LDAP Protocol

Teldat-Dm 790-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1 Introduction

1.1 LDAP Protocol: Introduction

LDAP (short for Lightweight Directory Access Protocol) is an application layer protocol that allows you to access a logical distributed directory service in order to locate different kinds of information in a network. Being a standard protocol, LDAP provides an extensive architecture that all applications can use for storing and managing the information that needs to be available in a distributed systems and services environment. Examples of the types of stored information you might find include: system user IDs and their associated passwords, permissions, public certificates, data encryption and security keys, the employee phone book, mail aliases, network resource location information, external client contact information, etc. The advantage of keeping all of this information together in one place is obvious: data is easier to maintain, update and synchronize, thus avoiding inconsistencies.

LDAP is optimized for intensive read operations from varied locations and platforms, but in cases where the data is rarely updated.

LDAP derives from the X.500 standard, though it is a simplified version better suited to meet the needs of users. Unlike X.500, LDAP is based on TCP/IP.

In short, LDAP is an access protocol linked to a set of network data. LDAP allows users to:

- Connect to the LDAP directory: establish a session with the LDAP server.
- Disconnect from the directory: close a previously established session.
- Search for information in the directory.
- Compare information.
- Add entries
- Update entries
- Delete entries

Version 3 of LDAP also provides encryption (SSl, etc.) and authentication mechanisms to allow secure access to information stored in the base.

1.2 LDAP Directory

A **directory** is a repository of information about certain objects arranged in a particular way to provide details on each object. It is a specialized database, also known as a **data repository**, where ordered information is stored in a way that allows applications to locate those resources with the necessary characteristics for a specific task. Directories have several trademark characteristics that differentiate them from general-purpose relational databases:

- Given that they are subject to a high volume of data read requests, directories are built optimized for these read operations. Conversely, general-purpose databases are typically used by applications that must also perform a large number of write operations (data updates).
- Directories do not have to support transactions, i.e., operations that must be treated as a group of actions to be carried out in a way that gives an “all-or-nothing” result depending on whether all the actions involved in the transaction were carried out successfully. Databases do need to support transactions.
- When performing update operations in a directory, two entries are allowed to contain the same or inconsistent information for a short period of time. This is because write operations are occasional and such anomalies are considered acceptable.
- Most databases support a very powerful standard data access methodology called *Structured Query Language* (SQL), which allows quite complex read and update operations at the expense of increasing the size and complexity of the application. LDAP directories use an optimized and simplified protocol that can be implemented in small and relatively simple applications.

1.2.1 LDAP directory structure

Each of the entries in the directory must be uniquely and unambiguously identified, and so are assigned a name, referred to as **Distinguished Name** (**DN**). Each entry's DN is constructed based on the structure where the LDAP directory is organized: LDAP directories store information hierarchically in a tree structure known as the **Directory Information Tree** (**DIT**). Each node in the tree is an entry or **Directory Service Entry** (**DSE**), and to obtain its DN, this individual node is read from, recursively through the tree, up to the highest level; this ensures that each entry has a unique identifier, thereby avoiding the duplication of entries.

The top level of the LDAP directory is the base, referred to as the "base DN". Beneath this level, containers to logically separate data are created; for historical reasons (inherited from X.500), these are usually set up as **Organizational Units** (**OU**). Each entry's DN is made up of two parts: the **Relative Distinguished Name** (**RDN**), which is the part of the DN that isn't related to the directory tree structure, and the location in the LDAP directory where the register resides, which, as already mentioned, is recursively constructed in reverse order following the tree nodes. An entry's RDN is derived from the entry's attributes. In the simplest and most common case, the RDN takes the form of <attribute_name> = <value>, although it can be composed of multiple attribute-value pairs separated by plus (+) signs.

Each entry has a set of attributes which contain information about the object the entry refers to. Each attribute has a type (specifying the syntax and classes of values that can be stored in the attribute) and one or several values. An "**object class**" defines a set of attributes. A **schema** describes a set of object classes and additionally indicates which attributes are mandatory and which are optional, along with other information used by the server to determine how to perform attribute comparison operations or whether or not the *add* and *update* operations should be allowed. Each entry in the directory consists of one or more object classes. The schema, therefore, defines the type of objects stored in the directory.

