dependent release identification.

2.39 Class org.ietf.ldap.URLList

Subclasses:
   Continuation, Referral

public class URLList
   extends Entity

A Referral is a list of urls to other servers that should be contacted to perform some operation, such as a search.

2.39.1 Constructors

public URLList(java.util.Vector urls)

public URLList(java.lang.String[] s)

2.39.2 getAll

public java.lang.String[] getAll()

Returns an array of the URL strings in the URLList.
2.39.3 getURLs

public java.util.Vector getURLs()

Returns a Vector of the URLs.

2.40 Class org.ietf.ldap.HopLimitException

public class HopLimitException
    extends InterfaceException

Thrown when an interface specific time limit is exceeded.

2.40.1 Constructors

public HopLimitException()

public HopLimitException(java.lang.String s)

2.40.2 Class org.ietf.ldap.InterfaceException

Subclasses:

Expires 8/99

JAVA LDAP API February 1999

HopLimitException, InterfaceTimeLimitException,
LDAPURLExtensionNotSupportedException, ParseException,
ReferralException

public class InterfaceException
    extends LDAPException

An exception thrown when an error occurs specific to the API interface as opposed to an LDAPException related to the protocol.

2.40.3 Constructors

public InterfaceException()

public InterfaceException(java.lang.String s)

2.41 Class org.ietf.ldap.InterfaceTimeLimitException

public class InterfaceTimeLimitException
    extends InterfaceException

Thrown when an interface specific time limit is exceeded.

2.41.1 Constructors

public InterfaceTimeLimitException()

public InterfaceTimeLimitException(java.lang.String s)

2.42 Class org.ietf.ldap.LDAPException

Subclasses:
    InterfaceException, ProtocolException
public class LDAPException
    extends java.lang.Exception

A general purpose exception to be thrown by LDAP operations.

2.42.1 Constructors

public LDAPException()

public LDAPException(java.lang.String s)

public LDAPException(java.lang.Exception e)

2.42.2 getMatchedDN

public DN getMatchedDN()

Returns the DN related to the LDAPException or null if there is no associated DN.

Expires 8/99

JAVA LDAP API                                            February 1999

2.43 Class org.ietf.ldap.ParseException

public class ParseException
    extends InterfaceException

ParseException is thrown for errors in the well-formedness of RFC 2254 filter specifications and DNs. Since the ParseException only occurs as a result of processing local to the API, it is an instance of InterfaceException.

2.43.1 Constructors

public ParseException()

public ParseException(java.lang.String s)

2.44 Class org.ietf.ldap.ProtocolException

Subclasses:
    AdministrationLimitExceededException, AffectsMultipleDASasException, AttributeOrValueExistsException, AuthenticationMethodNotSupportedException, BusyException, ConfidentialityRequiredException, ConstraintViolationException, DisconnectionException, EntryAlreadyExistsException, InappropriateAuthenticationException, InappropriateMatchingException, InsufficientAccessRightsException, InvalidAttributeSyntaxException, InvalidCredentialsException, LoopDetectException, MatchedDNException, NamingViolationException, NoSuchAttributeException, NotAllowedOnNonLeafException, NotAllowedOnRDNException, ObjectClassModificationsProhibitedException, ObjectClassViolationException, OperationsErrorException, OtherException, ProtocolErrorException, SizeLimitExceeded Exception, StrongAuthenticationRequiredException,
public class ProtocolException
    extends LDAPException

An exception thrown when an error occurs specific to the API protocol.

2.44.1 Constructors

public ProtocolException()

public ProtocolException(java.lang.String s)

3. Package org.ietf.ldap.client

3.0.1 Description

Expires 8/99

JAVA LDAP API                                                February 1999

Provides classes that model LDAP protocol components of that are
particularly relevant to client-side applications.

The principal class is DirectoryClient. This class provides
synchronous/blocking access to a directory. The ClientConnection
together with the Interaction class models the basic asynchronous
access to a directory. SearchResults implements the EntityEnumeration
allowing an application to iterate over a collection of Entries and
(optionally) SearchReferences.

3.1 Interface org.ietf.ldap.client.Binder

public abstract interface Binder

This interface defines a single method that is used to perform a bind
request (sequence) via a ClientConnection. A class implementing this
interface, see DefaultBinder, will implement a use of the bind request
that is appropriate for some server or class of servers.

Typically a Binder will be produced by a BinderFactory, see
DefaultBinderFactory, during implicit referral handling. The Binder is
given access to the connection over which the bind is to be performed.

3.1.1 bind

public BindResponse bind(ClientConnection conn)
    throws LDAPException

Returns the final response to the bind. In the event that the bind is
terminated in error then an exception is thrown. Note that if a SASL
bind is used there may be many steps and only the final response will
be returned.

Parameters:
    conn - the connection to bind to

Returns: the final response to the bind
3.2 Interface org.ietf.ldap.client.BinderFactory

public abstract interface BinderFactory

This interface specifies a single method that will return a Binder corresponding to the given host and port. A BinderFactory may use an authentication database, display a dialog to the user, or any other method that is appropriate for a given application, in order to generate an instance of a class that will be able to complete the bind operation for a given ClientConnection.

3.2.1 getBinder

```java
public Binder getBinder(String host, int port)
    throws LDAPException
```

Returns a Binder suitable for performing a bind to the given host and port.

Parameters:
- host - the name of the LDAP server host
- port - on which the server resides

Throws:
- LDAPException - in the event of a failure during the bind.

3.3 Interface org.ietf.ldap.client.PageHandler

public abstract interface PageHandler

Provides the interface for handlers supplied by the application to be called when a SearchDone is received with a SimplePagedControl in effect.

A PageHandler is established via SearchSpec.setPageHandler(PageHandler) on either the SearchSpec passed in the DirectoryClient.search or the default SearchSpec associated with a DirectoryClient or the SearchSpec on an active SearchResults

If a search is performed with a non-zero page size, i.e., with a SimplePagedControl, then if the SearchSpec.pageHandler() is null then the SearchResults fetched pages implicitly until the search is finally done.

3.3.1 endOfPage

```java
public boolean endOfPage(SearchSpec spec,
    ClientConnection conn,
    Interaction x)
    throws LDAPException
```

Should return true if the search is DONE else return false if more
results should be anticipated.

Parameters:
- spec - the SearchSpec in effect
- conn - the ClientConnection over which the last page was received
- x - the Interaction that is to be used in further requests

Returns: true if the search is considered DONE else false

3.4 Class org.ietf.ldap.client.AddResponse

public class AddResponse
    extends Response

AddResponse implements the response to an LDAP add request. There is no

specific information provided. See Response.

3.4.1 Constructors

public AddResponse(LDAPException ex, Referral ref)

3.5 Class org.ietf.ldap.client.BindResponse

public class BindResponse
    extends Response

BindResponse implements the response to an LDAP bind request. The
specific information provided is the in-progress status of the bind and
the SASL credentials (if any) from the server.

3.5.1 Constructors

BindResponse

public BindResponse(LDAPException ex, Referral ref,
    boolean b,
    byte[] creds)

3.5.2 isInProgress

public boolean isInProgress()

Returns true if the server reports SASL bind in-progress.

3.5.3 getServerCredentials

public byte[] getServerCredentials()

Returns any SASLCredentials that the server sent in the response. to a
bind request.

3.6 Class org.ietf.ldap.client.ClientConnection

public class ClientConnection
    extends org.ietf.ldap.apdu.Connection
A ClientConnection supports asynchronous interaction with an LDAP server. Classes such as DirectoryClient and SearchResults may use a ClientConnection.

Through a ClientConnection, a user may issue the various requests specified in RFC 2251: bind, unbind, search, modify, modify dn, compare, abandon, operations in addition to making extended requests. Each of the request methods returns an Interaction on which the user will receive any responses to the requested LDAP operation.

A ClientConnection runs a Thread when instantiated. This thread listens for messages arriving from the server to which the ClientConnection is connected. When a message is received it is passed to the appropriate interaction from which it may be retrieved for application processing.

All LDAP operations accept an interaction on which any messages received in response to the operation request will be placed. In this way a single interaction may be associated with many outstanding requests distributed across multiple ClientConnections. There is no ambiguity in responses from different servers on a single interaction since all messages over the life of the API instance are assigned a unique message id.

If the interaction argument is null then the ClientConnection will create one to be used for responses to the given request. In all cases the request method returns the interaction on which the response(s) will be delivered.

All operations except for connect accept a Controls to specify any Controls that may be meaningful for the operation on the connected server. If no Controls are to be sent then null is used to signal their absence.

3.6.1 Constructors

public ClientConnection(String host, int port) throws LDAPException

Establishes a connection to an LDAP server at the given host on the specified port.

Parameters:
- host - to contact.
- port - the server is listening on.

3.6.2 bind

public Interaction bind(int version, 
                      DN dn, 
                      String password, 
                      Interaction x)

throws LDAPException

Binds to the server using simple authentication.
Parameters:
- version - integer representing bind version
- dn - distinguished name of manager
- password - password of manager

Returns: the Interaction on which the response will be returned

Throws:
- LDAPException - a generic exception

3.6.3 bind

Expires 8/99

JAVA LDAP API

February 1999

```java
public Interaction bind(int version,
    DN dn,
    byte[] password,
    Interaction x)
    throws LDAPException;
```

Binds to the server using simple authentication, the password is an arbitrary binary value.

Parameters:
- version - integer representing bind version
- dn - distinguished name of manager
- password - password of manager

Returns: the Interaction on which the response will be returned

Throws:
- LDAPException - a generic exception

3.6.4 bind

```java
public Interaction bind(int version,
    DN dn,
    SaslCredentials sasl,
    Controls controls,
    Interaction x)
    throws LDAPException;
```

Binds to the server using SASL authentication and controls.

Parameters:
- version - integer representing bind version
- dn - distinguished name of manager
- control - array of controls to send to server
- sasl - SASL credentials of manager

Returns: the Interaction on which the response will be returned

Throws:
- LDAPException - a generic exception

3.6.5 close

```java
public void close();
```
Remove this connection from the table of active connections to servers and then call close on Connection.

3.6.6 unbind

public void unbind()

    throws LDAPException

Expires 8/99 [Page 55]

JAVA LDAP API February 1999

Unbinds from the server.

Throws:
    LDAPException – a generic exception

3.6.7 search

public Interaction search(SearchSpec spec,
    Interaction x)

    throws LDAPException

Requests a search.

Returns: the Interaction on which the response will be returned

Throws:
    LDAPException – a generic exception

3.6.8 search

public Interaction search(DN base,
    Scope scope,
    AliasDeref deref,
    int size_limit,
    int time_limit,
    boolean types_only,
    Filter filter,
    AttributeDescription[] atts,
    Controls controls,
    Interaction x)

    throws LDAPException

Requests a search.

Parameters:
    base – place in LDAP tree to start searching
    scope – Scope.BASE, Scope.ONELEVEL, Scope.SUBTREE
    deref_aliases – AliasDeref.NEVER, AliasDeref.SEARCHING,
        AliasDeref.FINDING, AliasDeref.ALWAYS
    size_limit – maximum number of entries to be returned,
        or 0 if unlimited
    time_limit – maximum number of seconds the server should devote to
        the search, or 0 if unlimited
    types_only – true = return attributes only
    filter – search criteria to use
    atts – array of attributes to return or null
3.6.9 modify

public Interaction modify(DN base,
    Modification[] mods,
    Controls controls,
    Interaction x)

    throws LDAPException

Requests the modification of an entry.

Parameters:
    base - place inLDAP tree to modify
    mods - array of modifications to be made
    control - array of controls to send to server
    x - the Interaction to use or null

Returns: the Interaction on which the response will be returned

Throws:
    LDAPException - a generic exception

3.6.10 add

public Interaction add(DN entry,
    AttributeSet attrs,
    Controls controls,
    Interaction x)

    throws LDAPException

An entry is added to the directory with the given distinguished name
and Attributes. Optional controls may be supplied.

