NAME

ldif − LDAP Data Interchange Format

DESCRIPTION

The LDAP Data Interchange Format (LDIF) is used to represent LDAP entries and change records in text form. LDAP tools, such as **ldapadd**(1) and **ldapsearch**(1), read and write LDIF entry records. **ldapmodify**(1) reads LDIF change records.

This manual page provides a basic description of LDIF. A formal specification of LDIF is published in RFC 2849.

ENTRY RECORDS

LDIF entry records are used to represent directory entries. The basic form of an entry record is:

```
dn: <distinguished name>
<attrdesc>: <attrvalue>
<attrdesc>: <attrvalue>
<attrdesc>:: <base64-encoded-value>
<attrdesc>:< <URL>
...
```

The value may be specified as UTF-8 text or as base64 encoded data, or a URI may be provided to the location of the attribute value.

A line may be continued by starting the next line with a single space or tab, e.g.,

```
dn: cn=Barbara J Jensen,dc=exam
 ple,dc=com
```

Lines beginning with a sharp sign ('#') are ignored.

Multiple attribute values are specified on separate lines, e.g.,

```
cn: Barbara J Jensen
cn: Babs Jensen
```

If an value contains a non-printing character, or begins with a space or a colon ':', the <attrtype> is followed by a double colon and the value is encoded in base 64 notation. e.g., the value " begins with a space" would be encoded like this:

```
cn:: IGJlZ2lucyB3aXRoIGEgc3BhY2U=
```

If the attribute value is located in a file, the <attrtype> is followed by a ':<' and a file: URI. e.g., the value contained in the file /tmp/value would be listed like this:

```
cn:< file:///tmp/value
```

Other URI schemes (ftp,http) may be supported as well.

Multiple entries within the same LDIF file are separated by blank lines.

ENTRY RECORD EXAMPLE

Here is an example of an LDIF file containing three entries.

```
dn: cn=Barbara J Jensen,dc=example,dc=com
cn: Barbara J Jensen
cn: Babs Jensen
objectclass: person
description:< file:///tmp/babs
sn: Jensen

dn: cn=Bjorn J Jensen,dc=example,dc=com
cn: Bjorn J Jensen
cn: Bjorn Jensen
objectclass: person
sn: Jensen
```

```
dn: cn=Jennifer J Jensen,dc=example,dc=com
cn: Jennifer J Jensen
cn: Jennifer Jensen
objectclass: person
sn: Jensen
jpegPhoto:: /9j/4AAQSkZJRgABAAAAAQABAAD/2wBDABALD
 A4MChAODQ4SERATGCgaGBYWGDEjJR0oOjM9PDkzODdASFxOQ
 ERXRTc4UG1RV19iZ2hnPk1xeXBkeFxlZ2P/2wBDARESEhgVG
...
```

Note that the description in Barbara Jensen's entry is read from file:///tmp/babs and the jpegPhoto in Jennifer Jensen's entry is encoded using base 64.

CHANGE RECORDS

LDIF change records are used to represent directory change requests. Each change record starts with line indicating the distinguished name of the entry being changed:

```
dn: <distinguishedname>

changetype: <[modify|add|delete|modrdn]>
```

Finally, the change information itself is given, the format of which depends on what kind of change was specified above. For a *changetype* of *modify*, the format is one or more of the following:

```
add: <attributetype>
<attrdesc>: <value1>
<attrdesc>: <value2>
...
–
```

Or, for a replace modification:

```
replace: <attributetype>
<attrdesc>: <value1>
<attrdesc>: <value2>
...
–
```

If no *attributetype* lines are given to replace, the entire attribute is to be deleted (if present).

Or, for a delete modification:

```
delete: <attributetype>
<attrdesc>: <value1>
<attrdesc>: <value2>
...
–
```

If no *attributetype* lines are given to delete, the entire attribute is to be deleted.

For a *changetype* of *add*, the format is:

```
<attrdesc1>: <value1>
<attrdesc1>: <value2>
...
<attrdescN>: <value1>
<attrdescN>: <value2>
```

For a *changetype* of *modrdn* or *moddn*, the format is:

```
newrdn: <newrdn>
deleteoldrdn: 0 | 1
newsuperior: <DN>
```

where a value of 1 for deleteoldrdn means to delete the values forming the old rdn from the entry, and a

value of 0 means to leave the values as non-distinguished attributes in the entry. The newsuperior line is optional and, if present, specifies the new superior to move the entry to.

For a *changetype* of *delete*, no additional information is needed in the record.

Note that attribute values may be presented using base64 or in files as described for entry records. Lines in change records may be continued in the manner described for entry records as well.

CHANGE RECORD EXAMPLE

The following sample LDIF file contains a change record of each type of change.

```
dn: cn=Babs Jensen,dc=example,dc=com
changetype: add
objectclass: person
objectclass: extensibleObject
cn: babs
cn: babs jensen
sn: jensen

dn: cn=Babs Jensen,dc=example,dc=com
changetype: modify
add: givenName
givenName: Barbara
givenName: babs
-
replace: description
description: the fabulous babs
-
delete: sn
sn: jensen
-

dn: cn=Babs Jensen,dc=example,dc=com
changetype: modrdn
newrdn: cn=Barbara J Jensen
deleteoldrdn: 0
newsuperior: ou=People,dc=example,dc=com

dn: cn=Barbara J Jensen,ou=People,dc=example,dc=com
changetype: delete
```

INCLUDE STATEMENT

The LDIF parser has been extended to support an **include** statement for referencing other LDIF files. The **include** statement must be separated from other records by a blank line. The referenced file is specified using a file: URI and all of its contents are incorporated as if they were part of the original LDIF file. As above, other URI schemes may be supported. For example:

```
dn: dc=example,dc=com
objectclass: domain
dc: example

include: file:///tmp/example.com.ldif

dn: dc=example,dc=org
objectclass: domain
dc: example
```

This feature is not part of the LDIF specification in RFC 2849 but is expected to appear in a future revision

of this spec. It is supported by the **ldapadd**(1), **ldapmodify**(1), and **slapadd**(8) commands.

SEE ALSO

ldap(3), **ldapsearch**(1), **ldapadd**(1), **ldapmodify**(1), **slapadd**(8), **slapcat**(8), **slapd−ldif**(5).

"LDAP Data Interchange Format," Good, G., RFC 2849.

ACKNOWLEDGEMENTS

OpenLDAP Software is developed and maintained by The OpenLDAP Project <http://www.openldap.org/>. **OpenLDAP Software** is derived from University of Michigan LDAP 3.3 Release.