1.2.2 Attributes: examples

Some examples of attributes are as follows:

- cn (*common name*): common name for an entry.
- o (*organization*), the person's company.
- ou (*organizationalUnitName*): organizational unit (such as the name of a department in a company).
- dc (*domainComponent*).
- c (*countryName*).
- uid (*user id*).
- sn (*surname*).
- *givenname*.
- *telephoneNumber*.
- *mail*.
- *owner*: owner of the entry.

The so-called operational attributes are those used by the servers to manage the directory system itself. Their values are not returned in search operations unless their name makes this explicit. These attributes are automatically maintained by the server and cannot be modified by the clients. The following are some of the sorts of operational attributes that an entry can contain:

- *creatorsName*: This is the DN for the user who introduced this entry in the directory.
- *createTimestamp*: This is the time the entry was added to the directory.
- *modifiersName*: This is the DN for the user who last modified the entry.
- *modifyTimestamp*: This is the time the entry was last modified.
- *subschemaSubentry*: This is the entry DN (or subentry) for the subschema that controls the schema for this entry. The following point explains subschema entries.

1.2.3 Subschema Entries

Subschema entries are used to administer information on the directory schema, specifically the classes of objects and the types of attributes supported by the directory servers. A single subschema entry contains all the schema identifications used by the entries in a particular part of the directory tree.

1.2.4 Suffixes and referrals

An LDAP server may not hold all of the *Directory Information Tree* (DIT). For example, it might store the entries for a specific company department and not the preceding ones in the tree. The highest node in the DIT stored by the server is called the **suffix**. Each server entry ends with that suffix, since in the *Distinguished Name* (DN) syntax the top level nodes go at the end.

A server can support multiple suffixes. So, continuing with the previous example, the entries corresponding to two different departments can be stored.

Given that a server might not hold the whole DIT, sometimes several servers need to be linked together to form a distributed directory that contains the entire DIT. This is accomplished with **referrals**: if a server doesn't have the data request in a search operation, it can return a referral to another LDAP server holding the desired information.

A referral is an entry whose object class is *referral*, and contains a ref attribute whose value is the URL of the entry named in the referencing server.

1.2.5 Server information

A version 3 LDAP server should provide information about itself. This information is found in a special entry in the tree root known as a *DSA-specific Entry* (DSE). This entry is associated with a zero-length DN (i.e., an empty DN) and contains attributes that describe the server and provide basic information about the server and the DIT held in the server.

For example, the server-specific information could include:

- The suffixes held by the server, also known as *naming contexts*.
- The DN for a special entry that contains a list of all the classes of object schemas and attributes defined in the server (subschema entry).
- The supported LDAP version or versions.
- The list of supported extended operations and controls.
- The list of SASL security mechanisms.
- A list of alternative LDAP servers.

1.3 Protocol: Description

The LDAP protocol is based on a client-server model: the client sends a request to the server describing the operation to be executed; upon receiving the request, the server then carries out the relevant actions in the directory; finally, the server sends the client a response message with the result of the operation, either that the operation went ahead without any problems, or the obtained error.

LDAP is designed to run over a reliable, connection-oriented transport protocol. *Transmission Control Protocol* (TCP) is used for this. The default **TCP port** where the LDAP server listens for client messages is **389**, although listening can be implemented in another port. Clients, by implementation, must be able to send requests to any TCP port.

