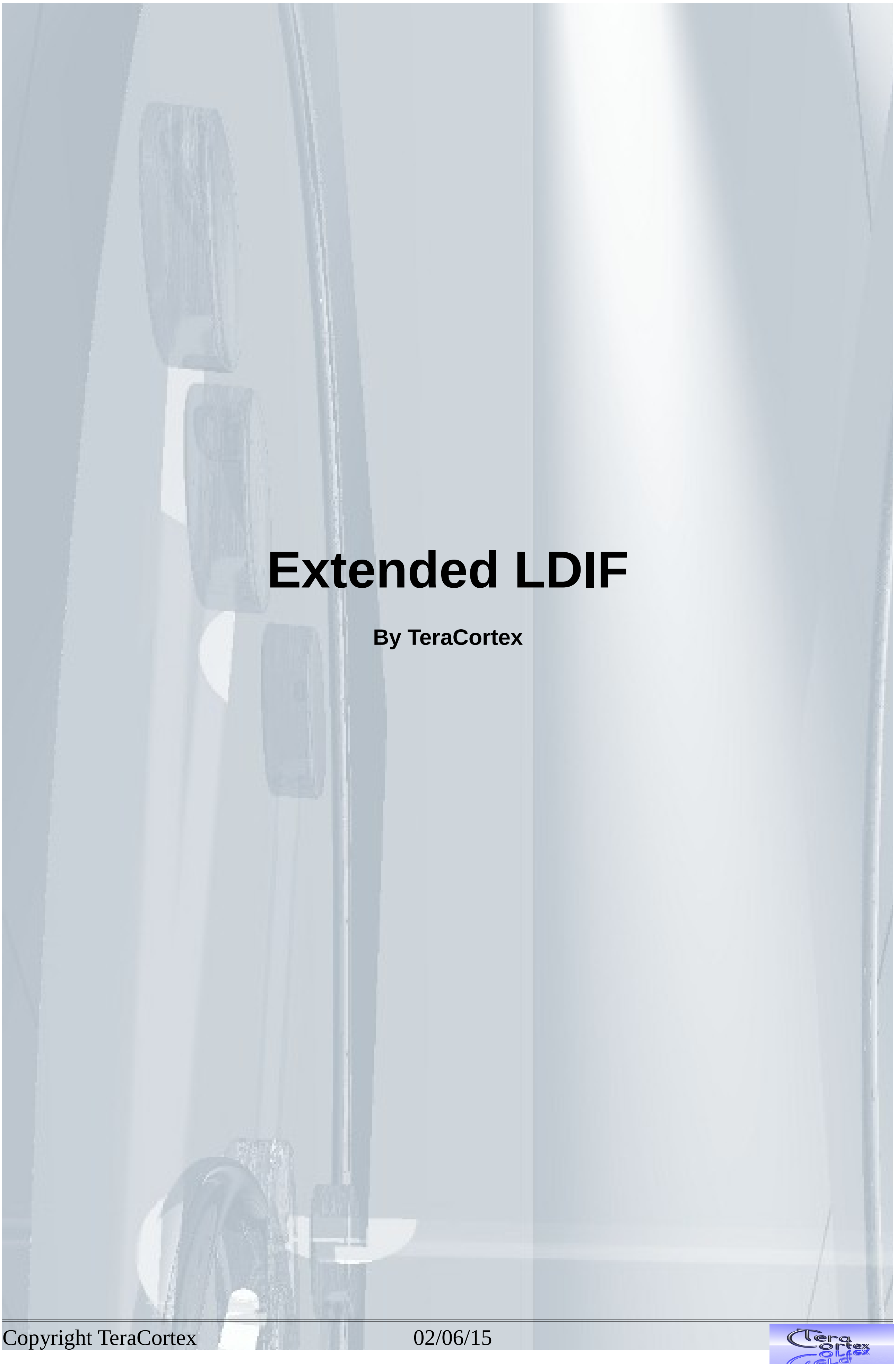

Extended LDIF

By TeraCortex

Table of Contents

Independent Submission Christian Hollstein
INTERNET-DRAFT TeraCortex
Intended Status: Proposed Standard February 2015
Expires 2015/08/26
draft-hollstein-extended-ldif-03.txt

The Extended LDAP Data Interchange Format (EXLDIF)

Status of This Memo

This document is not an Internet Standards Track specification. It is published for examination, experimental implementation, and evaluation. Distribution of this memo is unlimited.