Parameters:
    dn - of the entry to be added
    attrs - set of attributes to add under the given distinguished name
    controls - array of controls to send to server, or null
    x - the Interaction to use or null

Returns: the Interaction on which the response will be returned

Throws:
    LDAPException - a generic exception

3.6.11 delete

public Interaction delete(DN entry,
    Controls controls,
    Interaction x)
throws LDAPException

Requests a delete.

Parameters:

Expires 8/99

JAVA LDAP API

            entry - to delete
          control - list of controls to send or null
            x - the Interaction to use or null

Returns: the Interaction on which the response will be returned

Throws:
      LDAPException - a generic exception

3.6.12 modifyDN

public Interaction modifyDN(DN entry,
                       RDN new_rdn,
                       boolean delete_old,
                       DN new_superior,
                       Controls controls,
                       Interaction x)

            throws LDAPException

Requests the renaming of an entry or sub-tree or the movement of an
entire sub-tree.

Parameters:
     base - location in the DIT to modify
     newRDN - new relative DN
     delete_old - if true delete the old RDN values
     new_superior - new parent for entry
     control - array of controls to send to server
     x - the Interaction to use or null

Returns: the Interaction on which the response will be returned

Throws:
      LDAPException - a generic exception

3.6.13 compare

public Interaction compare(DN entry,
                       AttributeValueAssertion ava,
                       Controls controls,
                       Interaction x)

            throws LDAPException

Requests a comparison of an assertion with an entry in the directory.

Parameters:
     entry - DN of entry to compare
     ava - assertion on attribute value to compare
     control - array of controls to send to server
     x - the Interaction to use or null
Returns: the Interaction on which the response will be returned

Throws:

Expires 8/99

JAVA LDAP API

February 1999

LDAPException - a generic exception

3.6.14 abandon

public void abandon(int message_id,
                    Controls controls)
    throws LDAPException

Requests that a specific operation be abandoned. The message_id may be
retrieved via Interaction.getMessageId() on the interaction associated
with the request. In the event that the interaction is shared across
multiple outstanding requests, it is appropriate for the application to
use Interaction.getMostRecentId() when the request is sent and to
manage the messageIds itself for potential future abandonment.

Parameters:
    message_id - of the operation to abandon.
    control - array of controls to send to server

Throws:
    LDAPException - a generic exception

3.6.15 extendedRequest

public Interaction extendedRequest(OID name,
                                   byte[] value,
                                   Controls controls,
                                   Interaction x)
    throws LDAPException

Makes an extended request.

Parameters:
    name - oid of the request
    value - of the request
    control - array of controls to send to server
    x - the Interaction to use or null

Returns: the Interaction on which the response will be returned

Throws:
    LDAPException - a generic exception

3.6.16 issueExtended

public Interaction issueExtended(org.ietf.ldap.apdu.ExtendedRequest
                                  exReq,
                                  Controls controls,
                                  Interaction x)
    throws LDAPException

3.7 Class org.ietf.ldap.client.CompareResponse
public class CompareResponse

Expires 8/99

JAVA LDAP API February 1999

extends Response

CompareResponse returns the status of an LDAP compare request.

3.7.1 Constructors

public CompareResponse(LDAPException ex,
                        Referral ref,
                        boolean success)

3.7.2 isTrue

public boolean isTrue()

Returns the status of the compare.

Returns: true if the compare succeeded; otherwise false

3.8 Class org.ietf.ldap.client.DefaultBinder

public class DefaultBinder
    implements Binder

This Binder simply binds anonymously to the given ClientConnection.

3.8.1 Constructors

public DefaultBinder()

3.8.2 bind

public BindResponse bind(ClientConnection conn)
    throws LDAPException

Returns the BindResponse resulting from an anonymous bind over the
given ClientConnection. The response should be a successful response
otherwise an exception will have been thrown.

Parameters:
    conn - the connection to bind to

Returns: the response to the bind

3.9 Class org.ietf.ldap.client.DefaultBinderFactory

public class DefaultBinderFactory
    implements BinderFactory

This BinderFactory returns a Binder that performs an anonymous bind on
any connection.

3.9.1 Constructors
public DefaultBinderFactory()

3.9.2 getBinder

public Binder getBinder(String host, int port)
  throws LDAPException

Returns a simple Binder, see DefaultBinder, that performs an anonymous bind on any ClientConnection.

Parameters:
  host - ignored
  port - ignored

Returns: instance of DefaultBinder

Throws:
  LDAPException - never thrown

3.10 Class org.ietf.ldap.client.DelResponse

public class DelResponse
  extends Response

DelResponse implements the response object that is sent as a result of an LDAP delete request. There is no specific information for this response, see Response.

3.10.1 Constructors

DelResponse

public DelResponse(LDAPException ex, Referral ref)

3.11 Class org.ietf.ldap.client.DirectoryClient

public class DirectoryClient
  extends V2DirectoryClient

Provides a simple synchronous or blocking interface to an LDAP Directory. Through a DirectoryClient, a user can issue the various requests specified in RFC 2251:
  bind,
  unbind,
  search,
  modify,
  modify dn,
  compare,
  add,
  del,
  abandon

in addition to making extended requests.
All operations except for connect have a variant that accepts a Controls to specify any Control(s) that may be meaningful for the operation on the connected server.

See Client and WatchChange for sample code that uses this object.

3.11.1 Constructors

public DirectoryClient(String host, int port) throws LDAPException

Given a previously established ClientConnection a DirectoryClient is constructed that will operate over the connection.

3.11.2 getProtocolVersion

public int getProtocolVersion()

Returns the protocol version used: 3.

3.11.3 getHost

public String getHost()

Returns: the host to which the Connection is connected

3.11.4 getPort

public int getPort()

Returns: the port that was contacted to establish the connection

3.11.5 getClientConnection

public ClientConnection getClientConnection()

Returns the underlying ClientConnection.

3.11.6 setBinderFactory

public void setBinderFactory(BinderFactory bf)

Allows the application to establish the BinderFactory that will be used to generate Binders during referral following

Parameters:
  bf - the new BinderFactory.

3.11.7 getBinderFactory

public BinderFactory getBinderFactory()
Returns the current BinderFactory.

3.11.8 isReferring

public boolean isReferring()

Return whether referrals are implicitly followed or not.

3.11.9 setReferring

public boolean setReferring(boolean referring)

Set whether to follow referrals implicitly or not.

Returns: the previous state of referral following

3.11.10 isReturnReferrals

public boolean isReturnReferrals()

Return true if referrals are to be returned when not implicitly following referrals.

3.11.11 setReturnReferrals

public boolean setReturnReferrals(boolean returning)

Set whether to return referrals or not.

Returns: previous state of returning referrals.

3.11.12 performReferral

public Response performReferral(Message msg)
    throws ReferralException, ProtocolException

Handles implicit referral processing if any for an LDAP response.

Parameters:
    msg - the response, possibly a referral

3.11.13 bind

public BindResponse bind(DN dn, SaslCredentials sasl)
    throws LDAPException

Binds to the server using SASL authentication.

Parameters:
    dn - of manager
    sasl - credentials of manager

Throws:

Expires 8/99

JAVA LDAP API February 1999

LDAPException - in the event of failure of the bind
3.11.14 bind

public BindResponse
    bind(DN dn, SaslCredentials sasl, Controls controls)
    throws LDAPException

Binds to the server using SASL authentication and controls.

Parameters:
    dn - of manager
    sasl - credentials of manager
    controls - array of controls to send to server

Throws:
    LDAPException - in the event of failure of the bind

3.11.15 search

public SearchResults search(SearchSpec spec)
    throws LDAPException

Requests an LDAP search to be performed using the supplied SearchSpec.

Parameters:
    searchSpec - provides the base search constraints and any additional controls

3.11.16 search

public SearchResults search(DN base,
    Scope scope,
    AliasDeref deref,
    int size_limit,
    int time_limit,
    boolean attrsOnly,
    Filter filter,
    AttributeDescription[] atts,
    Controls controls,
    Interaction y)
    throws LDAPException

Requests a search with controls.

Parameters:
    base - place in LDAP tree to start searching
    scope - Scope.BASE, Scope.ONELEVEL, Scope.SUBTREE
    deref_aliases - AliasDeref.NEVER, AliasDeref.SEARCHING,
    AliasDeref.FINDING, AliasDeref ALWAYS
    size_limit - maximum number of entries to be returned
    time_limit - maximum number of seconds to wait for an answer
    attrsOnly - true = return attributes only
    filter - search criteria to use

Returns: SearchResults EntryEnumeration
Throws:

- LDAPException - a generic exception

**3.11.17 modify**

```java
public ModifyResponse modify(DN base, ModifyOp op, AttributeSet attrs, Controls controls)
throws LDAPException
```

**3.11.18 modify**

```java
public ModifyResponse modify(DN base, Modification[] mods, Controls controls)
throws LDAPException
```

Modifies an entry of a directory.

**Parameters:**
- `base` - place in LDAP tree to modify
- `mods` - array of modifications to be made
- `controls` - array of controls to send to server

**Returns:** ModifyResponse from server or null if error

**Throws:**

- LDAPException - a generic exception

**3.11.19 add**

```java
public AddResponse add(DN entry, AttributeSet attrs, Controls controls)
throws LDAPException
```

An entry is added to the directory with the given distinguished name and Attributes. Optional controls may be supplied.

**Parameters:**
- `dn` - of the entry to be added
- `attrs` - set of attributes to add under the given distinguished name
- `controls` - array of controls to send to server, or null

**Returns:** Response from server or null if error

**Throws:**

- LDAPException - a generic exception

**3.11.20 delete**

```java
public DelResponse delete(DN entry, Controls controls)
throws LDAPException
```

Deletes an Entry from the directory.

**Parameters:**
- `entry` - to delete
- `control` - array of controls to send to server
Returns: DeleteResponse from server or null if error

Throws:
   LDAPException - a generic exception

3.11.21 modifyDN

public ModifyDNResponse modifyDN(DN entry,
   RDN new_rdn,
   boolean delete_old,
   DN new_superior,
   Controls controls)

   throws LDAPException

Requests a dn modification with controls.

Parameters:
   base - place in LDAP tree to modify
   newRDN - array of new relative dn’s
   delete_old - true = delete the old entry
   new_superior - new parent for entry
   controls - array of controls to send to server

Returns: ModifyDNResponse from server or null if error

Throws:
   LDAPException - a generic exception

3.11.22 compare

public CompareResponse compare(DN entry,
   AttributeValueAssertion ava,
   Controls controls)

   throws LDAPException

Requests a comparison with controls.

Parameters:
   entry - dn of entry to compare
   ava - assertion of attribute to compare
   control - array of controls to send to server

Returns: CompareResponse from server or null if error

Throws:
   LDAPException - a generic exception

3.11.23 abandon

public void abandon(int message_id,
   Controls controls)

   throws LDAPException

Requests an abandonment with controls.
Parameters:
message_id - of the operation to abandon.
control - array of controls to send to server

Throws:
LDAPException - a generic exception

3.11.24 extendedRequest

public ExtendedResponse extendedRequest(OID name, byte[] value)
throws LDAPException

Makes an extended request.

Parameters:
name - oid of the request
value - of the request

Returns:  ExtendedResponse from server or null if error

Throws:
LDAPException - a generic exception

3.11.25 extendedRequest

public ExtendedResponse
  extendedRequest(OID name, byte[] value, Controls controls)
throws LDAPException

Makes an extended request with controls.

Parameters:
name - oid of the request
value - of the request
control - array of controls to send to server

Returns:  ExtendedResponse from server or null if error

Throws:
LDAPException - a generic exception

3.11.26 issueExtended

public ExtendedResponse
  issueExtended(org.ietf.ldap.apdu.ExtendedRequest exReq,
                 Controls controls)

Expires 8/99                                                [Page  67]
JAVA LDAP API                                            February 1999
  throws LDAPException

3.12 Class org.ietf.ldap.client.DynamicRefreshResponse

public class DynamicRefreshResponse
  extends ExtendedResponse

This notification may be used by the server to advise the client that
the server is about to close the connection due to an error condition.
Note that this notification is NOT a response to an unbind requested by
the client. This notification is intended to assist clients in distinguishing between an error condition and a transient network failure. As with a connection close due to network failure, the client MUST NOT assume that any outstanding requests which modified the directory have succeeded or failed.