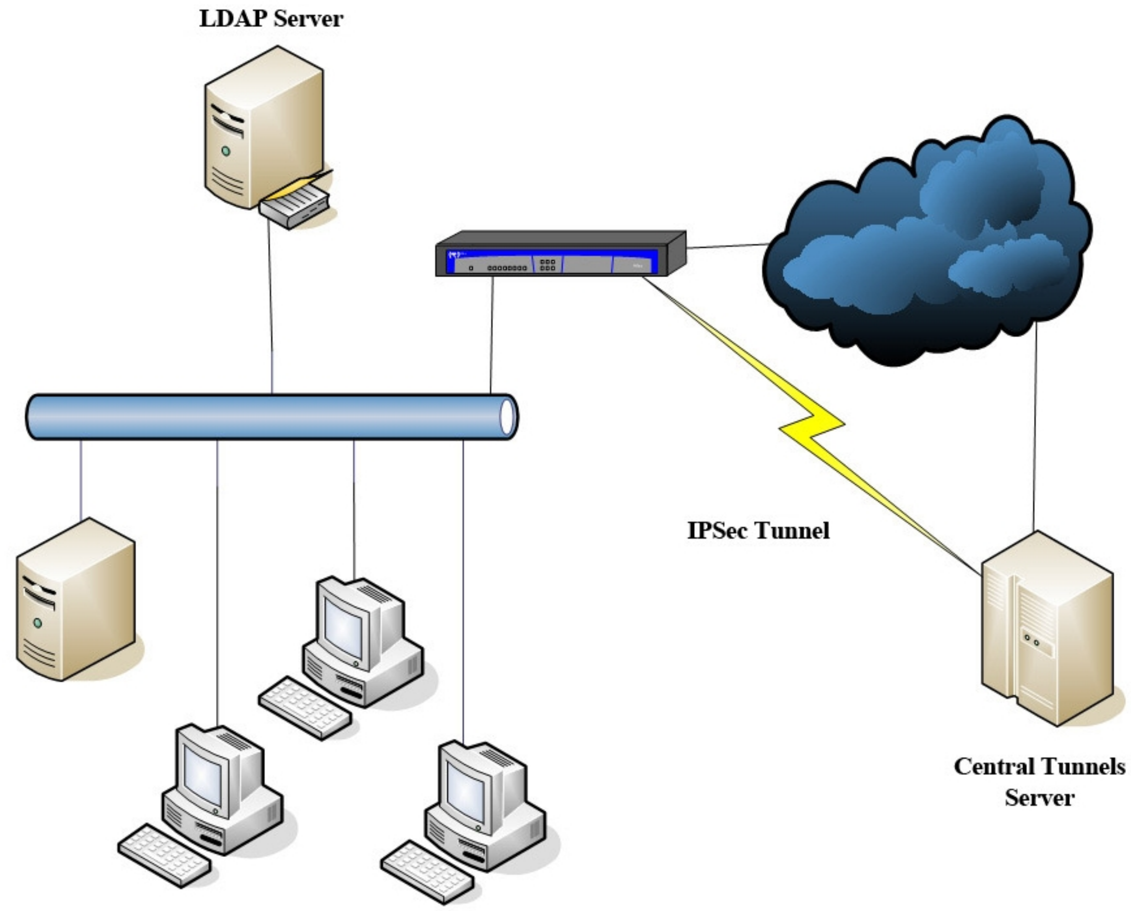
Although servers must return responses (provided this has been specified in the protocol definition), client requests and server responses do not have to be synchronized. That is, after the client performs a series of LDAP requests on the server, the server's responses can reach the client in a different order to the order in which the corresponding requests were sent; provided, mind you, that each request requiring a response does in fact receive one.

The LDAP description is carried out using *Abstract Syntax Notation 1* (ASN.1), and for message transmission a subset of ASN.1 *Basic Encoding Rules* (BER) is used.

The latest version of the LDAP protocol is version 3. Clients expressly indicate the version of the protocol they support by performing *bind* operation requests on the server. If the server hasn't received a *bind* request from the client, it must automatically assume that the client supports version 3 (this is because version 2 requires clients to perform *bind* requests before performing any other operation).

1.3.1 Operating Scenario

Here we show an example of the LDAP operating environment. As already mentioned, an LDAP server allows centralized access to the information that needs to be available in a distributed systems and services environment. This figure shows a network made up of work stations and servers, with authorized user access data stored in an LDAP directory. Mail aliases and the company phone directory could be stored too. We have also considered here the possibility of storing in the directory the keys necessary to set up a secure VPN through an IPSec tunnel established from a corporate network router towards a specific central tunnel server.



1.3.2 LDAP Messages

The PDU (*Protocol Data Unit*) generic format for exchanging LDAP messages is as follows:

messageID	**protocolOp**	**Controles**

The first field is the **message identifier**. This allows differentiating between requests executed in the same session: two requests pending a response from the server in the same connection cannot have the same message identifier.

After the identifier comes **a part of the message whose format depends on the type of message** being dealt with. The possible message types are given below. You can see that there is a pair of messages (*request*, which are requests from the client to the server, and *response* which are the responses sent by the server to the client) associated with the *bind*, *modify*, *add*, *delete*, *modify DN*, *compare* and *extended* operations, while the *unbind* and *abandon* operations are only available in the *request* message (these do not require a response from the server), and finally, in the case of *search* operations, 4 types of associated messages are defined. The following section lists the possible LDAP operations.

