OLDAP v2.0

by Brainy Data Limited

About LDAP

OLDAP is the Omnis Studio non-visual object implementation of the original Omnis Classic external set of commands. Both Windows and Mac OSX platforms are supported.

The following is a brief description of the various parts of this software.

- The dynamic linked library *oldap.dll* (*oldap.n_xcomp* or *oldap.u_xcomp* on Macintosh) implements the LDAP non-visual object and LDAP related constants that can be accessed from the catalog.
- The example library *ldap.lbs* implements a practical example relating to the use of the LDAP methods and constants.
- The LDAP middle ware (Windows only) consisting of the files LDAPSDK.DLL, LDAPSSL.DLL and LDAPX.DLL. Place these files in the Omnis folder or the windows system folder.

Installing the Software

Place the dynamic linked library in the *XCOMP* folder of your Omnis Studio tree.

On Windows only, place the LDAP middle ware files into the Omnis Studio root folder or the Windows System folder.

You can open the example library directly from the downloaded folder.

Deploying your software

Please refer to the license agreement for rules on deployment.

Documentation

This documentation describes the functionality provided by the external.

Table of Contents

Changes

Version	Description
v2.0	Port of the classic LDAP commands to the new non-visual external object. Please refer to the chapter Designing OLDAP for conversion instructions.

Introduction

Overview

As was detailed in the “Welcome” chapter, the OLDAP software consists of the external DLL, the middle ware DLLs (windows only) and the example library. The example library demonstrates the use of some of the LDAP functionality. You will require access to an LDAP server to use the examples.

Examples

The example library implements a single window that shows you how to connect to an LDAP server, return result sets and query or change LDAP settings.

External component Library

The external DLL implements a single non-visual object (object subtype LDAPSession) that implements all of the LDAP commands as object methods.

For a description of the methods please read the chapter External Component Reference. Before designing your first LDAP object, please read the chapter Designing OLDAP.

Designing OLDAP

This chapter gives a brief description of what is involved to connect to an LDAP server and return a result. An assumption is made that you have a working LDAP server and that you are familiar with the LDAP language. For a detailed description of the supported LDAP methods please read the chapter External Component Reference. The LDAP example library implements a functional interface for connecting to an LDAP server and searching and manipulating entries in an LDAP database. It uses an instance variable to host the non-visual object called ivLDAP.

WARNING: In the example, the instance variable is of type *Object reference* and the $construct and $destruct method call $newref() and $deleteref() to create and destroy the object instance. When using the non-visual LDAP object in your own library and you find you have to pass it as a parameter to Omnis methods, make sure you also use an *Object reference* variable to host the non-visual object.

Converting the old external commands

When converting your old library code, we recommend you use Studio 6 or older on the Macintosh or convert your library's methods on Windows, so you may have loaded both the old external commands DLL and the new non-visual object DLL. If you do not load the old external, the external commands will be displayed as unrecognisable numbers.

Please also note:

- The version two non-visual object is capable of maintaining the LDAP connection ID (CID) in the external instance and consequently the new LDAP methods do not require you to maintain the CID in your code. None of the methods that you call will ask for it. To test if you have a valid connection you can query the object property $session.
- The way that results are returned may have changed for some methods and methods now require you to pass proper constants from the catalog instead of constant names. Make sure you double check what the requirements are for the new LDAP methods. You can refer both to the example code and the external component reference.

A quick example

In this section we show you how to connect to an LDAP server and complete a search.

Check the LDAP middle ware

Before you can connect to your server you must make sure that your LDAP middle ware is functioning correctly. To do this you execute the ivLDAP.$available() method.

Example:

```
Do ivLDAP.$available() Returns #F
If flag true
     ;; LDAP is available
Else
     ;; LDAP is not available - Check the installation
End If
```

Connecting to an LDAP server

Connecting to an LDAP server requires the execution of the two methods ivLDAP.$init(...) and ivLDAP.$simple_bind_s(...). ivLDAP.$init(...) connects to the actual server using the given host name and port, and ivLDAP.$simple_bind_s(...) will establish your access rights using the given DN (Distinguished Name) and password.

When connecting to an LDAP version 3 compliant server you should tell the external to use UTF8 characters for communications by assigning the ivLDAP.$use_utf8 property.