3.12.1 Fields

public static OID responseName

3.12.2 Constructors

public DynamicRefreshResponse()

public DynamicRefreshResponse(LDAPException ex, Referral ref)

3.12.3 toString

public String toString()

3.13 Class org.ietf.ldap.client.ExtendedResponse

Subclasses:
  DynamicRefreshResponse, StartTLSResponse, UnsolicitedNotification

public class ExtendedResponse
  extends Response

ExtendedResponse provides a way to implement new types of responses without changing the underlying LDAP protocol. Each extended response is required by RFC 2251 to have a unique OID that identifies the response.

ExtendedResponse provides a registry that supports the construction of a specific extended response by the protocol decoder (ASN.1 interpreter).

3.13.1 Constructors

public ExtendedResponse()

This constructor must be present in each extended response so that <Class>.newInstance() can be called.

Expires 8/99
[Page 68]
JAVA LDAP API February 1999

public ExtendedResponse(LDAPException ex,
  Referral ref,
  OID responseName)

public ExtendedResponse(LDAPException ex,
  Referral ref,
  OID responseName,
  byte[] response)

3.13.2 Register

public static void register(OID oid, java.lang.Class c)
Establishes the association of an OID for an extended response and the Class that implements the response. Each such response class must implement appropriate init(...) routines.

Parameters:
- c - the Class implementing the extended response
- oid - the unique OID of the response

3.13.3 toExtended

public static java.lang.Class toExtended(String oid)

Returns the Class corresponding to the given OID

Parameters:
- oid - the unique identifier of the response

3.13.4 init

public void init(LDAPException ex, Referral ref, OID responseName)

This method is called when constructing an instance of some class of extended response for which there is no response specific data. Thus the only information that is conveyed is the exception state, referral, and the OID name of the response

Parameters:
- ex - the error information if any
- ref - the referral if any
- responseName - the OID of the response being initialized

3.13.5 init

public void init(LDAPException ex, Referral ref, OID responseName, byte[] response)

This method is used when there is additional initialization data specific to the extended response. The additional data is conveyed via a byte[] that contains the octets received via the LDAP protocol. It may need to be subjected to further decoding during initialization. This is a response dependent issue.

Parameters:
- ex - error information if any
- ref - referral if any
- response - the response data as a byte[]
- responseName - the OID of the response

3.13.6 getResponseName

public OID getResponseName()

Returns the OID of the response.
3.14 Class org.ietf.ldap.client.ModifyDNResponse

public class ModifyDNResponse
   extends Response

CompareResponse models the response including Controls and messageId for a compare request.

3.14.1 Constructors

ModifyDNResponse

public ModifyDNResponse(LDAPException ex, Referral ref)

3.15 Class org.ietf.ldap.client.ModifyResponse

public class ModifyResponse
   extends Response

CompareResponse models the response including Controls and messageId for a compare request.

3.15.1 Constructors

public ModifyResponse(LDAPException ex, Referral ref)

3.16 Class org.ietf.ldap.client.NoticeOfDisconnection

public class NoticeOfDisconnection
   extends UnsolicitedNotification

This notification may be used by the server to advise the client that the server is about to close the connection due to an error condition. Note that this notification is NOT a response to an unbind requested by the client. This notification is intended to assist clients in distinguishing between an error condition and a transient network failure. As with a connection close due to network failure, the client MUST NOT assume that any outstanding requests which modified the directory have succeeded or failed.

3.16.1 Fields

public static OID responseName

3.16.2 Constructors

public NoticeOfDisconnection()

public NoticeOfDisconnection(LDAPException ex, Referral ref)

3.17 Class org.ietf.ldap.client.Response

Subclasses:
   AddResponse, BindResponse, CompareResponse, DelResponse,
public class Response extends Message

Response is the super class of all the different LDAP response classes. It provides access to the LDAPException or Referral if any that result from an LDAP request. A Response that has no exception or referral is a successful response.

3.17.1 Constructors

public Response()

public Response(LDAPException ex, Referral ref)

public Response(LDAPException ex, Referral ref, String message)

3.17.2 getMessage

public String getMessage()

3.17.3 isSuccess

public boolean isSuccess()

3.17.4 getException

public LDAPException getException()

3.17.5 getReferral

public Referral getReferral()

3.18 Class org.ietf.ldap.client.SearchDone

public class SearchDone extends Response

Expires 8/99

JAVA LDAP API

February 1999

CompareResponse models the response including Controls and messageId for a compare request.

3.18.1 Constructors

public SearchDone(LDAPException ex, Referral ref)

3.19 Class org.ietf.ldap.client.SearchEntry

public class SearchEntry extends Message

SearchEntry contains an single Entry resulting from a search

3.19.1 Constructors

public SearchEntry(Entry entry)
### 3.19.2 getEntry

```java
public Entry getEntry()
```

### 3.20 Class org.ietf.ldap.client.SearchReference

```java
public class SearchReference extends Message

SearchReference contains an reference to a location where more search results may be found.
```

#### 3.20.1 Constructors

```java
public SearchReference(Continuation cont)
```

#### 3.20.2 getContinuation

```java
public Continuation getContinuation()

Returns the Continuation (URLList) for this SearchReference.
```

### 3.21 Class org.ietf.ldap.client.SearchResults

```java
public class SearchResults implements EntityEnumeration

A SearchResults is returned by the search request on the DirectoryClient. SearchResults implements EntityEnumeration. The elements returned by a search may be accessed via:

```java
nextElement()
```
returns an Object which is either an Entry or a Referral and may throw

```java
NoSuchElementException
```

```java
nextEntry()
```
returns an Entry or throws either NoSuchElementException or one of the defined LDAPExceptions.

```java
next()
```
returns an Entity or throws either NoSuchElementException or one of the defined LDAPExceptions.

`nextElement()` implements the standard method of Enumeration and will require that its result be cast to the appropriate type to be used.

`nextEntry()` is useful in the typical case when implicit referral handling is enabled and the user desires to process each of the entries from the search. Aside from the return type of this method, it differs from `nextElement()` in that it may throw an LDAP specific exception as well as NoSuchElementException.
next() returns an Entity which is either an Entry or a Referral, and hence will be useful when !isReferring() and isReturnReferrals(). This method may also throw LDAP specific exceptions.

There are two methods for testing whether the end of the SearchResults have been reached: hasMoreElements() conforming to the standard method of Enumeration; and hasMore() which allows for an LDAP specific exception to be thrown if appropriate.

3.21.1 Constructors

SearchResults

public SearchResults(ClientConnection conn,
                      int messageId,
                      Interaction x,
                      SearchSpec spec)

Constructs a SearchResults with the originating connection and messageId as well as an Interaction over which search results will arrive. Note that results may arrive from many different ClientConnections due to SearchReference processing.

Parameters:
conn - originating ClientConnection
messageId - original messageId when search was initiated
x - the Interaction over which results will arrive
spec - the SearchSpec of the original arguments for the search

3.21.2 setBinderFactory

public void setBinderFactory(BinderFactory bf)

3.21.3 getBinderFactory

public BinderFactory getBinderFactory()
3.21.4  getInteraction

```java
public Interaction getInteraction()
```

3.21.5  abandon

```java
public void abandon()
    throws LDAPException
```

Allows the user to gracefully terminate a search (and any in-progress sub-searches) of the directory.

Throws:
   LDAPException - in the event of errors sending the abandon request

3.21.6  isReferring

```java
public boolean isReferring()
```

Return whether referrals are implicitly followed are not.

3.21.7  setReferring

```java
public boolean setReferring(boolean referring)
```

Set whether to follow referrals implicitly or not

Returns: the previous state of referral following

3.21.8  isReturnReferrals

Expires 8/99

JAVA LDAP API

February 1999

```java
public boolean isReturnReferrals()
```

Indicate whether referrals are to be returned or if not-followed implicitly signalled via a ReferralException

Returns: true if referrals are to be returned when not implicitly following referrals

3.21.9  setReturnReferrals

```java
public boolean setReturnReferrals(boolean returning)
```

Set whether to return referrals or not

Returns: previous state of returning referrals

3.21.10  getControls