List of LDAP messages:

- *bindRequest*
- *bindResponse*
- *unbindRequest*
- *searchRequest*
- *searchResEntry*
- *searchResDone*
- *searchResRef*
- *modifyRequest*
- *modifyResponse*
- *addRequest*
- *addResponse*
- *delRequest*
- *delResponse*
- *modDNRequest*

- *modDNResponse*
- *compareRequest*
- *compareResponse*
- *abandonRequest*
- *extendedReq*
- *extendedResp*

The last part of the LDAP message is an **optional** field consisting of a **control list**. A control is a way of specifying additional information. Together with *extended* operations, they allow you to extend the protocol without changing it. Controls modify the behavior of an operation, adding at the end of the operation a series of parameters that can be used as an entry for another function or operate in another way with them. Controls sent as part of a request are only applied to that request and are not saved. Each control consists of the following:

- **Control type**: this is an *Object Identifier* (OID) that uniquely identifies the control.
- **Criticality field**: this is a boolean that tells the server how to behave when it receives a control type that it doesn't recognize or one that is inappropriate for the operation. If this field is TRUE, the server doesn’t perform the operation and returns the *unsupportedCriticalExtension* result code. If, on the other hand, this field is FALSE, the server ignores the control.
- **Control value**: information associated with the control and whose format is the one defined for this control.

Differences between the LDAP versions 2 and 3

As regards the messages exchanged between client and server, the main differences between LDAP versions 2 and 3 are as follows:

- In version 2, clients must establish a session with the server before performing other types of operations. Therefore, all message exchanges always begin with a *bindRequest*. Version 3 does not require clients to establish a session before they can issue other types of requests to the server.
- LDAP version 2 *search* operations involve the exchange of only 2 messages: *searchRequest* and *searchRespons*. In LDAP version 3, there are 4 types of messages associated with *search* operations.
- In LDAP version 3, a new type of filter has been added for use in *search*: *extensibleMatch* operations.
- The authentication types supported in LDAP version 2 (established in the *bindRequest* message) are simple authentication (using a password in clear) and Kerberos version 4. LDAP version 3 uses the Simple Authentication and Security Layer (SASL) authentication framework.
- LDAP version 2 does not support *extended* operations.
- LDAP version 2 doesn’t allow the use of controls to modify operations and add additional information. Therefore, the final part of the messages does not contain anything.
- The LDAP version 2 *modifyRDN* operation is renamed *modifyDN* in version 3. Also, in addition to allowing the least significant node-name component to change, version 3 adds the possibility of moving an entire subtree of entries and hanging them from a different source node.

1.3.3 Operations

LDAP operations can be divided into the following three categories:

(1) *Authentication* operations: *bind*, *unbind* and *abandon*, used to connect to and disconnect from an LDAP server, establish access rights and protect information.

(2) *Query* operations: Include the *search* and *compare* operations, used to retrieve information from a directory.

(3) *Update* operations: *add*, *delete*, *modify* and *modify DN*, these modify information stored in a directory.

1.3.3.1 Authentication Operations

Authentication operations are used to establish and end sessions between an LDAP client and an LDAP server. This session can have different security levels ranging from an insecure anonymous session, an authenticated session in which the client identifies itself by entering a password, to secure, encrypted sessions using *Simple Authentication and Security Layer* (SASL). SASL was introduced in LDAP version 3 to improve the weak authentication methods available in LDAP version 2; though some manufacturers chose to add more powerful authentication methods, such as Kerberos, in order to continue using version 2.

1.3.3.1.1 Bind

Initiates an LDAP session between a client and a server. Allows the client to prove his identity by authenticating himself with the server. The parameters included in the *bindRequest* message are as follows:

The LDAP version number to be used in the session. The version is unilaterally established by the client without negotiating with the server. In cases where the client chooses to use LDAP version 2, the server cannot include any version 3-specific fields in its messages.

The directory object name (DN) used by the client to establish the connection. This value can be left blank (zero length string) when attempting anonymous connections.

Information used to authenticate the above DN, if applicable. In the case of simple authentication, the only thing to go in the corresponding field is the password in clear. In the case of establishing non-authenticated or anonymous sessions, the field is left blank. If you use Kerberos authentication (only in LDAP version 2), you include the so-called “Kerberos ticket”. If SASL is used, you enter the mechanism name and credentials in this field.

On receiving a *bindRequest* message, the server authenticates the client (if necessary) and returns a *bindResponse* message indicating the result of the authentication and session establishment attempt. With some SASL mechanisms, several *bindRequest* and *bindResponse* messages must be exchanged. To indicate this circumstance, the *bindResponse* messages sent by the server to the client take the *saslBindInProgress* value in the *resultCode* field.

