Log Analysis

Error Codes

Indicated as `Err=##`

Non-Error Results Codes

The follow result codes (called “non-error” result codes) do not indicate an error condition:

```
success (0)
compareFalse (5)
compareTrue (6)
referral (10)
saslBindInProgress (14)
```

The *success*, *compareTrue*, and *compareFalse* result codes indicate successful completion (and, hence, are referred to as “successful” result codes).

The *referral* and *saslBindInProgress* result codes indicate the client needs to take additional action to complete the operation.

Standard LDAP Error Codes

Error/ Data	Text	Description
0	LDAP_SUCCESS	Indicates the requested client operation completed successfully.
2	LDAP_PROTOCOL_ERROR	Indicates that the server has received an invalid or malformed request from the client.
3	LDAP_TIMELIMIT_EXCEEDED	Indicates that the operation's time limit specified by either the client or the server has been exceeded. On search operations, incomplete results are returned.
4	LDAP_SIZELIMIT_EXCEEDED	Indicates that in a search operation, the size limit specified by the client or the server has been exceeded. Incomplete results are returned.
5	LDAP_COMPARE_FALSE	Does not indicate an error condition. Indicates that the results of a compare operation are false.
6	LDAP_COMPARE_TRUE	Does not indicate an error condition. Indicates that the results of a compare operation are true.
7	LDAP_AUTH_METHOD_NOT_SUPPORTED	Indicates that during a bind operation the client requested an authentication method not supported by the LDAP server.
8	LDAP_STRONG_AUTH_REQUIRED	Indicates one of the following: In bind requests, the LDAP server accepts only strong authentication. In a client request, the client requested an operation such as delete that requires strong authentication. In an unsolicited notice of disconnection, the LDAP server discovers the security protecting the

		communication between the client and server has unexpectedly failed or been compromised.
9		Reserved.
10	LDAP_REFERRAL	Does not indicate an error condition. In LDAPv3, indicates that the server does not hold the target entry of the request, but that the servers in the referral field may.
11	LDAP_ADMINLIMIT_EXCEEDED	Indicates that an LDAP server limit set by an administrative authority has been exceeded.
12	LDAP_UNAVAILABLE_CRITICAL_EXTENSION	Indicates that the LDAP server was unable to satisfy a request because one or more critical extensions were not available. Either the server does not support the control or the control is not appropriate for the operation type.
13	LDAP_CONFIDENTIALITY_REQUIRED	Indicates that the session is not protected by a protocol such as Transport Layer Security (TLS), which provides session confidentiality.
14	LDAP_SASL_BIND_IN_PROGRESS	Does not indicate an error condition, but indicates that the server is ready for the next step in the process. The client must send the server the same SASL mechanism to continue the process.
15		Not used.
16	LDAP_NO_SUCH_ATTRIBUTE	Indicates that the attribute specified in the modify or compare operation does not exist in the entry.
17	LDAP_UNDEFINED_TYPE	Indicates that the attribute specified in the modify or add operation does not exist in the LDAP server's schema.
18	LDAP_INAPPROPRIATE_MATCHING	Indicates that the matching rule specified in the search filter does not match a rule defined for the attribute's syntax.
19	LDAP_CONSTRAINT_VIOLATION	Indicates that the attribute value specified in a modify, add, or modify DN operation violates constraints placed on the attribute. The constraint can be one of size or content (string only, no binary).
20	LDAP_TYPE_OR_VALUE_EXISTS	Indicates that the attribute value specified in a modify or add operation already exists as a value for that attribute.
21	LDAP_INVALID_SYNTAX	Indicates that the attribute value specified in an add, compare, or modify operation is an unrecognized or invalid syntax for the attribute.
22-31		Not used.