The following example connects to a server and then disconnects using ivLDAP.$unbind_s().

Example:

```
Do ivLDAP.$use_utf8.$assign(kTrue)
Do ivLDAP.$init(”10.0.0.2”,389) Returns err
If not(err)
     Do ivLDAP.$simple_bind_s(”cn=root,dc=brainydata,dc=com”,”Password”) Returns err
End If
If not(err)
     ;; Connection test completed!
Else
     ;; Connection failed!
End If
Do ivLDAP.$unbind_s()
```

Searching an LDAP database

Searching an LDAP database and dissecting the result involves a series of commands. First you call ivLDAP.$search_ext_s(...) with a scope and a filter. Then you use ivLDAP.$first_attribute(...) and ivLDAP$next_attribute(...) together with ivLDAP$get_values(...).

Example:

```
Do ivLDAP.$search_ext_s(”dc=brainydata,dc=com”,kLDAP_SCOPE_BASE,“(objectclass=*)”)
      Returns search_result
Do ivLDAP.$first_entry(search_result) Returns result_entry
While len(result_entry)
      Do ivLDAP.$first_attribute(result_entry,ber) Returns attribute
      While len(attribute)
            Do ivLDAP.$get_values(result_entry,attribute) Returns values
            ;; do something with the values
            Do ivLDAP.$next_attribute(result_entry,ber) Returns attribute
      End While
      Do ivLDAP.$free(ber)
      Do ivLDAP.$next_entry(result_entry) Returns result_entry
End While
Do ivLDAP.$free(search_result)
```

Searching MS Active Directory

The Omnis LDAP external is build against traditional LDAP client software. When connecting to a Microsoft Active Directory server some searches may not work correctly, in particular when searching root level directories using kLDAP_SCOPE_ONELEVEL or kLDAP_SCOPE_SUBTREE.

External Component Reference

The new OLDAP external component reference is currently being worked on. In the meantime, please refer to the example code. When right-clicking the OLDAP object variable, you can display the interface manager which will display all the objects methods and properties.

Constants

LDAP Result Codes

Constants for testing the most common result codes returned by some LDAP methods.

Name	Value	Description
kLDAP_SUCCESS	0	operation was successful.
kLDAP_OPERATIONS_ERROR	1	an general error has occurred. To troubleshoot this type of error, check the server's error logs.
kLDAP_PROTOCOL_ERROR	2	client's request does not comply with the LDAP.
kLDAP_TIMELIMIT_EXCEEDED	3	time limit on a search operation has been exceeded. The time limit is specified in the search request. If you specify no time limit, the server will set one.
kLDAP_SIZELIMIT_EXCEEDED	4	maximum number of search results to return has been exceeded. This limit is specified in the search request. If you specify no size limit, the server will set one.
kLDAP_COMPARE_FALSE	5	returned after an LDAP compare operation is completed. The result indicates that the specified attribute value is not present in the specified entry.
kLDAP_COMPARE_TRUE	6	returned after an LDAP compare operation is completed. The result indicates that the specified attribute value is present in the specified entry.
kLDAP _STRONG_AUTH_NOT_SUPPORTED	7	returned as the result of a bind operation. The server does not recognize or support the specified authentication method.
kLDAP _STRONG_AUTH_REQUIRED	8	a stronger method of authentication is required to perform the operation.
kLDAP_REFERRAL	10	server is referring the client to another LDAP server. Includes a list of LDAP URLs that identify another LDAP server.