1.3.3.1.2 Unbind

Ends a session between the client and server. This operation does not require confirmation from the server to the client: the client sends an *unbindRequest* message and assumes that the session has ended, while the server, on receiving this message, also assumes the session is terminated and drops any pending requests.

1.3.3.1.3 Abandon

Allows the client to request the server to cancel an operation in progress. This operation does not have an associated response message from the server to the client: when the client sends an *abandonRequest* message to the server, both consider that the operation being executed should be canceled.

1.3.3.2 Query operations

1.3.3.2.1 Search

Search operations allow a client to request the LDAP server to search a certain part of the DIT for information that meets certain user-specified criteria. To carry out a search, the following parameters must be specified:

- *baseObject*: This is the DN corresponding to the DIT node where the search begins.
- Search scope: this specifies the depth of the search. There are three options:

  - *baseObject*: only the *baseObject* is considered.

  - *singleLevel*: all nodes immediately beneath and connected to the *baseObject* (without including the base object itself) are examined.

  - *wholeSubtree*: the *baseObject* and all of its subordinates are examined.

- Search filter: specifies the criteria an entry must meet to be returned from a search operation.
- Attributes to return from the entries that match the search criteria. You also need to indicate whether you want the response to only include attribute types or attribute types and values.
- How referrals should be treated. referrals indicate other servers that contain the requested search information. There are the following alternatives:

  - Do not convert aliases when processing the search nor when locating the point *baseObject* from which to start searching.

  - Convert aliases in the search but not in locating the *baseObject*.

  - Convert aliases when locating the *baseObject* but not in the search.

  - Always convert aliases, both when locating the *baseObject* and in the search.

- Size limit: this defines the maximum number of entries that should be returned in response to the search request.
- Time limit: this defines the maximum length of time, in seconds, that the search should last.

The purpose of having time and size limits is to prevent excessive usage of server resources. These limits are imposed by the client when requesting a search operation; however the server can place even stricter limits than those set by the client.

After executing a search, the server's response to the client will differ depending on the LDAP version being used:

- If LDAP version 2 is used, the server sends a *searchResponse* message to the client for each found entry and ends by sending another *searchResponse* message indicating the success of the operation or obtained error.
- In LDAP version 3, the server returns *searchResEntry* messages with the found entries, *searchResRef* with the un-

translated referrals, and ends by sending a *searchResDone* message indicating the success of the operation or obtained error.

1.3.3.2.2 Compare

The compare operation compares an entry (indicated as parameter) with an attribute value (also specified).

The server returns the comparison result or the obtained error.

1.3.3.3 Update operations

Update operations modify the directory contents. There are 4 possible operations in this category:

- *Add*
- *Delete*
- *Modify*
- *Modify DN*

1.3.3.3.1 Add

Adds a new entry to the directory. You must specify the DN and the attributes, at least those attributes making up the DN, the *objectClass* and those attributes identified as mandatory in this *objectClass*.

1.3.3.3.2 Delete

Deletes the entry with the specified DN from the directory. Only entries that don't have any branches can be deleted (those known as “leaves”).

1.3.3.3.3 Modify

This operation allows you to modify the entry with the specified DN using the specified modification list. Each modification consists of adding, modifying or deleting certain values for certain attributes:

- *add*: adds the listed values to the given attribute, creating the attribute if necessary.
- *delete*: Deletes the listed values from the given attribute, removing the entire attribute if no values are listed or if all current values for the attribute are found in this list. You cannot delete values that form part of the entry's RDN.
- *replace*: replaces all existing values of the given attribute with the new listed values, creating the attribute if necessary. In cases where no value is included in the list, the entire attribute is deleted.

1.3.3.3.4 Modify DN

Allows you to modify the least significant component of a node's name or move an entire subtree of entries to a new location in the directory. The following parameters are included in the *modDNRequest* message:

- *entry*: the DN of the entry to be modified.
- *newrdn*: the RDN that will form the least significant component of the new entry name.
- *deleteoldrdn*: a boolean parameter that controls whether the old RDN attribute values are to be retained as attributes of the entry, or deleted from the entry.
- *newSuperior*: if present, this is the DN of the entry that will be immediately above the existing entry. That is, it indicates the node the entire subtree now hangs from that begins with the specific entry being modified

The possibility of moving the entire subtree wasn't considered in LDAP version 2. In LDAP version 3, the operation is referred to as *modifyRDN* and for logical reasons the *newSuperior* parameter is not included in the client *request* messages.