Abstract

LDIF (LDAP Data Interchange Format) has been specified in RFC2849. It covers the LDAP operations ADD, MODIFY, DELETE and MODDN. This document specifies how other LDAP operations can be represented in a text file. Implementations may take such text files to send appropriate LDAP requests to a server and process the responses. This work was inspired by the need for a general configurable LDAP client used in functional and stress testing of LDAP servers.

Copyright Notice

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current.

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."

1. Overview

This document extends the LDIF format [RFC2849] to cover all LDAP operations that can be sent by a LDAP client. For each of the operations BIND, UNBIND, COMPARE, SEARCH, EXTENDED, ABANDON [RFC4511] it specifies the fields needed to encode a complete LDAP request for submission to a server.

1.1. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Extended LDIF Operations

2.1. Relation to existing standards documents

The ABNF of the LDIF syntax in [RFC2849] contains this definition of a change record:

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

This is extended as follows:

```
changerecord = "changetype:" FILL
               (change-add / change-delete /
               change-modify / change-moddn /
               operation-bind / operation-unbind /
               operation-compare / operation-search /
               operation-extended / operation-abandon)
```

Implementations MAY assume change-add if the "changetype:" line is absent. This violation of [RFC2849] is accepted to maintain backward compatibility with the widely distributed OpenLDAP shell API.

Either of the additional operations is covered in the following chapters. In order to maintain backward compatibility the keyword "changetype" is also used for the additional operations despite the fact that with the possible exception of EXTENDED they are no update operations, thus cannot change any data in a LDAP server.

The ABNF forms below make use of ABNF definitions already presented in [RFC2849] and [RFC4515]. Of particular interest are:

- ldif-change-record This is the basic entity containing the definitions for any particular LDAP operation.

- dn-spec This specifies a distinguished name. All additional operations have a distinguished name starting with the keyword "dn:". The distinguished name value may be empty. There is exactly one dn-spec per ldif-change-record.

- control This specifies a control (LDAP V3 only).

- attrval-spec This specifies the OID or name of an attribute possibly with attribute type options and / or attribute value.

- value-spec This specifies an attribute value or the value part of an keyword - value pair. Values may be given in clear text, base64 encoded or by import from an external file.

- ldap-oid This specifies an object identifier in numbers and dot notation.

- filter A search filter as specified in [RFC4515].

- DIGIT A single byte decimal character (0 - 9)

- FILL Zero or more spaces

- SEP Line separator

Beside making use of these definitions Extended LDIF has no backward impact on existing specifications nor does it have any effect on their implementations.

2.2. BIND

```
BIND              = dn-spec SEP *control SEP operation-bind

operation-bind    = "bind"  SEP
                     "version:" FILL 1*DIGIT SEP
                    "authentication:" FILL ("simple" / "sasl") SEP
                     authentication SEP

authentication    = (auth-simple / auth-sasl)

auth-simple       = "passwd" value-spec

auth-sasl         = "mechanism" value-spec SEP
                     "credentials" value-spec
```

2.3. UNBIND

```
UNBIND            = dn-spec SEP *control SEP operation-unbind

operation-unbind  = "unbind" SEP
```

2.4. COMPARE

```
COMPARE            = dn-spec SEP *control SEP operation-compare

operation-compare  = "compare" SEP
                      1*attrval-spec
```

Please note that the value in attrval-spec may be absent. [RFC4511] does not specify how a server shall respond to a compare request with empty attribute value or how a server shall respond to a compare request targeting an attribute with no stored value.