kLDAP_ADMINLIMIT_EXCEEDED	11	look-through limit on a search operation has been exceeded.
kLDAP_UNAVAILABLE_CRITICAL_EXTENSION	12	specified control or matching rule is not supported by the server.
kLDAP_CONFIDENTIALITY_REQUIRED	13	confidentiality is required for the operation.
kLDAP_SASL_BIND_IN_PROGRESS	14	used in multi-stage SASL bind operations. The server sends this result code back to the client to indicate that the authentication process has not yet completed.
kLDAP_NO_SUCH_ATTRIBUTE	16	specified attribute does not exist in the entry.
kLDAP_UNDEFINED_TYPE	17	request specifies an undefined attribute type.
kLDAP_INAPPROPRIATE_MATCHING	18	an extensible match filter in a search request contained a matching rule that does not apply to the specified attribute type.
kLDAP_CONSTRAINT_VIOLATION	19	a value in the request does not comply with certain constraints.
kLDAP_TYPE_OR_VALUE_EXISTS	20	request attempted to add an attribute type or value that already exists.
kLDAP_INVALID_SYNTAX	21	request contains invalid syntax.
kLDAP_NO_SUCH_OBJECT	32	server cannot find an entry specified in the request.
kLDAP_ALIAS_PROBLEM	33	alias is invalid.
kLDAP_INVALID_DN_SYNTAX	34	an invalid DN has been specified.
kLDAP_IS_LEAF	35	specified entry is a leaf entry.
kLDAP_ALIAS_DEREF_PROBLEM	36	a problem occurred when dereferencing an alias.
kLDAP_INAPPROPRIATE_AUTH	48	type of credentials are not appropriate for the method of authentication used.
kLDAP_INVALID_CREDENTIALS	49	credentials provided in the request are invalid.
kLDAP_INSUFFICIENT_ACCESS	50	client has insufficient access to perform the operation. Check that the user you are authenticating as has the appropriate permissions.
kLDAP_BUSY	51	server is currently too busy to perform the requested operation.

kLDAP_UNAVAILABLE	52	server is unavailable to perform the requested operation.
kLDAP _UNWILLING_TO_PERFORM	53	server is unwilling to perform the requested operation.
kLDAP_LOOP_DETECT	54	server was unable to perform the requested operation because of an internal loop.
kLDAP _SORT_CONTROL_MISSING	60	server did not receive a required server-side sorting control.
kLDAP_INDEX_RANGE_ERROR	61	search results exceeded the range specified by the requested offsets.
kLDAP_NAMING_VIOLATION	64	request violates the structure of the DIT.
kLDAP _OBJECT_CLASS_VIOLATION	65	request specifies a new entry or a change to an existing entry that does not comply with the server's schema.
kLDAP _NOT_ALLOWED_ON_NONLEA F	66	requested operation is allowed only on entries that do not have child entries (leaf entries as opposed to branch entries).
kLDAP _NOT_ALLOWED_ON_RDN	67	requested operation will affect the RDN of the entry.
kLDAP_ALREADY_EXISTS	68	request is attempting to add an entry that already exists in the directory.
kLDAP _NO_OBJECT_CLASS_MODS	69	request is attempting to modify an object class that should not be modified (for example, a structural object class).
kLDAP_RESULTS_TOO_LARGE	70	results of the request are too large.
kLDAP _AFFECTS_MULTIPLE_DSAS	71	requested operation needs to be performed on multiple servers, where this operation is not permitted.
kLDAP_OTHER	80	an unknown error has occurred.
kLDAP_SERVER_DOWN	81	cannot establish a connection with, or lost the connection to, the LDAP server.
kLDAP_LOCAL_ERROR	82	an error occurred in the LDAP client.
kLDAP_ENCODING_ERROR	83	LDAP client encountered an error when encoding the LDAP request to be sent to the server.
kLDAP_DECODING_ERROR	84	LDAP client encountered an error when decoding the LDAP response received from the server.

kLDAP_TIMEOUT	85	LDAP client timed out while waiting for a response from the server.
kLDAP_AUTH_UNKNOWN	86	an unknown authentication method was specified.
kLDAP_FILTER_ERROR	87	an error occurred when specifying the search filter.
kLDAP_USER_CANCELLED	88	user cancelled the LDAP operation.
kLDAP_PARAM_ERROR	89	an invalid parameter was specified.
kLDAP_NO_MEMORY	90	no memory is available.
kLDAP_CONNECT_ERROR	91	LDAP client cannot establish a connection, or has lost the connection, with the LDAP server.
kLDAP_NOT_SUPPORTED	92	LDAP client is attempting to use functionality that is not supported.
kLDAP _CONTROL_NOT_FOUND	93	a requested LDAP control was not found.
kLDAP _NO_RESULTS_RETURNED	94	no results were returned from the server.
kLDAP _MORE_RESULTS_TO_RETURN	95	there are more results in the chain of results.
kLDAP_CLIENT_LOOP	96	LDAP client detected a loop, for example, when following referrals.
kLDAP _REFERRAL_LIMIT_EXCEEDED	97	the referral hop limit was exceeded.