```java
public Controls getControls()
```

Returns the Controls for the current search result entry or search result reference. May be null if the current result is not defined or end of results has been reached.
3.21.11 hasMoreElements

public boolean hasMoreElements()

Implements Enumeration.hasMoreElements(). If there is an Entity (either 
an Entry or a Referral) available then it returns true otherwise false.

This method may block until a result has been received or the end of 
the search has been detected (in brief by receiving a SearchResultDone 
from the server, but the condition may be much more complex owing to 
referral following).

Returns: true if there are more search results; else false

3.21.12 hasMore

public boolean hasMore()
   throws LDAPException

Implements EntityEnumeration.hasMore(). If there is an Entry available 
then it returns true otherwise false.

An LDAP specific exception may be thrown if the end of the search has 
been reached and either the final SearchResultDone signalled an 
exceptional condition or referral following led to an exceptional 
condition at some point, including referral following not being enabled 
and a SearchResultReference or SearchResultDone with a referral being 
received.

This method may block until a result has been received or the end of

Expires 8/99

JAVA LDAP API		February 1999

the search has been detected (in brief by receiving a SearchResultDone 
from the server, but the condition may be much more complex owing to 
referral following).

Returns: true if there are more search results; else false

Throws:
   LDAPException - thrown on IOException and other times

3.21.13 nextElement

public java.lang.Object nextElement()
   throws java.util.NoSuchElementException

Implements Enumeration.nextElement(). An Object which is an Entity 
(either an Entry or a Referral) is returned or NoSuchElementException 
is thrown if the end of the search results has been reached.

Any LDAP specific exceptions that have been accumulated during the 
search may be retrieved via inError() and getErrors().

This method may block until a result has been received or the end of 
the search has been detected (in brief by receiving a SearchResultDone 
from the server, but the condition may be much more complex owing to 
referral following).
Returns: the next Entry or Referral

Throws:
   java.util.NoSuchElementException - thrown at the completion of the search

3.21.14 next

public Entity next()
   throws LDAPException,
       java.util.NoSuchElementException

The semantics of this method are the same as nextElement() except that the return type is Entity and an LDAP specific exception may be thrown in the event that the end of the search has been reached and one or more LDAP specific exceptions occurred.

This method may block until a result has been received or the end of the search has been detected (in brief by receiving a SearchResultDone from the server, but the condition may be much more complex owing to referral following).

Returns: the next SearchResultEntry

Throws:
   java.util.NoSuchElementException - thrown when server sends a SearchResultDone object indicating no more entries.

Expires 8/99

JAVA LDAP API February 1999

LDAPException - thrown when LDAP specific errors occur.

3.21.15 nextEntry

public Entry nextEntry()
   throws LDAPException,
       java.util.NoSuchElementException

This method implements EntityEnumeration.nextEntry() and either returns an Entry or at the end of the search throws either NoSuchElementException or an LDAP specific exception if one occurred during the search.

Returns: the next SearchResultEntry

Throws:
   java.util.NoSuchElementException - thrown when server sends a SearchResultDone object indicating no more entries.

LDAPException - thrown when LDAP specific errors occur.

3.21.16 inError

public boolean inError()

Returns true if errors have been encountered during processing of this search. There may be multiple errors owing to the possibility of multiple sub-searches arising during implicit referral following.
3.21.17 getErrors

public java.util.Vector getErrors()

Returns the list of LDAPExceptions encountered during processing of the search.

3.22 Class org.ietf.ldap.client.SearchSpec

public class SearchSpec

implements java.lang.Cloneable

The SearchSpec provides a convenient way to bundle the various parameters to an LDAP search so that they may be used with more than one connection or multiple times with same connection. A SearchSpec may be cloned to and a complete set of accessors and setters are provided.

3.22.1 Constructors

public SearchSpec()

Constructs the default search spec that will retrieve the root dse for a server.

public SearchSpec(DN base, Scope scope, AliasDeref deref, int sizeLimit, int timeLimit, boolean attrsOnly, Filter filter, AttributeDescription[] atts, int psz, Controls cntrls)

3.22.2 base

public DN base()

3.22.3 scope

public Scope scope()

3.22.4 deref

public AliasDeref deref()

3.22.5 size_limit

public int size_limit()

3.22.6 time_limit

public int time_limit()
3.22.7 pageSize

public int pageSize()

3.22.8 pageHandler

public PageHandler pageHandler()

3.22.9 attrsOnly

public boolean attrsOnly()

3.22.10 filter

public Filter filter()

3.22.11 attrs

public AttributeDescription[] attrs()

3.22.12 controls

public Controls controls()

Expires 8/99

JAVA LDAP API

February 1999

3.22.13 setBase

public void setBase(DN b)

3.22.14 setScope

public void setScope(Scope s)

3.22.15 setDeref

public void setDeref(AliasDeref ad)

3.22.16 setSizeLimit

public void setSizeLimit(int sl)

3.22.17 setTimeLimit

public void setTimeLimit(int tl)

3.22.18 setPageSize

public void setPageSize(int p)

3.22.19 setPageHandler

public void setPageHandler(PageHandler ph)

3.22.20 setAttrsOnly

public void setAttrsOnly(boolean ao)
3.22.21 setFilter
public void setFilter(Filter f)

3.22.22 setAttrs
public void setAttrs(AttributeDescription[] a)

3.22.23 setControls
public void setControls(Controls c)

3.23 Class org.ietf.ldap.client.StartTLSResponse
public class StartTLSResponse
    extends ExtendedResponse

This notification may be used by the server to advise the client of the status of a request to initiate TLS negotiations on the current connection.

3.24 Class org.ietf.ldap.client.UnsolicitedNotification
Subclasses:
    NoticeOfDisconnection

public class UnsolicitedNotification
    extends ExtendedResponse

An UnsolicitedNotification is an LDAPMessage sent from the server to the client which is not in response to any LDAPMessage received by the server. It is used to signal an extraordinary condition in the server or in the connection between the client and the server. The notification is of an advisory nature, and the server will not expect any response to be returned from the client.

3.25 Class org.ietf.ldap.client.V2DirectoryClient
Subclasses:
   DirectoryClient

class V2DirectoryClient

Provides a simple synchronous(blocking) interface to an LDAP v2 Directory. Through a DirectoryClient, a user can issue the various requests specified in RFC 1777:
   bind, unbind, search, modify, modify dn, compare, add, del

The abandon operation is not supported since this class is a blocking interface to the directory. The one exception to this is the search which returns a SearchResults immediately; however, this class does provide an abandon() method that will abandon all search activities that may be in progress for the given SearchResults.

3.25.1 Constructors

public V2DirectoryClient()

public V2DirectoryClient(String host, int port) throws LDAPException

Establishes a ClientConnection to the specified host and port as the underlying connection for this DirectoryClient

public V2DirectoryClient(ClientConnection con)

Constructs a new DirectoryClient with the underlying ClientConnection

3.25.2 getProtocolVersion

public int getProtocolVersion()

Returns the protocol version in use. This is overridden in subclasses for other (3) protocol versions.

3.25.3 getHost

public String getHost()

Returns the name of the host that was contacted to establish the underlying ClientConnection

3.25.4 getPort

public int getPort()
Returns the port that was contacted to establish the underlying ClientConnection

3.25.5 getClientConnection

public ClientConnection getClientConnection()

Returns the ClientConnection for this DirectoryClient

3.25.6 getSearchSpec

public SearchSpec getSearchSpec()

Returns the current SearchSpec that represents the defaults for searches initiated through this connection.

3.25.7 setSearchSpec

public SearchSpec setSearchSpec(SearchSpec spec)

Set the current SearchSpec that represents the defaults for searches initiated through this connection.

Returns the previous SearchSpec.

3.25.8 getTimeLimit

public int getTimeLimit()

Returns the amount of time that the interface will wait for a response to a request. A value of 0(zero) indicates that no time limit is in effect.

Note that this is different from the SearchSpec.getTimeLimit() which is the maximum amount of time that the server is to spend on a search. This timeLimit applies to the maximum time in seconds before the DirectoryClient will throw an InterfaceTimeLimitException.

3.25.9 setTimeLimit

public void setTimeLimit(int newTL)

Sets the time limit to be applied to all operations initiated via this DirectoryClient

Parameters:
    newTL - the time limit to impose on operations

3.25.10 getHopLimit

public int getHopLimit()

3.25.11 setHopLimit

public void setHopLimit(int hops)
Sets the hop limit to be applied to all operations initiated via this DirectoryClient. The hop limit constrains the number of steps of referral following that the interface will perform before throwing a HopLimitException.

The hop limit defaults to 10.

Parameters:
   newTL - the hop limit to impose on operations

3.25.12 isReferring

public boolean isReferring()

Return whether referrals are implicitly followed or not.

Expires 8/99

JAVA LDAP API February 1999

3.25.13 isReturnReferrals

public boolean isReturnReferrals()

Return true if referrals are to be returned when not implicitly following referrals.

3.25.14 performReferral

public Response performReferral(Response msg)
   throws ReferralException, ProtocolException

LDAP v2 doesn’t permit referrals so this method simply returns its argument. This method is specialized in DirectoryClient to actually provide for referral handling.

Parameters:
   msg - the response, possibly a referral

Returns: the response argument

3.25.15 bind

public BindResponse bind(String dn, String password)
   throws LDAPException

Binds to the server using simple authentication.

Parameters:
   name - of manager
   password - of manager

Throws:
   LDAPException - a generic exception

3.25.16 bind

public BindResponse bind(DN dn, String password)
   throws LDAPException
