Network Working Group
Request for Comments: 2849
Category: Standards Track

G. Good
iPlanet e-commerce Solutions

June 2000

The LDAP Data Interchange Format (LDIF) - Technical Specification

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

This document describes a file format suitable for describing directory information or modifications made to directory information. The file format, known as LDIF, for LDAP Data Interchange Format, is typically used to import and export directory information between LDAP-based directory servers, or to describe a set of changes which are to be applied to a directory.

Background and Intended Usage

There are a number of situations where a common interchange format is desirable. For example, one might wish to export a copy of the contents of a directory server to a file, move that file to a different machine, and import the contents into a second directory server.

Additionally, by using a well-defined interchange format, development of data import tools from legacy systems is facilitated. A fairly simple set of tools written in awk or perl can, for example, convert a database of personnel information into an LDIF file. This file can then be imported into a directory server, regardless of the internal database representation the target directory server uses.

The LDIF format was originally developed and used in the University of Michigan LDAP implementation. The first use of LDIF was in describing directory entries. Later, the format was expanded to allow representation of changes to directory entries.

Relationship to the application/directory MIME content-type:

The application/directory MIME content-type [1] is a general framework and format for conveying directory information, and is independent of any particular directory service. The LDIF format is a simpler format which is perhaps easier to create, and may also be used, as noted, to describe a set of changes to be applied to a directory.

The key words "MUST", "MUST NOT", "MAY", "SHOULD", and "SHOULD NOT" used in this document are to be interpreted as described in [7].

Definition of the LDAP Data Interchange Format

The LDIF format is used to convey directory information, or a description of a set of changes made to directory entries. An LDIF file consists of a series of records separated by line separators. A record consists of a sequence of lines describing a directory entry, or a sequence of lines describing a set of changes to a directory entry. An LDIF file specifies a set of directory entries, or a set of changes to be applied to directory entries, but not both.

There is a one-to-one correlation between LDAP operations that modify the directory (add, delete, modify, and modrdn), and the types of changerecords described below ("add", "delete", "modify", and "modrdn" or "moddn"). This correspondence is intentional, and permits a straightforward translation from LDIF changerecords to protocol operations.