LDAP Options

Option constants for use with $get_option() or $set_option()

Name	Value	Description
kLDAP_OPT_API_INFO	0	return basic information about the API and the specific implementation that is used.
kLDAP_OPT_DEREF	2	specify alternative rules for following aliases at the server
kLDAP_OPT_SIZELIMIT	3	specify the maximum number of entries that can be returned on a search operation.
kLDAP_OPT_TIMELIMIT	4	specify the number of seconds to wait for search results.
kLDAP_OPT_REFERRALS	8	specify whether the LDAP library automatically follows referrals that are returned by LDAP servers or not.

kLDAP_OPT_RESTART	9	specify whether the library should implicitly restart connections
kLDAP _OPT_PROTOCOL_VERSION	17	sets/gets the protocol version.
kLDAP _OPT_SERVER_CONTROLS	18	sets/gets the server-side controls to be used for all operations.
kLDAP _OPT_CLIENT_CONTROLS	19	sets/gets the client-side controls to be used for all operations.
kLDAP _OPT_API_FEATURE_INFO	21	gets LDAP API feature info
kLDAP_OPT_HOST_NAME	48	sets/gets a space-separated list of hosts to be contacted by the library when trying to establish a connection.
kLDAP_OPT_RESULT_CODE	49	sets/gets the LDAP result code
kLDAP_OPT_ERROR_NUMBER	49	sets/gets the LDAP result code
kLDAP_OPT_ERROR_STRING	50	sets/gets a string containing the error string
kLDAP_OPT_MATCHED_DN	51	sets/gets a string containing the matched DN
kLDAP_OPT_DEBUG_LEVEL	20481	sets/gets the debug level of the client library
kLDAP_OPT_TIMEOUT	20482	sets/gets a timeout value for the synchronous API calls.
kLDAP _OPT_NETWORK_TIMEOUT	20485	sets/gets the network timeout value
kLDAP_OPT_REFERRAL_LIST	20487	sets/gets an array containing the referral URIs
kLDAP_OPT_SESSION_REFCNT		n/a

LDAP Scope

Option constants for use with $search_ext() or $search_ext_s()

Name	Value	Description
kLDAP_SCOPE_DEFAULT	-1	use default scope
kLDAP_SCOPE_BASE	0	A base search limits the search to the base object. The maximum number of objects returned is always one. This search is useful to verify the existence of an object for retrieving group membership.

kLDAP_SCOPE_ONELEVEL	1	A one-level search is restricted to the immediate children of a base object, but excludes the base object itself. This setting can perform a targeted search for immediate child objects of a parent object.
kLDAP_SCOPE_SUBTREE	2	A subtree search (or a deep search) includes all child objects as well as the base object. You can request the LDAP provider to chase referrals to other LDAP directory services, including other directory domains or forests.

NV Object Properties

Name	Type	Description
$debug (v2.0)	Boolean	if true, debug messages are output to a log file
$session (v2.0)	Boolean	returns true if the last call to $init was successful (read-only)
$utf8 (v2.0)	Boolean	specifies preference to use utf8 (will use system os 8bit characters if turned off)

NV Object Methods

$add_att()

Syntax: *OLdapObjRef*.$add_att(Collection,Name,Value)

Version: 2.0

adds an attribute and its value to an attribute collection. See also $set_att(), $get_att() and $get_att_array_item.

Parameter	Description
Collection (char)	Specifies the collection.
Name (char)	Specifies the attribute name.
Value (char)	Specifies the attribute value.
returns (char)	The updated collection string.

$add_ext_s()

Syntax: *OLdapObjRef*.$add_ext_s(DN,Attributes,ServerControls,ClientControls)

Version: 2.0

synchronously adds an entry to the directory using LDAP client or server controls. See also $delete_ext_s(), $modify_ext_s(), $add_att().

Parameter	Description
DN (char)	Specifies the distinguished name of the entry to add, for example "cn=kim".
Attributes (char)	A collection of attributes and values to add with the entry. See $add_att().
ServerControls (char)	Reference to server controls. Specify an empty string if no server controls.
ClientControls (char)	Reference to client controls. Specify an empty string if no client controls.
returns (int)	See LDAP Result Codes.