32	LDAP_NO_SUCH_OBJECT	Indicates the target object cannot be found. This code is not returned on following operations: Search operations that find the search base but cannot find any entries that match the search filter. Bind operations.
33	LDAP_ALIAS_PROBLEM	Indicates that an error occurred when an alias was dereferenced.
34	LDAP_INVALID_DN_SYNTAX	Indicates that the syntax of the DN is incorrect. (If the DN syntax is correct, but the LDAP server's structure rules do not permit the operation, the server returns code 53: LDAP_UNWILLING_TO_PERFORM.)
35	LDAP_IS_LEAF	Indicates that the specified operation cannot be performed on a leaf entry. (This code is not currently in the LDAP specifications, but is reserved for this constant.)
36	LDAP_ALIAS_DEREF_PROBLEM	Indicates that during a search operation, either the client does not have access rights to read the aliased object's name or dereferencing is not allowed.
37-47		Not used.
48	LDAP_INAPPROPRIATE_AUTH	Indicates that during a bind operation, the client is attempting to use an authentication method that the client cannot use correctly. For example, either of the following cause this error: The client returns simple credentials when strong credentials are required...OR...The client returns a DN and a password for a simple bind when the entry does not have a password defined.
49	LDAP_INVALID_CREDENTIALS	Indicates that during a bind operation one of the following occurred: The client passed either an incorrect DN or password, or the password is incorrect because it has expired, intruder detection has locked the account, or another similar reason. See the data code for more information.
49 / 52e	AD_INVALID CREDENTIALS	Indicates an Active Directory (AD) AcceptSecurityContext error, which is returned when the username is valid but the combination of password and user credential is invalid. This is the AD equivalent of LDAP error code 49.
49 / 525	USER NOT FOUND	Indicates an Active Directory (AD) AcceptSecurityContext data error that is returned when the username is invalid.
49 / 530	NOT_PERMITTED_TO_LOGON_AT_THIS_TIME	Indicates an Active Directory (AD) AcceptSecurityContext data error that is logon

		failure caused because the user is not permitted to log on at this time. Returns only when presented with a valid username and valid password credential.
49 / 531	RESTRICTED_TO_SPECIFIC_MACHINES	Indicates an Active Directory (AD) AcceptSecurityContext data error that is logon failure caused because the user is not permitted to log on from this computer. Returns only when presented with a valid username and valid password credential.
49 / 532	PASSWORD_EXPIRED	Indicates an Active Directory (AD) AcceptSecurityContext data error that is a logon failure. The specified account password has expired. Returns only when presented with valid username and password credential.
49 / 533	ACCOUNT_DISABLED	Indicates an Active Directory (AD) AcceptSecurityContext data error that is a logon failure. The account is currently disabled. Returns only when presented with valid username and password credential.
49 / 568	ERROR_TOO_MANY_CONTEXT_IDS	Indicates that during a log-on attempt, the user's security context accumulated too many security IDs. This is an issue with the specific LDAP user object/account which should be investigated by the LDAP administrator.
49 / 701	ACCOUNT_EXPIRED	Indicates an Active Directory (AD) AcceptSecurityContext data error that is a logon failure. The user's account has expired. Returns only when presented with valid username and password credential.
49 / 773	USER MUST RESET PASSWORD	Indicates an Active Directory (AD) AcceptSecurityContext data error. The user's password must be changed before logging on the first time. Returns only when presented with valid user-name and password credential.
50	LDAP_INSUFFICIENT_ACCESS	Indicates that the caller does not have sufficient rights to perform the requested operation.
51	LDAP_BUSY	Indicates that the LDAP server is too busy to process the client request at this time but if the client waits and resubmits the request, the server may be able to process it then.
52	LDAP_UNAVAILABLE	Indicates that the LDAP server cannot process the client's bind request, usually because it is shutting down.
52e	AD_INVALID CREDENTIALS	Indicates an Active Directory (AD) AcceptSecurityContext error, which is returned