Formal Syntax Definition of LDIF

The following definition uses the augmented Backus-Naur Form specified in RFC 2234 [2].

```
ldif-file               = ldif-content / ldif-changes

ldif-content            = version-spec 1*(1*SEP ldif-attrval-record)

ldif-changes            = version-spec 1*(1*SEP ldif-change-record)

ldif-attrval-record     = dn-spec SEP 1*attrval-spec

ldif-change-record      = dn-spec SEP *control changerecord

version-spec            = "version:" FILL version-number
```

```
version-number           = 1*DIGIT
                           ; version-number MUST be "1" for the
                           ; LDIF format described in this document.

dn-spec                  = "dn:" (FILL distinguishedName /
                                  ":" FILL base64-distinguishedName)

distinguishedName        = SAFE-STRING
                           ; a distinguished name, as defined in [3]

base64-distinguishedName = BASE64-UTF8-STRING
                           ; a distinguishedName which has been base64
                           ; encoded (see note 10, below)

rdn                      = SAFE-STRING
                           ; a relative distinguished name, defined as
                           ; <name-component> in [3]

base64-rdn               = BASE64-UTF8-STRING
                           ; an rdn which has been base64 encoded (see
                           ; note 10, below)

control                  = "control:" FILL ldap-oid        ; controlType
                           0*1(1*SPACE ("true" / "false")) ; criticality
                           0*1(value-spec)                 ; controlValue
                           SEP
                           ; (See note 9, below)

ldap-oid                 = 1*DIGIT 0*1("." 1*DIGIT)
                           ; An LDAPOID, as defined in [4]

attrval-spec             = AttributeDescription value-spec SEP

value-spec               = ":" (    FILL 0*1(SAFE-STRING) /
                                ":" FILL (BASE64-STRING) /
                                "<" FILL url)
                           ; See notes 7 and 8, below

url                      = <a Uniform Resource Locator,
                            as defined in [6]>
                                   ; (See Note 6, below)

AttributeDescription     = AttributeType [";" options]
                           ; Definition taken from [4]

AttributeType            = ldap-oid / (ALPHA *(attr-type-chars))

options                  = option / (option ";" options)
```

```
option                  = 1*opt-char

attr-type-chars         = ALPHA / DIGIT / "-"

opt-char                = attr-type-chars

changerecord            = "changetype:" FILL
                          (change-add / change-delete /
                           change-modify / change-moddn)

change-add              = "add"                SEP 1*attrval-spec

change-delete           = "delete"             SEP

change-moddn            = ("modrdn" / "moddn") SEP
                           "newrdn:" (    FILL rdn /
                                      ":" FILL base64-rdn) SEP
                           "deleteoldrdn:" FILL ("0" / "1")  SEP
                           0*1("newsuperior:"
                           (    FILL distinguishedName /
                            ":" FILL base64-distinguishedName) SEP)

change-modify           = "modify"             SEP *mod-spec

mod-spec                = ("add:" / "delete:" / "replace:")
                          FILL AttributeDescription SEP
                          *attrval-spec
                          "-" SEP

SPACE                   = %x20
                          ; ASCII SP, space

FILL                    = *SPACE

SEP                     = (CR LF / LF)

CR                      = %x0D
                          ; ASCII CR, carriage return

LF                      = %x0A
                          ; ASCII LF, line feed

ALPHA                   = %x41-5A / %x61-7A
                          ; A-Z / a-z

DIGIT                   = %x30-39
                          ; 0-9
```

```
UTF8-1                  = %x80-BF

UTF8-2                  = %xC0-DF UTF8-1

UTF8-3                  = %xE0-EF 2UTF8-1

UTF8-4                  = %xF0-F7 3UTF8-1

UTF8-5                  = %xF8-FB 4UTF8-1

UTF8-6                  = %xFC-FD 5UTF8-1

SAFE-CHAR               = %x01-09 / %x0B-0C / %x0E-7F
                          ; any value <= 127 decimal except NUL, LF,
                          ; and CR

SAFE-INIT-CHAR          = %x01-09 / %x0B-0C / %x0E-1F /
                          %x21-39 / %x3B / %x3D-7F
                          ; any value <= 127 except NUL, LF, CR,
                          ; SPACE, colon (":", ASCII 58 decimal)
                          ; and less-than ("<" , ASCII 60 decimal)

SAFE-STRING             = [SAFE-INIT-CHAR *SAFE-CHAR]

UTF8-CHAR               = SAFE-CHAR / UTF8-2 / UTF8-3 /
                          UTF8-4 / UTF8-5 / UTF8-6

UTF8-STRING             = *UTF8-CHAR

BASE64-UTF8-STRING      = BASE64-STRING
                          ; MUST be the base64 encoding of a
                          ; UTF8-STRING

BASE64-CHAR             = %x2B / %x2F / %x30-39 / %x3D / %x41-5A /
                          %x61-7A
                          ; +, /, 0-9, =, A-Z, and a-z
                          ; as specified in [5]

BASE64-STRING           = [*(BASE64-CHAR)]
```

Notes on LDIF Syntax

1) For the LDIF format described in this document, the version number MUST be "1". If the version number is absent, implementations MAY choose to interpret the contents as an older LDIF file format, supported by the University of Michigan ldap-3.3 implementation [8].

2) Any non-empty line, including comment lines, in an LDIF file MAY be folded by inserting a line separator (SEP) and a SPACE. Folding MUST NOT occur before the first character of the line. In other words, folding a line into two lines, the first of which is empty, is not permitted. Any line that begins with a single space MUST be treated as a continuation of the previous (non-empty) line. When joining folded lines, exactly one space character at the beginning of each continued line must be discarded. Implementations SHOULD NOT fold lines in the middle of a multi-byte UTF-8 character.

