

# NET.QUADIUM.LDIF

## LDIF Parser/Writer User Manual

Tim Howe  
vsync@quadium.net

March 27, 2005

## 1 Overview

NET.QUADIUM.LDIF is a library for reading and writing LDIF (Lightweight Directory Interchange Format) records, or files and streams composed of them. This allows Common Lisp software to import data from from or export data to any LDAP (Lightweight Directory Access Protocol) server supporting that format. Currently NET.QUADIUM.LDIF supports importing and exporting of data records, as well as exporting change records.

## 2 Data Types

Currently NET.QUADIUM.LDIF uses simple nested lists for its records. However, this format is not considered stable and users are encouraged to use the opaque accessor methods detailed below, as the structure of the lists may change or the library may in future use, for example, CLOS objects. Accordingly, any operations dealing with change records<sup>1</sup> are considered to be using undocumented functionality and may require changes to work correctly with future releases of NET.QUADIUM.LDIF.

### 2.1 Records

**record-distinguished-name** *record*

Return the distinguished name (DN) of the record as a string.

**record-object-classes** *record*

Return a list of the object classes of the record in the same format as attributes. The **ATTRIBUTE-\*** functions will operate successfully on the elements of this list.

**record-attributes-list** *record*

Returns a list of the attributes of this record.

**record-attributes** *record attribute-name*

Returns a list of the attributes named by **ATTRIBUTE-NAME**.

**record-attribute** *record attribute-name &key (nil-errorp nil)*

Returns a single attribute named by **ATTRIBUTE-NAME**. An error is thrown if **RECORD** contains multiple values for this attribute. The argument **NIL-ERRORP** controls behavior if no attributes match **ATTRIBUTE-NAME**; if false, **NIL** is returned; if true, an error is signalled.

---

<sup>1</sup>Besides those generated by the simplistic converter functions **RECORD->ADD-RECORD** and **RECORD->REPLACE-RECORD**.

**record-attribute-value** *record attribute &key (nil-errorp nil)*

Finds a single attribute as by RECORD-ATTRIBUTE, and returns its value. The attribute's options are returned as a secondary value.

## 2.2 Attributes

**attribute-options** *attribute*

Returns a list of strings denoting ATTRIBUTE's options.

**attribute-value** *attribute*

Returns the string value of ATTRIBUTE.

## 2.3 Change Records

**change-record-change-type** *record*

Returns the change type of the change record RECORD. This will be a symbol in the LDIF package.

**record->add-record** *record*

Returns a change record with the ADD change type designed to create RECORD.

**record->replace-record** *record*

Returns a change record with the MODIFY change type designed to replace all attributes in RECORD.

## 3 Input/Output

**read-record** *&optional (stream \*standard-input\*) (eof-error-p t) eof-value*

Reads a single LDIF data record from STREAM.

**parse-ldif-stream** *stream*

Returns a list of data records read from STREAM up to end-of-file.

**parse-ldif-file** *pathname*

Returns a list of data records read from the file designated by PATHNAME.

**write-record** *record &optional (stream \*standard-output\*)*

Writes the data record RECORD as LDIF to STREAM.

**write-change-record** *record &optional (stream \*standard-output\*)*

Writes the change record RECORD as LDIF to STREAM.