1.3.3.4 Extended operations

Extended operations, together with controls, allow you to extend the LDAP protocol without changing it. In this case, new operations are added to the protocol. The message part of this type of operation consists of an OID, which must be unique for each *extended request*, and a character string containing other information specific to the operation (optional).

These operations are only available in LDAP version 3.

1.4 Standards the protocol is based on

The core LDAP specifications are fully defined in RFCs (*Request For Comments*). A complete list of LDAP-related RFCs is shown below.

Initial LDAP specifications:

RFC 1777 - Lightweight Directory Access Protocol.

RFC 1778 - The String Representation of Standard Attribute Syntaxes.

RFC 1779 - String Representation of Distinguished Names.

LDAP v3 specifications:

RFC 2251 - Lightweight Directory Access Protocol (v3).

RFC 2252 - Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions.

RFC 2253 - Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names.

RFC 2254 - The String Representation of LDAP Search Filters.

RFC 2255 - The LDAP URL Format.

RFC 2256 - A Summary of the X.500(96) User Schema for use with LDAPv3.

Additional specifications:

RFC 1823 - The LDAP Application Program Interface.

RFC 2079 - Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers.

RFC 2116 - X.500 Implementations Catalog-96.

RFC 2164 - Use of an X.500/LDAP directory to support MIXER address mapping.

RFC 2247 - Using DNS Domain names in LDAP/X.500 Distinguished Names.

RFC 2307 - An Approach for Using LDAP as a Network Information Service.

RFC 2377 - Naming Plan for Internet Directory-Enabled Applications.

RFC 2559 - Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2.

RFC 2596 - Use of Language Codes in LDAP.

RFC 2649 - An LDAP Control and Schema for Holding Operation Signatures.

RFC 2696 - LDAP Control Extension for Simple Paged Results Manipulation.

1.5 LDAP in the ROUTER

The **ROUTER** has an LDAP client, with the following restrictions:

- The supported LDAP version is **version 2**.
- Only **simple authentication** is supported.

The following chapter describes how to access the configuration for this functionality and the commands available to select the values of the various configurable operating parameters.

Chapter 2 Configuration

2.1 Accessing the LDAP configuration

As already said, the **ROUTER** has an LDAP client. To access the configuration environment for this feature, enter the following commands:

```
*process 4

Config>feature ldap

-- LDAP User Configuration --
LDAP config>
```

Or:

```
*config

Config>feature ldap

-- LDAP User Configuration --
LDAP config>
```

Once you've accessed the LDAP client configuration menu, the available commands are as follows:

Command	**Function**
? (HELP)	Lists the commands or available options.
AUTHENTICATION	Configures the password the client uses to authenticate with the server (using simple authentication).
DESTINATION-IP	Sets the LDAP server IP address where requests are sent.
LIST	Displays the LDAP client configuration.
NAME-AUTH	Name used by the LDAP client to execute the connection.
NO	Restores default values for various configuration parameters.
PORT	Configures the LDAP server TCP port that client requests are sent to.
SEARCH-OPTIONS	Allows you to configure various parameters related to search operations.
SOURCE-IP	Sets the source IP address for LDAP messages coming from the client.
EXIT	Exits the LDAP client configuration menu.

2.2 LDAP client configuration commands

This section describes the configuration parameters available for the LDAP client.

2.2.1 ? (HELP)

This command is used to list the valid commands at the level where the router is programmed. You can also use this command after a specific command to list the available options.

Syntax:

```
LDAP config>?
```

Example:

```
LDAP config>?
  authentication     Sets the authentication string
  destination-ip     Sets destination IP address (LDAP server)
  list               Displays configured LDAP parameters
  name-auth          Sets the name used for authentication
  no                 Sets default values to configuration parameters
  port               Sets LDAP port on server
  search-options     Configures search options
  source-ip          Sets source IP address (LDAP client)
  exit              Exits LDAP configuration menu
LDAP config>
```

2.2.2 AUTHENTICATION

Configures the password the client uses to authenticate with the server (using simple authentication). If this is left empty, the client tries to establish an anonymous or unauthenticated session. This parameter is closely related to **name-auth**, since when configuring a specific name to be used by the LDAP client in connections, the password to be established is the one associated with this specific name.