$available()

Syntax: *OLdapObjRef*.$available()

Version: 2.0

Tests if the low level LDAP functions are available. See also $init().

Parameter	Description

returns	boolean state indicating whether the LDAP functions are available. If this returns kFalse then please refer to the installation instructions.
$compare_s() Syntax: *OLdapObjRef*.$compare_s(DN,Attributes,Value) Version: 2.0 synchronously determines whether a specified entry contains a specified attribute value.	
Parameter	Description
DN (char)	Specifies the distinguished name of the entry to compare.
Attributes (char)	Specifies the name of the attribute to compare.
Value (char)	Specifies the string value of the attribute to compare.
returns (int)	0 (True), 1(False),2 (no such attribute), 3(No such object). For other values see LDAP Result Codes.
$count_entries() Syntax: *OLdapObjRef*.$count_entries(Search Reference) Version: 2.0 Counts the number of messages in the search. See also $result() and $search_ext_s.	
Parameter	Description
Search Reference (char)	Reference to the result chain as returned by $result or by a synchronous search function.
returns (int)	the number of LDAP messages that are of type LDAP_RES_SEARCH_ENTRY.
$count_messages() Syntax: *OLdapObjRef*.$count_messages(Search Reference) Version: 2.0 Counts the number of messages in the search. See also $result and $search_ext_s.	
Parameter	Description
Search Reference (char)	Reference to the result chain as returned by $result or by a synchronous search function.
returns (int)	the number of LDAP messages that are of any type.

$count_references()

Syntax: *OLdapObjRef*.$count_references(Search Reference)

Version: 2.0

Counts the number of references in the search. See also $result and $search_ext_s.

Parameter	Description
Search Reference (char)	Reference to the result chain as returned by $result or by a synchronous search function.
returns (int)	the number of LDAP messages that are of type LDAP_RES_SEARCH_REFERENCE.

$create_persistent_search_cont()

Syntax: *OLdapObjRef*.$create_persistent_search_cont (Type,ChangesOnly,ReturnEntryChanges,IsCritical,Control)

Version: 2.0

Creates and encodes a persistent search control. The control can then be used in $search_ext. See also $search_ext and $free.

Parameter	Description
Type (int)	Integer specifying changes (can be or'd). ADD(Value 1), DELETE (2), MODIFY (4), MODDN(8), ANY(15).
ChangesOnly (bool)	If true, the initial search is only used to establish a result set on the server. No results are returned from this initial search.
ReturnEntryChanges (bool)	If true, a entry change notification control is included with each entry. If 0, entry change notification controls are not included with the entries returned from the server.
IsCritical (bool)	If true, the control is critical to the search operation. If the server does not support persistent searches, the server will return the error kLDAP_UNAVAILABLE_CRITICAL_EXTENSION. If false, the control is not critical to the search operation. Even if the server does not support persistent searches, the search operation is still performed.
Control (char)	Returned reference to the control which is created. When you are finished with this control, you must free the reference.
returns	see LDAP Result Codes.

$delete_ext_s()

Syntax: *OLdapObjRef*.$delete_ext_s(DN[,ServerControls, UserControls])

Version: 2.0

Synchronously deletes the specified entry using LDAP client or server controls. See also $add_ext_s() and $modify_ext_s().

Parameter	Description
DN (int)	Specifies the distinguished name of the entry to delete.
ServerControls (char)	Reference to the server controls. May be empty if no server controls.
UserControls (char)	Reference to the user controls. May be empty if no user controls.
returns	see LDAP Result Codes.

$first_attribute()

Syntax: *OLdapObjRef*.$first_attribute(Reference,BerElement Reference)

Version: 2.0

This returns the name of the first attribute in an entry. See also $next_attribute.

Parameter	Description
Reference (char)	Reference to the entry whose attributes are being read.
BerElement Reference (char)	Reference to the BerElement. Can be used in further $next_attribute calls. This reference must be freed by using $free.
returns (char)	the name of the first attribute in an entry.

$first_entry()

Syntax: *OLdapObjRef*.$first_entry(Search reference)

Version: 2.0

This returns the first entry of message type LDAP_RES_SEARCH_ENTRY from a search result chain. See also $next_entry().