Binds to the server using simple authentication.

Parameters:
   dn - of manager
   password - of manager

Throws:
   LDAPException - a generic exception

3.25.17 unbind

public void unbind() throws LDAPException

Expires 8/99

JAVA LDAP API February 1999

Unbinds from the server.

Throws:
   LDAPException - a generic exception

3.25.18 search

public SearchResults search(String base,
   Scope scope,
   String filter,
   String[] attrs,
   boolean attrsOnly)
   throws LDAPException

Corresponds to ldap_search_s of the C api except that a asynchronous
SearchResults is returned rather than waiting for all the results
before proceeding

Parameters:
   base - the distinguished name to base the search
   scope - one of Scope.BASE, Scope.ONELEVEL, or Scope.SUBTREE
   filter - a string conforming to RFC 2254 expressing the filter to use
   attrs - a list of the attribute description strings of the attributes
to return
   attrsOnly - true if only the attribute values should be returned

Returns: the SearchResults of the search

Throws:
   LDAPException - thrown if an error occurs during the search or
processing of the filter of base

3.25.19 search

public SearchResults search(SearchSpec spec)
   throws LDAPException

Requests an LDAP search to be performed using the supplied SearchSpec.

Parameters:
   searchSpec - provides the base search constraints, if any controls
are present they are ignored.

3.25.20 search

public SearchResults search(DN base,
   Scope scope,
   AliasDeref deref,
   int size_limit,
   int time_limit,
   boolean attrsOnly,
   Filter filter,
   AttributeDescription[] atts)

Requests a search.

Parameters:
   base - place in LDAP tree to start searching
   scope - Scope.BASE, Scope.ONELEVEL, Scope.SUBTREE
   deref_aliases - AliasDeref.NEVER, AliasDeref.SEARCHING,
                  AliasDeref.FINDING, AliasDeref.ALWAYS
   size_limit - maximum number of entries to be returned
   time_limit - maximum number of seconds to wait for an answer
   attrsOnly - true = return attributes only
   filter - search criteria to use
   atts - array of attributes to return

Returns:  SearchResults EntryEnumeration

Throws  LDAPException - a generic exception

3.25.21 modify

public ModifyResponse modify(DN base, ModifyOp op, Attribute attr)
   throws LDAPException

3.25.22 modify

public ModifyResponse modify(DN base, ModifyOp op, AttributeSet attrs)
   throws LDAPException

3.25.23 modify

public ModifyResponse modify(DN base, Modification[] mods)
   throws LDAPException

Requests the modification of an Entry.

Parameters:
   base - place in LDAP tree to modify
   mods - array of modifications to be made

Returns:  ModifyResponse from server or null if error

Throws:
LDAPException - typically signalling a protocol error

3.25.24 add

public AddResponse add(Entry entry)
    throws LDAPException

An Entry is added to the directory. The Entry supplies the
distinguished name and attributes that constitute the new entry.

Parameters:
    entry - to be added

Returns: Response from server or null if error

Throws:
    LDAPException - a generic exception

3.25.25 add

public AddResponse add(DN dn, AttributeSet attrs)
    throws LDAPException

An entry is added to the directory with the given distinguished name
and Attributes.

Parameters:
    dn - of the entry to be added
    attrs - set of attributes to add under the given distinguished name

Returns: Response from server or null if error

Throws:
    LDAPException - a generic exception

3.25.26 delete

public DelResponse delete(DN entry)
    throws LDAPException

Deletes an Entry from the directory

Parameters:
    entry - to delete

Returns: DeleteResponse from server or null if error

Throws:
    LDAPException - a generic exception

3.25.27 modifyDN

public ModifyDNResponse modifyDN(DN entry,
    RDN new_rdn,
    boolean delete_old,
    DN new_superior)
Throws LDAPException

Requests a dn modification: either modifying the leftmost RDN of an
Entry or moving a sub-tree rooted at new_superior.

Parameters:
  entry - place in LDAP tree to modify
  newRDN - new relative dn’s

Expires 8/99 [Page 86]

JAVA LDAP API February 1999

  delete_old - true if the old entry is to be deleted
  new_superior - new parent for entry

Returns: ModifyDNResponse from server or null if error

Throws:
  LDAPException - a generic exception

3.25.28 compare

public CompareResponse compare(DN entry, AttributeValueAssertion ava)
  throws LDAPException

Requests a comparison.

Parameters:
  entry - dn of entry to compare
  ava - assertion of attribute to compare

Returns: CompareResponse from server or null if error

Throws:
  LDAPException - a generic exception

3.25.29 abandon

public void abandon(int message_id)
  throws LDAPException

Requests an abandonment.

Parameters:
  message_id - of the operation to abandon.
  control - array of controls to send to server

Throws:
  LDAPException - a generic exception

3.26 Class org.ietf.ldap.client. AdministrationLimitExceededException

public class AdministrationLimitExceeded Exception
  extends ProtocolException

Exception thrown by client operations when the result code returned
from the server is adminLimitExceeded

3.26.1 Constructors
public AdministrationLimitExceededException()

public AdministrationLimitExceededException(String s)

3.27 Class org.ietf.ldap.client.AffectsMultipleDSAsException

Expires 8/99                                                [Page  87]
JAVA LDAP API                                            February 1999

public class AffectsMultipleDSAsException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is affectsMultipleDSAs.

3.27.1 Constructors

public AffectsMultipleDSAsException()

public AffectsMultipleDSAsException(String s)

3.28 Class org.ietf.ldap.client.AliasDereferencingProblemException

public class AliasDereferencingProblemException
    extends MatchedDNException

Exception thrown by client operations when the result code returned from the server is aliasDereferencingProblem.

3.28.1 Constructors

public AliasDereferencingProblemException(String s, DN matchedDN)

3.29 Class org.ietf.ldap.client.AliasProblemException

public class AliasProblemException
    extends MatchedDNException

Exception thrown by client operations when the result code returned from the server is aliasProblem.

3.29.1 Constructors

public AliasProblemException(String s, DN matchedDN)

3.30 Class org.ietf.ldap.client.AttributeOrValueExistsException

public class AttributeOrValueExistsException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is attributeOrValueExists.

3.30.1 Constructors

public AttributeOrValueExistsException()

public AttributeOrValueExistsException(String s)
3.31 Class
org.ietf.ldap.client.AuthenticationMethodNotSupportedException

public class AuthenticationMethodNotSupportedException

Expires 8/99                                               [Page  88]
JAVA LDAP API                                             February 1999

    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is authenticationMethodNotSupported.

3.31.1 Constructors

public AuthenticationMethodNotSupportedException()

public AuthenticationMethodNotSupportedException(String s)

3.32 Class org.ietf.ldap.client.BusyException

public class BusyException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is busy.

3.32.1 Constructors

public BusyException()

public BusyException(String s)

3.33 Class org.ietf.ldap.client.ConfidentialityRequiredException

public class ConfidentialityRequiredException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is confidentialityRequired.

3.33.1 Constructors

public ConfidentialityRequiredException()

public ConfidentialityRequiredException(String s)

3.34 Class org.ietf.ldap.client.ConstraintViolationException

public class ConstraintViolationException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is constraintViolation.

3.34.1 Constructors

public ConstraintViolationException()

public ConstraintViolationException(String s)
3.35 Class org.ietf.ldap.client.DisconnectionException

A DisconnectionException is thrown in the case that an ExtendedResponse is received that has messageId == 0 and the responseName is 1.3.6.1.4.1.1466.20036.

Due to the asynchronous support in the SDK a response message may be received and enqueued for processing well in advance of when it is actually selected for processing. Delaying the signalling of the exception until processing makes it easier to process protocol and related interface events in the order in which they occurred. In other words there is no out of band signalling of exceptions.

3.35.1 Constructors

public DisconnectionException()

public DisconnectionException(String s)

3.36 Class org.ietf.ldap.client.EntryAlreadyExistsException

Exception thrown by client operations when the result code returned from the server is entryAlreadyExists.

3.36.1 Constructors

public EntryAlreadyExistsException()

public EntryAlreadyExistsException(String s)

3.37 Class org.ietf.ldap.client.InappropriateAuthenticationException

Exception thrown by client operations when the result code returned from the server is inappropriateAuthentication.

3.37.1 Constructors

public InappropriateAuthenticationException()

public InappropriateAuthenticationException(String s)

3.38 Class org.ietf.ldap.client.InappropriateMatchingException

Exception thrown by client operations when the result code returned from the server is inappropriateMatching.
Exception thrown by client operations when the result code returned from the server is inappropriateMatching.

3.38.1 Constructors

public InappropriateMatchingException()

public InappropriateMatchingException(String s)

3.39 Class org.ietf.ldap.client.InsufficientAccessRightsException

public class InsufficientAccessRightsException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is insufficientAccessRights.

3.39.1 Constructors

public InsufficientAccessRightsException()

public InsufficientAccessRightsException(String s)

3.40 Class org.ietf.ldap.client.InvalidAttributeSyntaxException

public class InvalidAttributeSyntaxException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is invalidAttributeSyntax.

3.40.1 Constructors

public InvalidAttributeSyntaxException()

public InvalidAttributeSyntaxException(String s)

3.41 Class org.ietf.ldap.client.InvalidCredentialsException

public class InvalidCredentialsException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is invalidCredentials.

3.41.1 Constructors

public InvalidCredentialsException()

public InvalidCredentialsException(String s)

3.42 Class org.ietf.ldap.client.InvalidDNSyntaxException

public class InvalidDNSyntaxException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is invalidDNSyntax.
extends MatchedDNException

Exception thrown by client operations when the result code returned from the server is invalidDNSyntax.

### 3.42.1 Constructors