3) Any line that begins with a pound-sign ("#", ASCII 35) is a comment line, and MUST be ignored when parsing an LDIF file.

4) Any dn or rdn that contains characters other than those defined as "SAFE-UTF8-CHAR", or begins with a character other than those defined as "SAFE-INIT-UTF8-CHAR", above, MUST be base-64 encoded. Other values MAY be base-64 encoded. Any value that contains characters other than those defined as "SAFE-CHAR", or begins with a character other than those defined as "SAFE-INIT-CHAR", above, MUST be base-64 encoded. Other values MAY be base-64 encoded.

5) When a zero-length attribute value is to be included directly in an LDIF file, it MUST be represented as AttributeDescription ":" FILL SEP. For example, "seeAlso:" followed by a newline represents a zero-length "seeAlso" attribute value. It is also permissible for the value referred to by a URL to be of zero length.

6) When a URL is specified in an attrval-spec, the following conventions apply:

   a) Implementations SHOULD support the file:// URL format. The contents of the referenced file are to be included verbatim in the interpreted output of the LDIF file.
   b) Implementations MAY support other URL formats. The semantics associated with each supported URL will be documented in an associated Applicability Statement.

7) Distinguished names, relative distinguished names, and attribute values of DirectoryString syntax MUST be valid UTF-8 strings. Implementations that read LDIF MAY interpret files in which these entities are stored in some other character set encoding, but implementations MUST NOT generate LDIF content which does not contain valid UTF-8 data.

8) Values or distinguished names that end with SPACE SHOULD be base-64 encoded.

9) When controls are included in an LDIF file, implementations MAY choose to ignore some or all of them. This may be necessary if the changes described in the LDIF file are being sent on an LDAPv2 connection (LDAPv2 does not support controls), or the particular controls are not supported by the remote server. If the criticality of a control is "true", then the implementation MUST either include the control, or MUST NOT send the operation to a remote server.

10) When an attrval-spec, distinguishedName, or rdn is base64-encoded, the encoding rules specified in [5] are used with the following exceptions: a) The requirement that base64 output streams must be represented as lines of no more than 76 characters is removed. Lines in LDIF files may only be folded according to the folding rules described in note 2, above. b) Base64 strings in [5] may contain characters other than those defined in BASE64-CHAR, and are ignored. LDIF does not permit any extraneous characters, other than those used for line folding.

Examples of LDAP Data Interchange Format

Example 1: An simple LDAP file with two entries

```
version: 1
dn: cn=Barbara Jensen, ou=Product Development, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Barbara Jensen
cn: Barbara J Jensen
cn: Babs Jensen
sn: Jensen
uid: bjensen
telephonenumber: +1 408 555 1212
description: A big sailing fan.

dn: cn=Bjorn Jensen, ou=Accounting, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Bjorn Jensen
sn: Jensen
telephonenumber: +1 408 555 1212
```

Example 2: A file containing an entry with a folded attribute value

```
version: 1
dn:cn=Barbara Jensen, ou=Product Development, dc=airius, dc=com
objectclass:top
objectclass:person
objectclass:organizationalPerson
cn:Barbara Jensen
cn:Barbara J Jensen
cn:Babs Jensen
sn:Jensen
uid:bjensen
telephonenumber:+1 408 555 1212
description:Babs is a big sailing fan, and travels extensively in sea
 rch of perfect sailing conditions.
title:Product Manager, Rod and Reel Division
```

Example 3: A file containing a base-64-encoded value

```
version: 1
dn: cn=Gern Jensen, ou=Product Testing, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Gern Jensen
cn: Gern O Jensen
sn: Jensen
uid: gernj
telephonenumber: +1 408 555 1212
description:: V2hhdCBhIGNhcmVmdWwgcmVhZGVyIHlvdSBhcmUhICBUaGlzIHZhbHVl
IGlzIGJhc2UtNjQtZW5jb2RlZCBiZWNhdXNlIGl0IGhhcyBhIGNvbnRyb2wgY2hhcmFjdG
VyIGluIGl0IChhIENSKS4NICBCeSB0aGUgd2F5LCB5b3Ugc2hvdWxkIHJlYWxseSBnZXQg
b3V0IG1vcmUu
```