2.5. SEARCH

```
SEARCH             = dn-spec SEP *control SEP operation-search

operation-search   = "scope:" FILL ("base" / "one" / "sub") SEP
                     "deref:" FILL ("never" / "search" /
                     "find" / "always") SEP
                     "sizelimit:" FILL 1*DIGIT SEP
                     "timelimit:" FILL 1*DIGIT SEP
                     "typesonly:" FILL ("true" / "false") SEP
                    "filter:" FILL filter SEP
                     attribute-selector

attribute-selector = *("attribute:" FILL AttributeType SEP)
```

2.6. EXTENDED

```
EXTENDED           = dn-spec SEP *control SEP operation-extended

operation-extended = "extended" SEP
                     "oid:" FILL ldap-oid SEP
                     "value" 0*1(value-spec) SEP
                     "responses:" FILL 1*DIGIT SEP
                      commit-rollback

commit-rollback      = *("commit:" FILL ("true" / "false") SEP
```

The value-spec contains either no value or a single value. The value for the "responses" keyword is an integer. It MUST give the number of responses the client will receive from the server. In most cases there will be just one response. Some implementations of extended requests might call for zero or more than one responses.

In case of an extended request encoding a transaction end the "commit-rollback" directive tells the client to commit or rollback the transaction regardless of the outcome of the requests contained therein.

2.7. ABANDON

```
ABANDON           = dn-spec SEP *control SEP operation-abandon

operation-abandon = "messageId:" FILL 1*DIGIT SEP
```

The value of the "messageId" keyword is an integer. It gives the identifier of a previous LDAP message sent by the client.

3. Response Value Processing

This document does not define any procedural logic in the sense of algorithmic behavior. This means, that a simple implementation can just take the sequence of Extended LDIF records from an input text file, translate them to LDAP protocol level and send them to the wire. However, there are a couple of points that need consideration:

SEARCH and COMPARE operations may yield result data containing valuable information beyond the fact whether the operation was successful or not. In many cases the LDAP client should display the received data or use data content for further decision taking.

EXTENDED operations could yield response values that must be used in subsequent LDAP operations, as is the case in LDAP transactions [RFC5805].

- Message identifiers of LDAP operations may be used in subsequent ABANDON operations to cancel previous requests.

- It may be useful to execute LDAP operations in repetitive loops or execute them conditionally based on the outcome of previous operations or external program input.

These issues cannot be solved within the scope of Extended LDIF. They will be addressed by a different specification.

4. Security Considerations

In addition to the security issues of LDIF files [RFC2849] Extended LDIF may contain authentication information used for BIND operations. This sensitive data MUST NOT be displayed to unauthorized people.

General security considerations [RFC4510], especially those associated with update operations [RFC4511], apply to this extension.

5. IANA Considerations

There are no new object identifiers associated with this specification.

6. Acknowledgments

The author gratefully acknowledges the contributions made by Internet Engineering Task Force participants.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC2849] Good, G., "The LDAP Data Interchange Format (LDIF) - Technical Specification", RFC 2849, June 2000.

[RFC4510] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

[RFC4511] Sermersheim, J., Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

[RFC4515] Smith, M., Ed., Howes, T., "Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters", RFC 4515, June 2006.

7.2. Informative References

[RFC5805] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP) Transactions", RFC 5805, March 2010.

Author's address

Christian Hollstein
TeraCortex
Hopfenbrede 2
D-49179 Ostercappeln

E-Mail: chollstein@teracortex.com
Phone: 0049 / 5473 / 9933
Mobile: 0049 / 160 / 96220958

Appendix A: Changes

01/13/14 Changed "attrsonly" to "typesonly" in chapter 2.5
(Search operation) to comply with [RFC4511]

02/06/15 Introduced forced rollback for extended transaction end requests