```
public InvalidDNSyntaxException(String s, DN matchedDN)
```

### 3.43 Class org.ietf.ldap.client.LDAPURLExtensionNotSupportedException

defines LDAPURLExtensionNotSupportedException

extends InterfaceException

This exception is thrown by an application if it attempts to process an LDAP URL containing a critical extension which is not supported as specified in RFC 2255.

### 3.43.1 Constructors

```
public LDAPURLExtensionNotSupportedException()

public LDAPURLExtensionNotSupportedException(String s)
```

### 3.44 Class org.ietf.ldap.client.LoopDetectException

defines LoopDetectException

extends ProtocolException

Exception thrown by client operations when the result code returned from the server is loopDetect

### 3.44.1 Constructors

```
public LoopDetectException()

LoopDetectException

public LoopDetectException(String s)
```

### 3.45 Class org.ietf.ldap.client.MatchedDNException

Subclasses:
- AliasDereferencingProblemException
- AliasProblemException
- InvalidDNSyntaxException
- NoSuchObjectException

defines MatchedDNException

extends ProtocolException

This exception is the root of a sub-tree of the exceptions that contain a matchedDN property. This exception is not itself generated within the interface and hence never thrown. See its subclasses.

### 3.45.1 Constructors

Expires 8/99
public MatchedDNException(String s, DN matchedDN)

3.45.2 getMatchedDN

public DN getMatchedDN()

3.46 Class org.ietf.ldap.client.NamingViolationException

public class NamingViolationException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is namingViolation.

3.46.1 Constructors

public NamingViolationException()

public NamingViolationException(String s)

3.47 Class org.ietf.ldap.client.NoSuchAttributeException

public class NoSuchAttributeException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is noSuchAttribute.

3.47.1 Constructors

public NoSuchAttributeException()

public NoSuchAttributeException(String s)

3.48 Class org.ietf.ldap.client.NoSuchObjectException

public class NoSuchObjectException
    extends MatchedDNException

Exception thrown by client operations when the result code returned from the server is noSuchObject.

3.48.1 Constructors

public NoSuchObjectException(String s, DN matchedDN)

3.49 Class org.ietf.ldap.clientNotAllowedOnNonLeafException

public class NotAllowedOnNonLeafException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is notAllowedOnNonLeaf.

3.49.1 Constructors
public NotAllowedOnNonLeafException()

public NotAllowedOnNonLeafException(String s)

3.50 Class org.ietf.ldap.client.NotAllowedOnRDN Exception

public class NotAllowedOnRDNException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is notAllowedOnRDN.

3.50.1 Constructors

public NotAllowedOnRDNException()

public NotAllowedOnRDNException(String s)

3.51 Class org.ietf.ldap.client.ObjectClassModificationsProhibitedException

public class ObjectClassModificationsProhibitedException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is objectClassModificationsProhibited.

3.51.1 Constructors

public ObjectClassModificationsProhibitedException()

public ObjectClassModificationsProhibitedException(String s)

3.52 Class org.ietf.ldap.client.ObjectClassViolationException

public class ObjectClassViolationException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is objectClassViolation.

3.52.1 Constructors

public ObjectClassViolationException()

public ObjectClassViolationException(String s)

3.53 Class org.ietf.ldap.client.OperationsErrorException

public class OperationsErrorException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is operationsError.
3.53 Constructors

public OperationsErrorException()

public OperationsErrorException(String s)

3.54 Class org.ietf.ldap.client.OtherException

public class OtherException
  extends ProtocolException

Exception thrown by client operations when the result code returned from the server is other.

3.54.1 Constructors

public OtherException()

public OtherException(String s)

3.55 Class org.ietf.ldap.client.ProtocolErrorException

public class ProtocolErrorException
  extends ProtocolException

Exception thrown by client operations when the result code returned from the server is protocolError.

3.55.1 Constructors

public ProtocolErrorException()

public ProtocolErrorException(String s)

3.56 Class org.ietf.ldap.client.ReferralException

public class ReferralException
  extends InterfaceException

ReferralException is thrown in the event that either a Referral Result or a SearchResultReference is received on a ClientConnection and referral processing is not enabled on that ClientConnection.

3.56.1 Constructors

public ReferralException()

public ReferralException(String s)

3.57 Class org.ietf.ldap.client.SizeLimitExceededException

Expiration 8/99

JAVA LDAP API February 1999

Expires 8/99

public class SizeLimitExceededException
  extends ProtocolException

This exception is thrown when a response Message is processed that contains an LDAPResult that signals sizeLimitExceeded.
Due to the asynchronous support in the SDK a response message may be received and enqueued for processing well in advance of when it is actually selected for processing. Delaying the signalling of the exception until processing makes it easier to process protocol and related interface events in the order in which they occurred. In other words there is no out of band signalling of exceptions.

3.57 Constructors

public SizeLimitExceededException()

public SizeLimitExceededException(String s)

3.58 Class org.ietf.ldap.client.StrongAuthRequiredException

public class StrongAuthenticationRequiredException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is strongAuthenticationRequired.

3.58.1 Constructors

public StrongAuthenticationRequiredException()

public StrongAuthenticationRequiredException(String s)

3.59 Class org.ietf.ldap.client.TimeLimitExceededException

public class TimeLimitExceededException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is timeLimitExceeded.

3.59.1 Constructors

public TimeLimitExceededException()

public TimeLimitExceededException(String s)

3.60 Class org.ietf.ldap.client.UnavailableCriticalExtensionException

public class UnavailableCriticalExtensionException
    extends ProtocolException

Exception thrown by client operations when the result code returned from the server is unavailableCriticalExtension.

3.60.1 Constructors

public UnavailableCriticalExtensionException()

public UnavailableCriticalExtensionException(String s)
3.61 Class org.ietf.ldap.client.UnavailableException

public class UnavailableException
   extends ProtocolException

Exception thrown by client operations when the result code returned from the server is unavailable.

3.61.1 Constructors

public UnavailableException()

public UnavailableException(String s)

3.62 Class org.ietf.ldap.client.UndefinedAttributeTypeException

public class UndefinedAttributeTypeException
   extends ProtocolException

Exception thrown by client operations when the result code returned from the server is undefinedAttributeType.

3.62.1 Constructors

public UndefinedAttributeTypeException()

public UndefinedAttributeTypeException(String s)

3.63 Class org.ietf.ldap.client.UnwillingToPerformException

public class UnwillingToPerformException
   extends ProtocolException

Exception thrown by client operations when the result code returned from the server is unwillingToPerform.

3.63.1 Constructors

public UnwillingToPerformException()

public UnwillingToPerformException(String s)

4. Package org.ietf.ldap.ldif

4.0.1 Description

Provides classes that implement reading and writing LDAP Interchange Format streams.

The principal classes are LDIFReader and LDIFWriter. These classes are used in the translation between LDIF to and from Record objects. Records are composed of a directory name and a RecordContent object, which is an abstract class subclassed as an AddRecordContent, AttributeRecordContent, DeleteRecordContent, ModifyDNRecordContent, or ModifyRecordContent. RecordContent stores information about the change type (or information in the case of AttributeRecordContent) specified
by the LDIF record.

4.1 Class org.ietf.ldap.ldif.AddRecordContent

public class AddRecordContent
    extends RecordContent

A subtype of RecordContent which stores attribute content to be added to an LDAP directory element.

4.1.1 Constructors

public AddRecordContent()

public AddRecordContent(Attribute attribute)

Parameters:
    attribute - an Attribute to be added to an existing LDAP directory entry

public AddRecordContent(Attribute[] attributes)

Parameters:
    attribute - an array of Attributes to be added

4.1.2 toString

public String toString()

Returns: an LDIF String representation.

4.1.3 addAttribute

public void addAttribute(Attribute attribute)

Adds an Attribute element to the list of Attributes to be added to an existing LDAP directory entry.

Parameters:
    attribute - the Attribute to be added to the entry

4.1.4 getAttributes

public Attribute[] getAttributes()

Returns: an array of Attribute objects to be added to an existing LDAP directory

4.1.5 addElement

public void addElement(String element)
    throws ParseException

Parameters:
    element - A line of text to be parsed. Since this object stores attribute additions, element should be a parsable LDIF String, an
attribute to be added to an existing LDAP directory entry.

Throws:
  ParseException - If error reading the input element

4.2 Class org.ietf.ldap.ldif.AttributeRecordContent

public class AttributeRecordContent
  extends RecordContent

A subtype of RecordContent which stores records of attributes, used when adding entries to a directory remotely.

4.2.1 Constructors

public AttributeRecordContent()

public AttributeRecordContent(Attribute attribute)

public AttributeRecordContent(AttributeSet attributes)

4.2.2 toString

public String toString()

Returns: an LDIF String representation.

4.2.3 addAttribute

public void addAttribute(Attribute attribute)

Parameters:
  attribute - the attribute to add

4.2.4 getAttributes

public Attribute[] getAttributes()

Returns: an array of Attributes

4.2.5 addElement

Expires 8/99

JAVA LDAP API

February 1999

public void addElement(String element)
  throws ParseException

Parameters:
  element - A line of text to be parsed. Since this object stores attributes, element should be a String representation of an attribute.

Throws:
  ParseException - If error reading the input element

4.3 Class org.ietf.ldap.ldif.DeleteRecordContent

public class DeleteRecordContent
extends RecordContent

A subtype of RecordContent. Since there is no content for a deletion record, this class doesn’t do much.

4.3.1 toString

public String toString()

Returns: a String representation of itself

4.3.2 addElement

public void addElement(String ignored)
  throws ParseException

Since it is undefined to add elements to a delete record, just throw an exception.

Throws:
   ParseException - Always thrown since it is undefined to add elements to a DeleteRecordContent.

4.4 Class org.ietf.ldap.ldif.LDIFReader

public class LDIFReader

LDIFReader reads LDIF information from either the keyboard or file, creating Record objects which can be retrieved by calling read().

4.4.1 Constructors

public LDIFReader(java.io.File f)
  throws java.io.FileNotFoundException

Constructs an instance which reads LDIF information read from the File parameter.

Parameters:
   f - File object from which to read LDIF information.
Throws:
    java.io.FileNotFoundException - If there is a problem using File

public LDIFReader(java.io.InputStream is)

Constructs an instance which reads LDIF information read from the File parameter.

Parameters:
    is - InputStream from which to read LDIF content.

4.4.2 read

public Record read()
    throws ParseException

Reads a Record from the input stream set at construction time.

Returns:  Record from input stream. Null if end of input file or if
"#quit" is encountered.

Throws:
    ParseException - If a syntax error is detected.

4.4.3 getVersion

public int getVersion()

Returns: int representing version of LDIF input (defaults to 0)

4.5 Class org.ietf.ldap.ldif.LDIFWriter

public class LDIFWriter

LDIFWriter serializes Record objects according to LDIF syntax onto an
OutputStream.

4.5.1 Constructors

public LDIFWriter(java.io.OutputStream os)

Parameters:
    os - OutputStream to which to write LDIF information

public LDIFWriter(java.io.File f)
    throws java.io.IOException

Parameters:
    f - File to which to write LDIF information.

Throws:
    java.io.IOException - If the File cannot be created.
4.5.2 write

public void write(Record record)
    throws java.io.IOException
        ProtocolException

Writes a Record object directly to the output stream. Note: a given
LDAP file is either a series of directory entries (Records with
AttributeRecordContent set), or a series of modifications (Records with
AddRecordContent, DeleteRecordContent, ModifyRecordContent, or
ModifyDNRecordContent. An LDIF file MUST NOT contain both types of
records. An attempt to deviate from this protocol will result in an
exception.

Parameters:
    record - to be written

Throws:
    java.io.IOException - on error during writing
    ProtocolException - If an attempt is made to combine a set of
directory entries with directory change records

4.5.3 write

public void write(Entry entry)
    throws java.io.IOException

Writes an Entry object directly to the output stream.

Parameters:
    entry - to be written

Throws:
    java.io.IOException - If there is a problem LDIF writing information
to the output source

4.5.4 close

public void close()
    throws java.io.IOException

Closes the underlying OutputStream.

4.5.5 isSafe

public static boolean isSafe(String s)

Tests the "safety" of a string to be written. Returns false if the
String should be base-64 encoded before being written.

Parameters:
    s - the String to test

4.6 Class org.ietf.ldap.ldif.ModifyDNRecordContent
public class ModifyDNRecordContent
    extends RecordContent

A subtype of RecordContent. Stores information specific to an LDIF dn modification record.

4.6.1 Constructors

public ModifyDNRecordContent()

Constructs a default instance.

public ModifyDNRecordContent(RDN new_rdn,
                                      boolean delete_old_rdn,