Parameter	Description
Search reference (char)	Reference to the result chain as returned by $result or by a synchronous search function.
returns (char)	character reference to the next entry in the chain or empty if no more entries or failure.

$first_message()

Syntax: *OLdapObjRef*.$first_message(Reference)

Version: 2.0

This returns the first entry of message type LDAP_RES_SEARCH_ENTRY from a search result chain. See also $next_message().

Parameter	Description
Reference (char)	Reference to the result chain as returned by $result() or by a synchronous search function.
returns (char)	a character reference to the first message in the chain, or empty if no more messages or failure. This may be of type LDAP_RES_SEARCH_ENTRY, LDAP_RES_SEARCH_RESULT or LDAP_RES_SEARCH_REFERENCE.

$free()

Syntax: *OLdapObjRef*.$free(Reference)

Version: 2.0

Free a Ber, Control or Search reference. This must be called after the reference is no longer required.

Parameter	Description
Reference (char)	The reference to release.
No return	

$get_att()

Syntax: *OLdapObjRef*.$get_att(Collection,Name)

Version: 2.0

Returns the value of the attribute from an attribute collection. Attribute collections are used for the attribute parameters in the $add_ext_s() and $modify_ext_s() functions. See also $add_ext_s(), $modify_ext_s(), $set_att(), $add_att(), $get_att_array_item().

Parameter	Description
Collection (char)	Specifies the collection.
Name (char)	Specifies the attribute name.
returns	The attribute's value.

$get_att_array_item()

Syntax: *OLdapObjRef*.$get_att_array_item(Collection,Name,Index)

Version: 2.0

Returns the value of the nth attribute from an attribute collection. It is possible for a collection to contain more than one value for the same attribute. When this is the case, you use this command to retrieve the individual values of the same attribute. If the index is out of range, an empty value is returned. Attribute collections are used for the attribute parameters in the $add_ext_s() and $modify_ext_s() functions. See also $add_ext_s(), $modify_ext_s(), $set_att(), $get_att(), $add_att().

Parameter	Description
Collection (char)	Specifies the collection.
Name (char)	Specifies the name.
Index (int)	Specifies the index (from 1).
returns (char)	the value of the nth attribute

$get_dn()

Syntax: *OLdapObjRef*.$get_dn(Reference)

Version: 2.0

Obtains the distinguished name of an entry from a search result chain. See also $first_entry() and $next_entry().

Parameter	Description
Reference (char)	Reference to the chain as returned by $first_entry() or $next_entry().
returns	name of the entry, or empty if failure.

$get_option()

Syntax: *OLdapObjRef*.$get_option(Option)

Version: 2.0

Gets the value of the session-wide specified parameter. See also $set_option(), LDAP Options.

Parameter	Description
Option (int)	Option which may be one of the kLDAP_OPT... constants.
Returns (char or int)	Character or Integer (depending on the option) specifying the current value.

$get_values()

Syntax: *OLdapObjRef*.$get_values(Entry Reference,Attribute)

Version: 2.0

Obtains the string value of a specified attribute from an entry. See also $get_values_len().

Parameter	Description
Entry Reference (char)	Reference to the message chain as returned by $first_entry() or $next_entry().
Attribute (char)	The attribute as returned from $first_attribute(), $next_attribute() or the name of an attribute.
returns (char)	a comma-separated list. For binary data use $get_values_len().

$get_values_len()

Syntax: *OLdapObjRef*.$get_values_len(Entry Reference,Attribute)

Version: 2.0

Obtains the binary value of a specified attribute from an entry. See also $get_values().

Parameter	Description
Entry Reference (char)	Reference to the message chain as returned by $first_entry() or $next_entry().
Attribute (char)	The attribute as returned from $first_attribute(), $next_attribute() or the name of an attribute.
returns (bin)	binary value.

$init()

Syntax: *OLdapObjRef*.$init(Host,Port)

Version: 2.0

Opens the specified port on the host and returns a character string indicating the LDAP reference (empty on failure). You must assign the property $use_utf8 with the appropriate value prior to calling this command. See also $use_utf8, $available(), $unbind_s().

Parameter	Description
Host (char)	Specifies the host.
Port (int)	Specifies the port.
returns (char)	a character reference to the LDAP connection.