		when the username is valid but the combination of password and user credential is invalid. This is the AD equivalent of LDAP error code 49: LDAP_INVALID_CREDENTIALS.
53	LDAP_UNWILLING_TO_PERFORM	Indicates that the LDAP server cannot process the request because of server-defined restrictions. This error is returned for the following reasons: The add entry request violates the server's structure rules...OR...The modify attribute request specifies attributes that users cannot modify...OR...Password restrictions prevent the action...OR...Connection restrictions prevent the action.
54	LDAP_LOOP_DETECT	Indicates that the client discovered an alias or referral loop, and is thus unable to complete this request.
55-63		Not used.
64	LDAP_NAMING_VIOLATION	Indicates that the add or modify DN operation violates the schema's structure rules. For example, The request places the entry subordinate to an alias. The request places the entry subordinate to a container that is forbidden by the containment rules. The RDN for the entry uses a forbidden attribute type.
65	LDAP_OBJECT_CLASS_VIOLATION	Indicates that the add, modify, or modify DN operation violates the object class rules for the entry. For example, the following types of request return this error: The add or modify operation tries to add an entry without a value for a required attribute. The add or modify operation tries to add an entry with a value for an attribute which the class definition does not contain. The modify operation tries to remove a required attribute without removing the auxiliary class that defines the attribute as required.
66	LDAP_NOT_ALLOWED_ON_NONLEAF	Indicates that the requested operation is permitted only on leaf entries. For example, the following types of requests return this error: The client requests a delete operation on a parent entry. The client request a modify DN operation on a parent entry.
67	LDAP_NOT_ALLOWED_ON_RDN	Indicates that the modify operation attempted to remove an attribute value that forms the entry's relative distinguished name.
68	LDAP_ALREADY_EXISTS	Indicates that the add operation attempted to add an entry that already exists, or that the

		modify operation attempted to rename an entry to the name of an entry that already exists.
69	LDAP_NO_OBJECT_CLASS_MODS	Indicates that the modify operation attempted to modify the structure rules of an object class.
70	LDAP_RESULTS_TOO_LARGE	Reserved for CLDAP.
71	LDAP_AFFECTS_MULTIPLE_DSAS	Indicates that the modify DN operation moves the entry from one LDAP server to another and requires more than one LDAP server.
72-79		Not used.
80	LDAP_OTHER	Indicates an unknown error condition. This is the default value for NDS error codes which do not map to other LDAP error codes.

A list of error result codes can be found here:

http://www.openldap.org/doc/admin24/appendix-common-errors.html
http://www.openldap.org/doc/admin24/appendix-common-errors.html#Other Errors
http://www.openldap.org/doc/admin24/appendix-ldap-result-codes.html#Non-Error

Gotchas!

Checklist

The following checklist can help track down the cause of a problem. Please try to use if before requesting support.

- Use the slaptest tool to verify configurations before starting slapd
- Verify that slapd is listening to the specified port(s) (389 and 636, generally) before trying the ldapsearch
- Can you issue an ldapsearch?
- If not, have you enabled complex ACLs without fully understanding them?
- Do you have a system wide LDAP setting pointing to the wrong LDAP Directory?
- Are you using TLS?
- Have your certificates expired?

OpenLDAP Bugs

Sometimes you may encounter an actual OpenLDAP bug, in which case please notify Symas support (support@symas.com). You may visit the OpenLDAP project's Issue Tracking System (http://www.openldap.org/its/) to view existing issues requiring additional development. Please note:

- Bugs in historic versions of OpenLDAP will not be considered
- Bugs in released versions that are no longer present in the Git master branch, either because they have been fixed or because they no longer apply, will not be considered
- Bugs in distributions of OpenLDAP software that are not related to the software as provided by OpenLDAP will not be considered; in those cases please refer to the distributor.

Bug requests require specific information as defined in http://www.openldap.org/faq/data/cache/59.html and you may be asked by Symas Support to provide this information.

Easy-to-Fix Issues

Size Limit Problems

A failure to get the results expected, or, replication does not seem to be completing, check for sizeLimitExceeded (4) error codes. These indicate the user or replication user has a sizelimit set somewhere.

Bind Failures

This can indicate invalid or erroneous credentials. Check for InvalidCredentials (49) error codes. Look for typos in the DN or password.

Command Perform Failures

Attempting to write to a read-only directory will produce the UnwillingToPerform (53) error code. Make sure the referral reference and chaining overlay (if used) are in place and configured correctly.

No Response

Check the logs for Busy (51) error codes which could indicate a hung process, insufficient memory or processor availability, etc.

No Results

Often when searching or attempting to modify an object you will encounter the NoSuchObject (32) error. This indicates the object was not found in the database. Check for spelling errors and correct DN or CN syntax. In the event these are correct, this indicates the object has been deleted from the database or did not exist originally.