                                      DN new_superior)

Constructs an instance with specified information.

4.6.2 toString

public String toString()

Returns: a String representation of itself in LDIF

4.6.3 setNewRDN

public void setNewRDN(RDN new_rdn)

Sets the new rdn value of the entry to set when making the modification.

Parameters:
    new_rdn - String representing new rdn

4.6.4 getNewRDN

public RDN getNewRDN()

Returns: String representing new rdn value to set when making the modification

4.6.5 setDeleteOldRDN

public void setDeleteOldRDN(boolean delete_old_rdn)

Sets the contents of the delete old rdn value which describes what to do about the old rdn when making the modification.

Parameters:
    delete_old_rdn - boolean representing said flag in an LDAP modify dn message operation

4.6.6 getDeleteOldRDN
public boolean getDeleteOldRDN()

Returns: boolean representing whether or not to delete old rdn when making the modification.

4.6.7 setNewSuperior

public void setNewSuperior(DN new_superior)

Sets the new superior value of the entry when making the modification. This optional parameter may be null, meaning keep the same parent of the entry being modified. A blank DN is used to indicate the entry should be moved below the root entry.

Parameters:
   new_superior - DN representing said information in an LDAP modify dn message operation

4.6.8 getNewSuperior

public DN getNewSuperior()

Returns: DN representing the new superior value of the entry when making the modification. This optional parameter may be null, meaning keep the same parent of the entry being modified. A blank DN is used to indicate the entry should be moved below the root entry.

4.6.9 addElement

public void addElement(String element)
   throws ParseException

Parses the input String and sets relevant dn modification parameters represented in this object.

Parameters:
   element - A String containing some information relevant to a dn modification operation in LDIF format

Throws:
   ParseException - If there is an error reading the input element

4.7 Class org.ietf.ldap.ldif.ModifyRecordContent

public class ModifyRecordContent
   extends RecordContent

A subtype of RecordContent which stores the modification content of an LDAP element to be modified.

4.7.1 Constructors
public ModifyRecordContent()

public ModifyRecordContent(Modification modification)

public ModifyRecordContent(Modification[] modifications)

4.7.2 toString

public String toString()

Returns: an String representation of itself in LDIF

4.7.3 addModification

public void addModification(Modification modification)

Adds an Modification element to be represented

Parameters:
  modification - the modification to add

4.7.4 getModifications

public Modification[] getModifications()

Returns: an array of Modification objects stored in this object.

4.7.5 addElement

public void addElement(String element)
  throws ParseException

Parameters:
  element - A line of text to be parsed. Since this object stores
            modifications, element should be a String representation of an
            modification or separator (-) as specified by LDIF.

Throws:
  ParseException - If error reading input element

4.8 Class org.ietf.ldap.ldif.Record

public class Record

Record is an object created by an LDIFReader when parsing LDIF
information. It consists of a dn and an object implementing the
RecordContent interface, which represents record specific information.

4.8.1 Constructors

public Record(DN dn)

public Record(DN dn, RecordContent rc)
4.8.2 toString

public String toString()

Returns: a String representation of itself in LDIF

4.8.3 setContent

public void setContent(RecordContent content)

Sets the RecordContent to content.

4.8.4 getDN

public DN getDN()

Returns: DN of directory entry referred to by this record.

4.8.5 getContent

public RecordContent getContent()

Returns: RecordContent object which specifies change type specific information.

4.9 Class org.ietf.ldap.ldif.RecordContent

Subclasses:
AddRecordContent, AttributeRecordContent, DeleteRecordContent,
ModifyDNRecordContent, ModifyRecordContent

public abstract class RecordContent

RecordContent is an abstract class which specifies information specific
to change operations add, delete, modify, and modify dn.

4.9.1 Fields

public static final int CONTENT_UNDEFINED

public static final int CONTENT_ADD

public static final int CONTENT_DELETE

public static final int CONTENT_MODIFY

public static final int CONTENT_MODIFY_DN

public static final int CONTENT_ATTRIBUTES

4.9.2 Constructors

public RecordContent(int type)
Constructs an instance of the specified type.

4.9.3 getType

public int getType()

Returns: an integer representing the type of this instance. Use public
constants CONTENT_ADD, CONTENT_DELETE, CONTENT_MODIFY,
CONTENT_MODIFY_DN, CONTENT_ATTRIBUTES for comparison.

4.9.4 toString

public abstract String toString()

Returns: a String representation of itself in LDIF

4.9.5 addElement

public abstract void addElement(String element)
     throws ParseException

Adds the element to the record content.

Parameters:
   element - String of content specific information to be parsed. The
            action taken is dependent on the specific subclass of
            RecordContent.

Throws:
   ParseException - if there is a syntax error in the input String

5. Package org.ietf.ldap.schema

5.0.1 Description

Provides classes that implement the components of the schema on LDAP
servers representing Attributes, Objectclasses, and MatchingRules.

5.1 Class org.ietf.ldap.schema.AttributeDefinition

public class AttributeDefinition
     extends SchemaDefinition

AttributeDefinition represents the Attribute Types as part of an LDAP
schema defined in section 4.2 of RFC 2252. Attribute type definitions
identify the object identifier by which an attribute is known, its
syntax, associated matching rules, whether it is an operational
attribute and if so its type, whether it is a collective attribute,
whether it is permitted to have multiple values and whether or not it
is derived from another attribute type.

5.1.1 Fields

public static final String TYPE
5.1.2 Constructors

public AttributeDefinition(String value) throws ProtocolException

Constructs an instance from a String containing terminal symbols and
variables as specified in section 4.2 of RFC 2252.

Parameters:
  value - the String containing information to be parsed

Throws:
  ProtocolException - If there is a problem with the data in value

AttributeDefinition

public AttributeDefinition(AttributeType oid)

Constructs an instance from specified parameters

Parameters:
  oid - AttributeType representing the oid of this attribute

5.1.3 toString

public String toString()

Returns: a String representation of itself, suitable for storing back
in directory.

5.1.4 toAttribute

public Attribute toAttribute()

Returns: an Attribute which contains the type and value of this
definition

5.1.5 getOid

public AttributeType getOid()

Returns: AttributeType representing the oid of the attribute defined
here

5.1.6 getNames

public String[] getNames()

Returns: an array of Strings representing the names by which this
attribute is known. Since this field is optional, this method may
return null.

5.1.7 setNames
public void setNames(String[] names)

Parameters:
   names - an array of Strings representing the names by which this
           attribute is known.

5.1.8 getDescription

public String getDescription()

Returns: a String representing a verbose description of this attribute.
         Since this field is optional, this method may return null.

5.1.9 setDescription

public void setDescription(String description)

Parameters:
   description - a String representing a verbose description of this
                 attribute.

5.1.10 isObsolete

public boolean isObsolete()

Returns: true if this attribute is being phased out; false otherwise;
         default is false

5.1.11 setObsolete

public void setObsolete(boolean isObsolete)

Parameters:
   isObsolete - true if this attribute is being phased out; false
                otherwise;

5.1.12 getSuperior

public AttributeType getSuperior()

Returns: AttributeType representing the oid of an attribute type from
         which this attribute is derived. Since this field is optional, this
         method may return null.

5.1.13 setSuperior

public void setSuperior(AttributeType superior)

Parameters:
   superior - AttributeType of an attribute type from which this
              attribute is derived.

5.1.14 getEquality
public MatchingRuleId getEquality()

Returns: an MatchingRuleId of the equality matching rule to use when evaluating attribute values for selection. Since this field is optional, this method may return null.

5.1.15 setEquality

public void setEquality(MatchingRuleId equality)

Parameters:
equality - a MatchingRuleId of the equality matching rule to use when evaluating attribute values for selection.

5.1.16 getOrdering

public MatchingRuleId getOrdering()

Returns: a MatchingRuleId of the ordering matching rule to use when evaluating attribute values for selection. Since this field is optional, this method may return null.

5.1.17 setOrdering

public void setOrdering(MatchingRuleId ordering)

Returns: an MatchingRuleId of the ordering matching rule to use when evaluating attribute values for selection.

5.1.18 getSubstr

public MatchingRuleId getSubstr()

Returns: a MatchingRuleId of the substring matching rule to use when evaluating attribute values for selection. Since this field is optional, this method may return null.

5.1.19 setSubstr

public void setSubstr(MatchingRuleId substr)

Returns: a MatchingRuleId of the substring matching rule to use when evaluating attribute values for selection. Since this field is optional, this method may return null.

5.1.20 getSyntax

public SyntaxId getSyntax()

Returns: the SyntaxID representing the LDAP syntax used by this attribute. Since this field is optional, this method may return null.

5.1.21 setSyntax
public void setSyntax(SyntaxId syntax)

Parameters:
    syntax - the SyntaxID representing the LDAP syntax used by this 
            attribute.

5.1.22 getSuggestedSyntaxLength

public int getSuggestedSyntaxLength()

Returns: int representing the suggested minimum upper bound on the 
number of characters in value with a string-based syntax, or the 
number of bytes in a value for all other syntaxes; -1 if there is no 
suggested length

5.1.23 setSuggestedSyntaxLength

public void setSuggestedSyntaxLength(int length)

Parameters:
    length - int representing the suggested minimum upper bound on the 
number of characters in value with a string-based syntax, or the 
number of bytes in a value for all other syntaxes; -1 if there is no 
suggested length

5.1.24 isSingleValue

public boolean isSingleValue()

Returns: true if this attribute can contain more than one value; false 
otherwise; default is false

5.1.25 setSingleValue

public void setSingleValue(boolean isSingleValue)

Parameters:
    isSingleValue - true if this attribute can contain more than one 
value; false otherwise;

5.1.26 isCollective

public boolean isCollective()

Returns: true if this attribute is collective; false otherwise; default 
is false; A collective attribute is one whose values are the same for 
each member of an entry collection.

5.1.27 setCollective

public void setCollective(boolean isCollective)

Parameters:
isCollective - true if this attribute is collective; false otherwise; A collective attribute is one whose values are the same for each member of an entry collection.

5.1.28 isNoUserModification

public boolean isNoUserModification()

Returns: true if this attribute can modified by the directory user; false otherwise; default is false.

5.1.29 setNoUserModification

public void setNoUserModification(boolean isNoUserModification)

Parameters:
   isNoUserModification - true if this attribute can modified by the directory user; false otherwise.

5.1.30 getUsage

public AttributeUsage getUsage()

Returns: a Usage representing the intended use of this attribute

5.2 Class org.ietf.ldap.schema.AttributeUsage

public final class AttributeUsage

Provides an enumeration of usage values for an AttributeDefinition.

5.2.1 Fields

public static final AttributeUsage USER_APPLICATIONS

public static final AttributeUsage DIRECTORY_OPERATIONS

public static final AttributeUsage DISTRIBUTED_OPERATIONS

public static final AttributeUsage DSA_OPERATIONS

5.2.2 fromString

public static AttributeUsage fromString(String usageString)

Parameters:
   usageString - String representing a valid usage string; should be one for the following:
       userApplications
directoryOperation
distributedOperation
dSAOperation

Returns: an AttributeUsage corresponding to a string value; null if
input String is not one of the valid usage strings

5.3 Class org.ietf.ldap.schema.MatchingRuleDefinition

public class MatchingRuleDefinition
    extends SchemaDefinition

MatchingRuleDefinition represents the Matching Rule as part of an LDAP schema defined in section 4.2 of RFC 2252. A matching rule allows a set of entries in a DIB to be selected by making particular assertions concerning their attribute values.

5.3.1 Fields

public static final String TYPE

5.3.2 Constructors

public MatchingRuleDefinition(String value)
    throws ProtocolException

Constructs an instance from a String containing terminal symbols and variables as specified in section 4.2 of RFC 2252.

Parameters:
    value - the String containing information to be parsed

Throws:
    ProtocolException - If there is a problem with the data in value

public MatchingRuleDefinition(MatchingRuleId oid, SyntaxId syntax)

Parameters:
    oid - MatchingRuleId representing the oid of this rule
    syntax - SyntaxId representing the oid of the syntax used by this rule

5.3.3 toAttribute

public Attribute toAttribute()

Returns: an Attribute which contains the type and value of this definition

5.3.4 getOid

public MatchingRuleId getOid()

Returns: the MatchingRuleId representing the oid of the rule defined here

5.3.5 getNames

public String[] getNames()
Returns: an array of Strings representing the names by which this rule is known. Since this field is optional, this method may return null.

5.3.6 setNames