$modify_ext_s()

Syntax: *OLdapObjRef*.$modify_ext_s(DN,Mods,ServerControls,ClientControls)

Version: 2.0

Synchronously modifies the specified attributes of an entry on an LDAP server, using LDAP client or server controls. See also $add_ext_s(), $delete_ext_s(), $add_att().

Parameter	Description
DN (char)	Specifies the distinguished name of the entry to modify, for example "cn=kim".
Mods (char)	Collection of attributes and values.
ServerControls (char)	Reference to server controls. Specify an empty string if no server controls.
ClientControls (char)	Reference to client controls. Specify an empty string if no client controls.
returns	See LDAP Result Codes.

$msgid()

Syntax: *OLdapObjRef*.$msgid(Message Reference)

Version: 2.0

Obtains the ID of the message.

Parameter	Description
Message Reference (char)	Reference to the message.
returns	The message ID or -1 if failure.

$msgtype()

Syntax: *OLdapObjRef*.$msgtype(Message Reference)

Version: 2.0

Obtains the type of the message.

Parameter	Description
Message Reference (char)	Reference to the message.
returns	The message type or -1 if failure.

$next_attribute()

Syntax: *OLdapObjRef*.$next_attribute(Reference,BerElement Reference)

Version: 2.0

Returns the name of the next attribute in an entry. See also $first_attribute().

Parameter	Description
Reference (char)	Reference to the entry whose attributes are being read.
BerElement Reference (char)	Reference to the BerElement. Can be used in further $next_attribute() calls. This reference must be freed by calling $free().
returns	the name of the next attribute in an entry, or empty if no more attributes.

$next_entry()

Syntax: *OLdapObjRef*.$next_entry(Reference)

Version: 2.0

This returns the next entry of message type, LDAP_RES_SEARCH_ENTRY from a search result chain. See also $first_entry().

Parameter	Description
Reference (char)	Reference to the chain as returned by $first_entry().
returns	character reference to the next entry in the chain or empty if no more entries or failure.

$next_message()

Syntax: *OLdapObjRef*.$next_message(Message Reference)

Version: 2.0

This returns the next message in the result chain. See also $first_message().

Parameter	Description
Reference (char)	Reference to the message chain as returned by $first_message().
returns	a character reference to the next message in the chain or empty if no more messages or failure. This may be of type LDAP_RES_SEARCH_ENTRY, LDAP_RES_SEARCH_RESULT or LDAP_RES_SEARCH_REFERENCE.

$parse_reference()

Syntax: *OLdapObjRef*.$parse_reference(Message,Referral,ServerControl,FreeRes)

Version: 2.0

Extracts URLs and controls from a Message of type LDAP_RES_SEARCH_REFERENCE. See also $first_message().

Parameter	Description

Message (char)	Specifies the message reference.
Referral (char)	Returned comma-separated string containing alternative LDAP server URLs.
ServerControls (char)	Returned server control references. You must free this reference.
FreeRes (boolean)	Boolean specifying whether you wish to release the resources. If you specify kFalse then you must free the Message.
returns	See LDAP Result Codes

$parse_result()

Syntax: *OLdapObjRef*.$parse_result(Message, ErrorCode, MatchString, ErrorMsg, Referral, ServerControl, FreeRes)

Version: 2.0

Extracts URLs and controls from a Message of type LDAP_RES_SEARCH_REFERENCE.

Parameter	Description
Message (char)	Specifies the message reference.
Error code (int)	Returned error code of last LDAP operation.
MatchString (char)	Returned character string specifying how much of the name in the request was recognised.
ErrorMsg (char)	Returned character string of error message associated with error code.
Referral (char)	Returned comma-separated string containing alternative LDAP server URLs.
ServerControls (char)	Returned server control references. You must free this reference.
FreeRes (boolean)	Boolean specifying whether you wish to release the resources. If you specify kFalse then you must free the Message.
returns	See LDAP Result Codes

$result()

Syntax: *OLdapObjRef*.$result(MsgId,All[,Timeout])

Version: 2.0

Obtains results from a previous asynchronously initiated operation.. See also $count_entries(), $count_references(), $count_messages()

Parameter	Description

MsgId (char)	Specifies the message ID returned. Can be LDAP_RES_UNSOLICITED or LDAP_RES_ANY.
All (char)	Specifies how many messages to be retrieved in a single call to LDAP RESULT. Can be a number or "LDAP_MSG_ONE", "LDAP_MSG_ALL" or "LDAP_MSG_RECEIVED".
Timeout (int)	Specifies how long (in seconds) to wait for the results to be returned.
returns	a character reference to the results of the search. If no results are returned this may be 0 (time out) or -1 (error). Free this reference when you are done with it.