Example 4: A file containing an entries with UTF-8-encoded attribute values, including language tags. Comments indicate the contents of UTF-8-encoded attributes and distinguished names.

```
version: 1
dn:: b3U95Za25qWt6YOoLG89QWlyaXVz
# dn:: ou=<JapaneseOU>,o=Airius
objectclass: top
objectclass: organizationalUnit
ou:: 5Za25qWt6YOo
# ou:: <JapaneseOU>
ou;lang-ja:: 5Za25qWt6YOo
# ou;lang-ja:: <JapaneseOU>
ou;lang-ja;phonetic:: 44GI44GE44GO44KH44GG44G2
```

```
# ou;lang-ja:: <JapaneseOU_in_phonetic_representation>
ou;lang-en: Sales
description: Japanese office

dn:: dWlkPXJvZ2FzYXdhcmEsb3U95Za25qWt6YOoLG89QWlyaXVz
# dn:: uid=<uid>,ou=<JapaneseOU>,o=Airius
userpassword: {SHA}O3HSv1MusyL4kTjP+HKI5uxuNoM=
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: rogasawara
mail: rogasawara@airius.co.jp
givenname;lang-ja:: 44Ot44OJ44OL44O8
# givenname;lang-ja:: <JapaneseGivenname>
sn;lang-ja:: 5bCP56yg5Y6f
# sn;lang-ja:: <JapaneseSn>
cn;lang-ja:: 5bCP56yg5Y6fIOODreODieODi+ODvA==
# cn;lang-ja:: <JapaneseCn>
title;lang-ja:: 5Za25qWt6YOoIOmDqOmVtw==
# title;lang-ja:: <JapaneseTitle>
preferredlanguage: ja
givenname:: 44Ot44OJ44OL44O8
# givenname:: <JapaneseGivenname>
sn:: 5bCP56yg5Y6f
# sn:: <JapaneseSn>
cn:: 5bCP56yg5Y6fIOODreODieODi+ODvA==
# cn:: <JapaneseCn>
title:: 5Za25qWt6YOoIOmDqOmVtw==
# title:: <JapaneseTitle>
givenname;lang-ja;phonetic:: 44KN44Gp44Gr44O8
# givenname;lang-ja;phonetic::
<JapaneseGivenname_in_phonetic_representation_kana>
sn;lang-ja;phonetic:: 44GK44GM44GV44KP44KJ
# sn;lang-ja;phonetic:: <JapaneseSn_in_phonetic_representation_kana>
cn;lang-ja;phonetic:: 44GK44GM44GV44KP44KJIOOCjeOBqeOBq+ODvA==
# cn;lang-ja;phonetic:: <JapaneseCn_in_phonetic_representation_kana>
title;lang-ja;phonetic:: 44GI44GE44GO44KH44GG44G2IOOBtuOBoeOCh+OBhg==
# title;lang-ja;phonetic::
# <JapaneseTitle_in_phonetic_representation_kana>
givenname;lang-en: Rodney
sn;lang-en: Ogasawara
cn;lang-en: Rodney Ogasawara
title;lang-en: Sales, Director
```

Example 5: A file containing a reference to an external file

```
version: 1
dn: cn=Horatio Jensen, ou=Product Testing, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Horatio Jensen

cn: Horatio N Jensen
sn: Jensen
uid: hjensen
telephonenumber: +1 408 555 1212
jpegphoto:< file:///usr/local/directory/photos/hjensen.jpg
```