Unwilling to Perform

When attempting to delete content from the cn=config database, frequently the response will be UnwillingtoPerform(80). This is expected behavior.

TLS Failures

The “TLS init def ctx failed: -1” or “ldap_sasl_interactive_bind_s failed (-1)” errors indicate slapd is unable to find or access the certificate files. Be sure to check the paths for typos and set the file permissions to allow the slapd user access.

The “Error, ldap_start_tls failed (-11)” error either indicates an expired cert or the FQDN used in creating the new cert does not match the FQDN the Consumer servers are looking for (see “provider” in slapd configuration syncrepl stanzas). After replacing the expired or erroneous certificate, restart slapd on all servers. Depending on the latency replication should bring the Consumer servers back into sync fairly quickly.

Dropped Connections

The “connection_read(15): no connection!” occurs when a client drops the connection with the ldap server without issuing the prerequisite unbind command. This can be caused by network failures, firewall or load-balancer issues or problems with the client, itself.

Too Many Open Files

The “accept(9) failed errno=24” occurs when the nofile setting is too low. This can happen during high volume periods such as the first day of classes on a campus or end-of-month audits and payroll processing. To resolve this, increase the nofiles limit.

Low-Impact or Non-Issues

The logs will often contain information that alerts you to an issue that may or may not require any intervention on your part. Some examples with their explanations are included below.

```
conn=14986577 op=559643 do_abandon: bad msgid 0
```

A client sent an "abandon" request with a bad message ID (ID < 0). See https://ldapwiki.com/wiki/Abandon%20Request. This would generally indicate a bug in whatever client is sending the abandon request, but is not problematic for the server.

```
conn=24294246 op=4119410 do_extended: unsupported operation
"1.1.3.6.1.4.1.3830.1.1.7"
```

A client initiated an extended operation requesting OID 1.1.3.6.1.4.1.3830.1.1.7. OpenLDAP is unaware of what this operation is. Often it isa private custom MS AD extension, etc. It is quite common for clients to query LDAP servers to see if they have various extensions enabled and is not indicative of a problem.

```
conn=53742377 deferring operation: binding
```

The bind operation for this connection was deferred as the server was currently busy processing other operations for this connection. This is very common. This generally happens on a connection that sends many operations at once. You can examine surrounding lines in the log to get more data about the bind attempt. https://www.openldap.org/lists/openldap-software/200704/msg00146.html contains a good example.

`connection_read(32): no connection!`

This is very common and indicates a misbehaving client that is failing to correctly follow the LDAP protocol and send an unbind request prior to disconnecting. The server will gracefully clean up the connection.

Comments in slapd.conf

On a side point. Be careful with the use of comments within slapd.conf. Lines beginning with '#' are ignored and assumed to be comments. However, if a line begins with white space, it is considered a continuation of the previous line. Continuation lines are unwrapped before comment processing is applied.

Thus sometimes unexpected results occur. e.g.

```
access to dn.base=""
attrs=supportedSASLMechanisms,namingContexts,subschemaSubentry,objectClass,entry
        by domain.subtree="example.com" read
        by peername.ip="127.0.0.1" read
#       by peername.ip="112.123.123.12" read
        by peername.ip="112.123.123.13" read
        by * none
```

You might think this only removes 112.123.123.12. However, because the following lines all begin with whitespace, this comments out all entries to the end of the stanza, until there is a blank line. Problem! So, if you want to remove an item you have three options:

1. Delete the unwanted line completely.
2. Don't indent the following line. Use no whitespace at all at the start of the line.
3. Follow the commented line with a blank line (one that contains no whitespace).

```
access to dn.base=""
attrs=supportedSASLMechanisms,namingContexts,subschemaSubentry,objectClass,entry
        by domain.subtree="example.com" read
        by peername.ip="127.0.0.1" read
#       by peername.ip="112.123.123.12" read
        by peername.ip="112.123.123.13" read
        by peername.ip="112.123.123.14" read
        by * none
```

When in doubt, open a ticket with Symas!