$search_ext()

Syntax: *OLdapObjRef*.$search_ext(Base, Scope [,Filter, Attrs, ServerControls, UserControls, AttrsOnly, Timeout])

Version: 2.0

Asynchronously searches the directory using LDAP client or server controls. See also $search_ext_s()

Parameter	Description
Base (char)	Specifies the distinguished name of the entry from which to start the search.
Scope (char)	Specifies the scope of the search and can be kLDAP_SCOPE_BASE, kLDAP_SCOPE_ONELEVEL or kLDAP_SCOPE_SUBTREE.
Filter (char)	Filter string if none is specified then the default filter ("objectclass=*") is used.
ServerControls (char)	Reference to server controls, if any.
UserControls (char)	Reference to user controls, if any.
Attrs (char)	Comma-separated list specifies which attributes to return.
AttrsOnly (char)	Specifies whether to return both attributes & values (default) or only attributes.
Timeout (int)	Specifies the time out in seconds. Default is 10 seconds.
Returns	-1 if failure otherwise the message id of the operation.

$search_ext_s()

Syntax: *OLdapObjRef*.$search_ext_s(Base, Scope [,Filter, Attrs, ServerControls, UserControls, AttrsOnly, Timeout])

Version: 2.0

Synchronously searches the directory using LDAP client or server controls. See also $search_ext(), $count_entries(), $count_messages(), $count_references().

Parameter	Description
Base (char)	Specifies the distinguished name of the entry from which to start the search.
Scope (char)	Specifies the scope of the search and can be kLDAP_SCOPE_BASE, kLDAP_SCOPE_ONELEVEL or kLDAP_SCOPE_SUBTREE.
Filter (char)	Filter string if none is specified then the default filter ("objectclass=*") is used.
ServerControls (char)	Reference to server controls, if any.
UserControls (char)	Reference to user controls, if any.
Attrs (char)	Comma-separated list specifies which attributes to return.
AttrsOnly (char)	Specifies whether to return both attributes & values (default) or only attributes.
Timeout (int)	Specifies the time out in seconds. Default is 10 seconds.
Returns	-1 if failure otherwise the message id of the operation.

$set_att()

Syntax: *OLdapObjRef*.$set_att(Collection,Name,Value)

Version: 2.0

Sets the value of an existing attribute in the attribute collection. Attribute collections are used for the attribute parameters in the $add_ext_s() and $modify_ext_s() functions. See also $add_ext_s(), $modify_ext_s(), $add_att(), $get_att(), $get_att_array_item().

Parameter	Description
Collection (char)	The collection of attributes.
Name (char)	Name of the attribute.
Value (char)	The value for the attribute.
returns	The updated collection is returned.

$set_option()

Syntax: *OLdapObjRef*.$set_option(Option,Value)

Version: 2.0

Sets the value of the session-wide parameters. See also $get_option().

Parameter	Description
Option (int)	One of the kLDAP_OPT... constants.
Value (char/int)	The new value.
returns	kLDAP_SUCCESS or

$simple_bind_s()

Syntax: *OLdapObjRef*.$simple_bind_s([LoginDN,Password])

Version: 2.0

Synchronously authenticates the specified client to the LDAP server using a distinguished name and password. See also $init(), $unbind_s().

Parameter	Description
LoginDN (char)	Distinguished name of the entry who is authenticating. For an anonymous authentication, do not specify this parameter.
Password (char)	Client's password. For anonymous authentication, do not specify this parameter.
returns	See LDAP Result Codes

$unbind_s()

Syntax: *OLdapObjRef*.$unbind_s()

Version: 2.0

Unbinds from the directory, closes the connection. See also $init(), $simple_bind_s().

Parameter	Description
returns	See LDAP Result Codes