Example 6: A file containing a series of change records and comments

```
version: 1
# Add a new entry
dn: cn=Fiona Jensen, ou=Marketing, dc=airius, dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Fiona Jensen
sn: Jensen
uid: fiona
telephonenumber: +1 408 555 1212
jpegphoto:< file:///usr/local/directory/photos/fiona.jpg

# Delete an existing entry
dn: cn=Robert Jensen, ou=Marketing, dc=airius, dc=com
changetype: delete

# Modify an entry's relative distinguished name
dn: cn=Paul Jensen, ou=Product Development, dc=airius, dc=com
changetype: modrdn
newrdn: cn=Paula Jensen
deleteoldrdn: 1

# Rename an entry and move all of its children to a new location in
# the directory tree (only implemented by LDAPv3 servers).
dn: ou=PD Accountants, ou=Product Development, dc=airius, dc=com
changetype: modrdn
newrdn: ou=Product Development Accountants
deleteoldrdn: 0
newsuperior: ou=Accounting, dc=airius, dc=com
```

```
# Modify an entry: add an additional value to the postaladdress
# attribute, completely delete the description attribute, replace
# the telephonenumber attribute with two values, and delete a specific
# value from the facsimiletelephonenumber attribute
dn: cn=Paula Jensen, ou=Product Development, dc=airius, dc=com
changetype: modify
add: postaladdress
postaladdress: 123 Anystreet $ Sunnyvale, CA $ 94086
-

delete: description
-
replace: telephonenumber
telephonenumber: +1 408 555 1234
telephonenumber: +1 408 555 5678
-
delete: facsimiletelephonenumber
facsimiletelephonenumber: +1 408 555 9876
-

# Modify an entry: replace the postaladdress attribute with an empty
# set of values (which will cause the attribute to be removed), and
# delete the entire description attribute. Note that the first will
# always succeed, while the second will only succeed if at least
# one value for the description attribute is present.
dn: cn=Ingrid Jensen, ou=Product Support, dc=airius, dc=com
changetype: modify
replace: postaladdress
-
delete: description
-
```

Example 7: An LDIF file containing a change record with a control

```
version: 1
# Delete an entry. The operation will attach the LDAPv3
# Tree Delete Control defined in [9]. The criticality
# field is "true" and the controlValue field is
# absent, as required by [9].
dn: ou=Product Development, dc=airius, dc=com
control: 1.2.840.113556.1.4.805 true
changetype: delete
```

Security Considerations

Given typical directory applications, an LDIF file is likely to contain sensitive personal data. Appropriate measures should be taken to protect the privacy of those persons whose data is contained in an LDIF file.

Since ":<" directives can cause external content to be included when processing an LDIF file, one should be cautious of accepting LDIF files from external sources. A "trojan" LDIF file could name a file with sensitive contents and cause it to be included in a directory entry, which a hostile entity could read via LDAP.

LDIF does not provide any method for carrying authentication information with an LDIF file. Users of LDIF files must take care to verify the integrity of an LDIF file received from an external source.

Acknowledgments

The LDAP Interchange Format was developed as part of the University of Michigan LDAP reference implementation, and was developed by Tim Howes, Mark Smith, and Gordon Good. It is based in part upon work supported by the National Science Foundation under Grant No. NCR-9416667.

Members of the IETF LDAP Extensions Working group provided many helpful suggestions. In particular, Hallvard B. Furuseth of the University of Oslo made many significant contributions to this document, including a thorough review and rewrite of the BNF.

References

[1] Howes, T. and M. Smith, "A MIME Content-Type for Directory Information", RFC 2425, September 1998.

[2] Crocker, D., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[3] Wahl, M., Kille, S. and T. Howes, "A String Representation of Distinguished Names", RFC 2253, December 1997.

[4] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, July 1997.

[5] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[6] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.

[7] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[8] The SLAPD and SLURPD Administrators Guide. University of Michigan, April 1996. <URL: http://www.umich.edu/~dirsvcs/ldap/doc/guides/slapd/toc.html>

[9] M. P. Armijo, "Tree Delete Control", Work in Progress.

Author's Address

Gordon Good
iPlanet e-commerce Solutions
150 Network Circle
Mailstop USCA17-201
Santa Clara, CA 95054, USA

Phone: +1 408 276 4351
EMail: ggood@netscape.com

Full Copyright Statement

Copyright (C) The Internet Society (2000). 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.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.