```java
public void setNames(String[] names)
```

Parameters:
- names - an array of Strings representing the names by which this rule is known.

5.3.7 getDescription

```java
public String getDescription()
```

Returns: a String representing a verbose description of this rule. Since this field is optional, this method may return null.

5.3.8 setDescription

```java
public void setDescription(String description)
```

Parameters:
- description - a String representing a verbose description of this rule.

5.3.9 isObsolete

```java
public boolean isObsolete()
```

Returns: true if this rule is being phased out; false otherwise; default is false

5.3.10 setObsolete

```java
public void setObsolete(boolean isObsolete)
```

Parameters:
- isObsolete - true if this rule is being phased out; false otherwise

5.3.11 getSyntax

```java
public SyntaxId getSyntax()
```

Returns: OID of the syntax used to express an assertion of each specific type of match

5.3.12 setSyntax

```java
public void setSyntax(SyntaxId syntax)
```

Parameters:
- syntax - the SyntaxID representing the LDAP syntax used by this rule
5.4 Class org.ietf.ldap.schema.ObjectClassDefinition

public class ObjectClassDefinition
    extends SchemaDefinition

ObjectClassDefinition represents the Object Class as part of an LDAP schema defined in section 4.2 of RFC 2252. Object class definitions define the basic set of mandatory and optional attributes that shall be present, and may be present, respectively, in an entry of a give class, and which indicate the kind of object class that is being defined.

5.4.1 Fields

public static final String TYPE

5.4.2 Constructors

public ObjectClassDefinition(String value)
    throws ProtocolException

Constructs an instance from a String containing terminal symbols and variables as specified in section 4.2 of RFC 2252.

Parameters:
    value - the String containing information to be parsed

Throws:
    ProtocolException - If there is a problem with the data in value

public ObjectClassDefinition(ObjectClassId oid)

Parameters:
    oid - ObjectClassId representing the oid of this class

5.4.3 toAttribute

public Attribute toAttribute()

Returns: an Attribute which contains the type and value of this definition

5.4.4 getOid

public ObjectClassId getOid()

Returns: the ObjectClassId of the object class defined here

5.4.5 getNames

public String[] getNames()

Returns: an array of Strings representing the names by which this object class is known Since this field is optional, this method may
return null.

5.4.6 setNames

public void setNames(String[] names)

Parameters:
  names - an array of Strings representing the names by which this
class is known.

5.4.7 getDescription

public String getDescription()

Returns: a String representing a verbose description of this class.
  Since this field is optional, this method may return null.

5.4.8 setDescription

public void setDescription(String description)

Parameters:
  description - a String representing a verbose description of this
class.

5.4.9 isObsoleted

public boolean isObsoleted()

Returns: true if this object class is being phased out; false
  otherwise; default is false

5.4.10 setObsoleted

public void setObsoleted(boolean isObsoleted)

Parameters:
  isObsoleted - true if this class is being phased out; false otherwise;

5.4.11 getSuperiors

public ObjectClassId[] getSuperiors()

Returns: an array of ObjectClassIds representing the oids of classes
  from which this object class is derived. Since this field is
  optional, this method may return null.

5.4.12 setSuperiors

public void setSuperiors(ObjectClassId[] superiors)

Parameters:
  superior - ObjectClassIds representing the oids of classes from which
  this object class is derived.
5.4.13 getObjectClassType

public ObjectClassType getObjectClassType()

Returns: an ObjectClassType representing the kind of object class represented

5.4.14 setObjectClassType

public void setObjectClassType(ObjectClassType type)

Parameters:
  type - an ObjectClassType representing the kind of object class represented

5.4.15 getMusts

public AttributeType[] getMusts()

Returns: an array of AttributeTypes representing the oids of mandatory attribute types an entry of the object class shall contain in addition to the mandatory attribute types of all its superiors. Since this field is optional, this method may return null.

5.4.16 setMusts

public void setMusts(AttributeType[] musts)

Parameters:
  musts - an array of AttributeTypes representing the oids of mandatory attribute types an entry of the object class shall contain in addition to the mandatory attribute types of all its superiors.

5.4.17 getMays

public AttributeType[] getMays()

Returns: an array of AttributeTypes representing the oids of optional attribute types an entry of the object class may contain in addition to the optional attribute types of all its superiors. Since this field is optional, this method may return null.

5.4.18 setMays

public void setMays(AttributeType[] mays)

Parameters:
  mays - an array of AttributeTypes representing the oids of optional attribute types an entry of the object class may contain in addition to the mandatory attribute types of all its superiors.

5.5 Class org.ietf.ldap.schema.ObjectClassId
public class ObjectClassId
    extends SchemaElementId

The ObjectClassId represents the OID and/or name by which a given
ObjectClass is known. If both a name and OID are present in an instance
of ObjectClassId then the instance can considered to assign the name to
the OID in the context of a specific type of server.

5.5.1 Constructors

public ObjectClassId(OID oid)

public ObjectClassId(String name)

public ObjectClassId(OID oid, String name)

5.6 Class org.ietf.ldap.schema.ObjectClassType

public final class ObjectClassType

Provides an enumeration of types of object classes specified in
ObjectClassDefinition.

5.6.1 Fields

public static final ObjectClassType ABSTRACT

public static final ObjectClassType STRUCTURAL

public static final ObjectClassType AUXILIARY

5.6.2 fromString

public static ObjectClassType fromString(String typeString)

Parameters:
typeString - String representing a valid type; should be one for the
          following:
          ABSTRACT
          STRUCTURAL
          AUXILIARY

Returns: an ObjectClassType corresponding to a string value; null if
         input String is not one of the valid type strings

5.7 Class org.ietf.ldap.schema.Schema

public class Schema

Schema is the collection of attribute type definitions, object class
definitions and other information which a server uses to determine how
to match a filter or attribute value assertions (in a compare
operation) against the attributes of an entry, and whether to permit
add and modify operations. Schema is defined formally in RFC 2252.
5.7.1 Constructors

public Schema(Entry entry)
    throws ProtocolException

Constructs an instance from an LDAP directory entry, assumed to be a
subschema entry referred to in the root DS entry attribute
"subschemaentry".

Parameters:
    entry - Entry assumed to contain schema information

Throws:
    ProtocolException - If there is a problem parsing the information in
    the entry.

public Schema(SchemaDefinition[] defArray)

Parameters:
    defArray - an array of SchemaDefinition objects which to store in
    this Schema

5.7.2 toAttributeSet

public AttributeSet toAttributeSet()

Returns: an AttributeSet that contains the attributes of this schema

5.7.3 toEntry

public Entry toEntry(DN dn)

Parameters:
    dn - DN of the new entry

Returns: an Entry containing this schema, suitable for placement in the
directory

5.7.4 addDefinition

public void addDefinition(SchemaDefinition definition)

Adds a SchemaDefinition to the Schema.

Parameters:
    definition - SchemaDefinition to add

5.7.5 getAttributeTypes

public AttributeDefinition[] getAttributeTypes()

Returns: an array of all SchemaDefinitions stored here which are
subclassed as AttributeDefinition
5.7.6 getObjectClasses

public ObjectClassDefinition[] getObjectClasses()

Returns: an array of all SchemaDefinitions stored here which are subclassed as ObjectClassDefinition

5.7.7 getSyntaxes

public SyntaxDefinition[] getSyntaxes()

Returns: an array of all SchemaDefinitions stored here which are subclassed as SyntaxDefinition

5.7.8 getMatchingRules

public MatchingRuleDefinition[] getMatchingRules()

Returns: an array of all SchemaDefinitions stored here which are subclassed as MatchingRuleDefinition

5.7.9 getSchemaDescriptions

public SchemaDescriptionDefinition[] getSchemaDescriptions()

Returns: an array of all SchemaDefinitions stored here which are subclassed as SchemaDescriptionDefinition

5.8 Class org.ietf.ldap.schema.SchemaDefinition

Subclasses:
AttributeDefinition, MatchingRuleDefinition, ObjectClassDefinition,
SchemaDescriptionDefinition, SyntaxDefinition

public abstract class SchemaDefinition

SchemaDefinition represents a definition found in an LDAP schema as defined in RFC 2252.

5.8.1 Constructors

public SchemaDefinition()

5.8.2 toAttribute

public abstract Attribute toAttribute()

Returns: an Attribute which contains the type and value of this definition

5.9 Class org.ietf.ldap.schema.SchemaDescriptionDefinition

public class SchemaDescriptionDefinition
extends SchemaDefinition

SchemaDescriptionDefinition represents the Attribute Types as part of an LDAP schema defined in draft-ietf-schema-ldap-00. Each value of the LDAP schema definition defines one schema and contains the elements needed for an LDAPv3 server to correctly process operations which use its definitions.

The "NAME" field contains optional human-readable labels for the schema. The "OBSOLETE" field is present if the schema is obsolete. The "IMPORTS" field lists the OIDs of other schemas which are to be incorporated by reference into this schema. It is an error to have an attribute type or object class defined in a schema with the same name but a different OID as an attribute type or object class in an imported schema. It is also an error to import from two schema definitions in which there are attribute types or object classes with the same names but different OIDs.

The "CLASSES" field lists the OIDs of object classes defined in this schema. A schema must not contain two object class definitions of the same name but with different OIDs. The "ATTRIBUTES" field lists the OIDs of attribute types defined in this schema. A schema need not contain any object class definitions. A schema must not contain two attribute type definitions of the same name but with different OIDs. The "MATCHING-RULES" field lists the OIDs of matching rules defined in this schema. A schema need not contain any matching rules. The "SYNTAXES" field lists the OIDs of syntaxes defined in this schema.

A schema need not contain any syntaxes.

5.9.1 Fields

public static final String TYPE

5.9.2 Constructors

SchemaDescriptionDefinition

public SchemaDescriptionDefinition(String value)
    throws ProtocolException

Constructs an instance from a String containing terminal symbols and variables as specified in draft-ietf-schema-ldap-00.

Parameters:
    value - the String containing information to be parsed

Throws:
    ProtocolException - If there is a problem with the data in value

5.9.3 SchemaDescriptionDefinition

public SchemaDescriptionDefinition(SchemaDescriptionId oid)
Parameters:
  oid - SchemaDescriptionId representing the oid of this schema

5.9.4 toAttribute

public Attribute toAttribute()

Returns: an Attribute which contains the type and value of this definition

5.9.5 getOid

public SchemaDescriptionId getOid()

Returns: the SchemaDescriptionId representing the oid of the schema defined here

5.9.6 getNames

public String[] getNames()

Returns: an array of Strings representing the names by which this schema is known. Since this field is optional, this method may return null.

5.9.7 setNames

public void setNames(String[] names)

Parameters:
  names - an array of Strings representing the names by which this schema is known.

5.9.8 isObsolete

public boolean isObsolete()

Returns: true if this attribute is being phased out; false otherwise; default is false

5.9.9 setObsolete

public void setObsolete(boolean isObsolete)

Parameters:
  isObsolete - true if this schema is being phased out; false otherwise;

5.9.10 getImports

public SchemaDescriptionId[] getImports()

Returns: an array of SchemaDescriptionIds of other LDAP schemas

Expires 8/99
included by reference into this schema. Since this field is optional, this method may return null.

5.9.11 setImports

public void setImports(SchemaDescriptionId[] imports)

Parameters:
  imports - return an array of SchemaDescriptionIds of other LDAP schemas included by reference into this schema.

5.9.12 getClasses

public ObjectClassId[] getClasses()

Returns: an array of ObjectClassIds representing the oids of object classes included in this LDAP schema. Since this field is optional, this method may return null.

5.9.13 setClasses

public void setClasses(ObjectClassId[] classes)

Parameters:
  classes - an array of ObjectClassIds representing the oids of object classes included in this LDAP schema.

5.9.14 getAttributes

public AttributeType[] getAttributes()

Returns: an array of AttributeTypes of attributes included in this LDAP schema. Since this field is optional, this method may return null.

5.9.15 setAttributes

public void setAttributes(AttributeType[] attributes)

Parameters:
  attributes - an array of AttributeTypes of attributes included in this LDAP schema.

5.9.16 getMatchingRules

public MatchingRuleId[] getMatchingRules()

Returns: an array of MatchingRuleIds representing the oids of matching rules included in this LDAP schema. Since this field is optional, this method may return null.

5.9.17 setMatchingRules

public void setMatchingRules(MatchingRuleId[] rules)
Parameters:
  rules - an array of MatchingRuleIds representing the oids of matching
  rules included in this LDAP schema.

5.9.18 getSyntaxes

public SyntaxId[] getSyntaxes()

Returns: an array of SyntaxIds representing the oids of syntaxes
  included in this LDAP schema. Since this field is optional, this
  method may return null.

5.9.19 setSyntaxes

public void setSyntaxes(SyntaxId[] syntaxes)

Parameters:
  syntaxes - an array of SyntaxIds representing the oids of syntaxes
  included in this LDAP schema.

5.10 Class org.ietf.ldap.schema.SchemaDescriptionId

public class SchemaDescriptionId
  extends SchemaElementId

The SchemaDescriptionId represents the OID and/or name by which a given
SchemaDescription is known. If both a name and OID are present in an
instance of SchemaDescriptionId then the instance can be considered to
assign the name to the OID in the context of a specific type of server.

5.10.1 Constructors

public SchemaDescriptionId(OID oid)

public SchemaDescriptionId(String name)

public SchemaDescriptionId(OID oid, String name)

5.11 Class org.ietf.ldap.schema.SyntaxDefinition

public class SyntaxDefinition
  extends SchemaDefinition

SyntaxDefinition represents the Syntax as part of an LDAP schema
defined in section 4.2 of RFC 2252. A syntax defines what kind of
information is allowed to be stored in an attribute’s values and how
those values behave during searches and other directory operations.

5.11.1 Fields

public static final String TYPE

5.11.2 Constructors
public SyntaxDefinition(String value)
    throws ProtocolException

Constructs an instance from a String containing terminal symbols and variables as specified in section 4.2 of RFC 2252.

Parameters:
    value - the String containing information to be parsed

Throws:
    ProtocolException - If there is a problem with the data in value

SyntaxDefinition

public SyntaxDefinition(SyntaxId oid)

Constructs an instance from specified parameters

Parameters:
    oid - SyntaxId representing the oid of this syntax

5.11.3 toAttribute

public Attribute toAttribute()

Returns: an Attribute which contains the type and value of this definition

5.11.4 getOid

public SyntaxId getOid()

Returns: the SyntaxId of the syntax defined here

5.11.5 getDescription

public String getDescription()

Returns: a String representing a verbose description of this syntax.
    Since this field is optional, this method may return null.

5.11.6 setDescription

public void setDescription(String description)

Parameters:
    return - a String representing a verbose description of this syntax.

5.12 Class org.ietf.ldap.schema.SyntaxId

public class SyntaxId
    extends SchemaElementId

The SyntaxId represents the OID and/or name by which a given Syntax is
known. If both a name and OID are present in an instance of SyntaxId then the instance can considered to assign the name to the OID in the context of a specific type of server. NOTE: the use of a name to describe a syntax ID is not provided for in the specification defined in RFC 2252. This ability is included here for use with directory servers not conforming to the specification.

5.12.1 Constructors

public SyntaxId(OID oid)

public SyntaxId(String name)

public SyntaxId(OID oid, String name)

5.13 Class org.ietf.ldap.schema.SyntaxTable

public class SyntaxTable

Provides routines for converting syntax oids to descriptions and vice versa. Lookups are based on the contents of file syntax.properties in this package and if defined, the contents of syntax.properties in the directory of the program using this class. In this way syntaxes can be extended by the user. The contents of syntax.properties contains on one syntax oid-name combination per line in the form =.

5.13.1 getDescription

public static String getDescription(OID oid)

Retrieves a description given an OID.

Parameters:
   oid - OID of a syntax

Returns: String representing the description of the OID

5.13.2 getOid

public static OID getOid(String description)

Retrieves an OID given a description.

Parameters:
   description - String representing the description of the OID

Returns: oid OID of a syntax

6. Security Considerations

LDAP supports security through protocol-level authentication, using clear-text passwords or other more secure mechanisms. It also supports running over TLS, which provides strong security at the transport layer. This draft supports TLS implementations, via the
SocketHandler interface.

7. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

8. Bibliography


Although there are quite a few classes in the org.ietf.ldap.* packages, relatively few of them are essential for getting started with LDAP. Thus, it is best to take a top down approach in learning the package, starting with the DirectoryClient. After creating this object, interaction with the server is performed by calling methods on this object that correspond to the requests of the LDAP protocol in RFC 2251. The following example is adapted from Appendix A of RFC 1823 - The C LDAP API and should illustrate the main points of the Java version of the API:

```java
import org.ietf.ldap.*;
import org.ietf.ldap.client.*;
import java.util.*;

class SimpleSearch {
    public static void main(String[] args) {
        if (args.length < 4) {
            System.err.println("Usage: java SimpleSearch <host> <port> <dn> <filter> ");
            System.exit(1);
        }
        try {
            /* open an LDAP session */
```
DirectoryClient client =
    new DirectoryClient(args[0], Integer.parseInt(args[1]));
/* authenticate as nobody */
client.bind(null, null);
/* search for entries rooted at dn matching filter,
return all attrs */
String dn = args[2];
String filter = args[3];
SearchResults results =
    client.search(dn, Scope.SUBTREE, filter, null, false);
/* step through each entry returned */
while (results.hasMoreElements()) {
    Entry entry = results.next();
    /* print its name */
    System.out.println("dn: " + entry.getDN());
    Enumeration enum = entry.getAttributes().elements();
    /* print each attribute */
    while (enum.hasMoreElements()) {
        Attribute attr = (Attribute)enum.nextElement();
        System.out.println("attribute: " + attr.getDescription());
        AttributeValue[] values = attr.getValues();
        /* print each value */
        for (int i = 0; i < values.length; i++)
            System.out.println(" value: " + values[i]);
    }
    /* close the LDAP session */
    client.unbind();
} catch (Exception ex) {
    ex.printStackTrace();
}
}

Running the program as follows:

    java SimpleSearch thundercloud 389 "dc=ietf,dc=org" "cn=Thomas Mann"

might give output similar to:

dn: cn=Thomas Mann,dc=ietf,dc=org
attribute: cn
value: Thomas Mann
attribute: sn
value: Mann
attribute: telephonenumber
value: 555-1234
attribute: objectclass
value: top
value: person
attribute: createtimestamp
value: 19980528221642Z
attribute: creatorsname
value: cn=manager,dc=ietf,dc=org
attribute: description
value: writer of some repute
attribute: modifytimestamp
value: 19980812153340Z
attribute: modifiersname
value: cn=manager

For general asynchronous support see ClientConnection and Interaction. These classes support applications that pipeline requests to single servers and process concurrent requests across multiple servers. The Interaction will merge results from multiple requests, while the ClientConnection provides the basic non-blocking interface to submitting requests to a server.