NAME

ldap_dup, ldap_destroy, - Duplicate and destroy LDAP session handles

LIBRARY

OpenLDAP LDAP (libldap, -lldap)

SYNOPSIS

#include <ldap.h>

LDAP *ldap_dup(
LDAP ***old* **);

int ldap_destroy(
LDAP ***old* **);

DESCRIPTION

ldap_dup() duplicates an existing LDAP (**LDAP ***) session handle. The new session handle may be used concurrently with the original session handle. In a threaded environment, different threads may execute concurrent requests on the same connection/session without fear of contamination. Each session handle manages its own private error results.

ldap_destroy() destroys an existing session handle.

The **ldap_dup()** and **ldap_destroy()** functions are used in conjunction with a "thread safe" version of **libldap** to enable operation thread safe API calls, so that a single session may be simultaneously used across multiple threads with consistent error handling.

When a session is created through the use of one of the session creation functions including **ldap_open**(3), **ldap_init**(3), **ldap_initialize**(3) or **ldap_init_fd**(3) an **LDAP *** session handle is returned to the application. The session handle may be shared amongst threads, however the error codes are unique to a session handle. Multiple threads performing different operations using the same session handle will result in inconsistent error codes and return values.

To prevent this confusion, **ldap_dup()** is used duplicate an existing session handle so that multiple threads can share the session, and maintain consistent error information and results.

The message queues for a session are shared between sibling session handles. Results of operations on a sibling session handles are accessible to all the sibling session handles. Applications desiring results associated with a specific operation should provide the appropriate msgid to **ldap_result()**. Applications should avoid calling **ldap_result()** with **LDAP_RES_ANY** as that may "steal" and return

results in the calling thread that another operation in a different thread, using a different session handle, may require to complete.

When **ldap_unbind()** is called on a session handle with siblings, all the siblings become invalid.

Siblings must be destroyed using **ldap_destroy()**. Session handle resources associated with the original (**LDAP ***) will be freed when the last session handle is destroyed or when **ldap_unbind()** is called, if no other session handles currently exist.

ERRORS

If an error occurs, **ldap_dup()** will return NULL and *errno* should be set appropriately. **ldap_destroy()** will directly return the LDAP code associated to the error (or *LDAP_SUCCESS* in case of success); *errno* should be set as well whenever appropriate.

SEE ALSO

ldap_open(3), **ldap_init**(3), **ldap_initialize**(3), **ldap_init_fd**(3), **errno**(3)

ACKNOWLEDGEMENTS

This work is based on the previously proposed **LDAP C API Concurrency Extensions** draft **(**draft-zeilenga-ldap-c-api-concurrency-00.txt**)** effort. **OpenLDAP Software** is developed and maintained by The OpenLDAP Project <http://www.openldap.org/>. **OpenLDAP Software** is derived from the University of Michigan LDAP 3.3 Release.