Syntax:

```
LDAP config>authentication <password>
```

Example:

```
LDAP config>authentication prueba
LDAP config>
```

2.2.3 DESTINATION-IP

Sets the IP address of the LDAP server that the client requests are addressed to. Only one LDAP server can be configured. This parameter must be configured in order to perform the requests sent by the client.

Syntax:

```
LDAP config>destination-ip <ip_address>
```

Example:

```
LDAP config>destination-ip 172.24.0.201
LDAP config>
```

2.2.4 LIST

Displays the LDAP client configuration.

Syntax:

```
LDAP config>list
```

Example:

```
LDAP config>list
Bind Information
----------------
IP Source      [  0.  0.  0.  0]
IP Destination [172. 24.  0.201]
Server Port    [            389]
Authentication [rootpswd]
Name for auth  [root@teldat.com]

Search Information
------------------
Timeout           : 5
Max Search Results: 10
Implicit NULL     : YES
Window Size       : 536

Scope             : BASE
Deref Aliases     : NEVER
Size Limit        : 0
Time Limit        : 0
Filter Attributes :
  No Attributes were established.
LDAP config>
```

This includes the following information:

- Connection or session establishment data:

- Source IP address configured for the LDAP client (if 0.0.0.0 has been configured, the messages take the output interface IP address as source address).

- LDAP server IP address used to try and establish the sessions from the client.

- Server TCP port the client sends the LDAP messages to.

- Password used by the client to authenticate with the server. This must be the one associated with the directory administrator name.

- Name of the directory object used by the client to establish the connections. This is the DN of the directory administrator. In the case of anonymous or unauthenticated connections, this name can be left in blank (zero string).

- Information used in search operations:

- Timer to control the TCP transmission.

- Maximum number of search results that the client can take into account.

- Whether or not the implicit *null* option is enabled, which allows coding the empty base DN as 04 00 (in hexadecimal) instead of 04 04 4E 55 4C 4C.

- Size of the TCP window that controls the reception of responses to transmitted messages.

- Search scope or depth of the search in the DIT. The possible values are:

- BASE: only the base object (*baseObject*) is examined.
- WHOLE: the *baseObject* and all of its subordinates are examined.
- SINGLE: only the nodes that are immediately beneath and directly connected to the *baseObject* (without including the base object itself) are examined.

- How to treat aliases or referrals (references to other servers where the requested information is held). The possible values are:

- NEVER: Do not convert aliases encountered when searching nor when locating the starting point of the search (*baseObject*).
- SEARCHING: Convert aliases when searching but not in locating the *baseObject* of the search.
- FINDING: Convert aliases in locating the *baseObject* of the search, but not while searching.
- ALWAYS: Always convert aliases, both when searching and when locating the *baseObject* of the search.

- Maximum number of entries the server returns in the responses to search requests. 0 indicates no limit on the number of entries.

- The time the server spends performing the search requests sent by the client. 0 indicates no time limit.

- List of attributes for each found entry that the server includes in the responses to search requests. If there aren't any attributes in the list, the server returns all available attributes for each entry found in the search.

2.2.5 NAME-AUTH

Configures the directory object name used by the client to establish the connections. This is the directory's administrator DN. In the case of anonymous or non-authenticated connections, this name can be left blank (zero length string).

The format to use in the name configured through this parameter depends on the format required by the LDAP server the connection was made with.

This parameter is closely related to the **authentication** parameter, since when configuring a specific name for the LDAP client to use in the connections, the password to establish will be the one associated with this specific username.

Syntax:

```
LDAP config>name-auth <name>
```

Example:

```
LDAP config>name-auth admin@teldat.com
LDAP config>
```

2.2.6 NO

Restores the default values for the various configuration parameters.

Syntax:

```
LDAP config>no ?
  authentication    Sets the authentication string
  destination-ip    Sets destination IP address (LDAP server)
  name-auth         Sets the name used for authentication
  port              Sets LDAP port on server
  search-options    Configures search options
  source-ip         Sets source IP address (LDAP client)
```

2.2.6.1 no authentication

Deletes the password used by the client when attempting to authenticate with the server (using simple authentication). In this case the client tries to establish an anonymous or non-authenticated session.

Syntax:

```
LDAP config>no authentication
```

Example:

```
LDAP config>no authentication
LDAP config>
```

2.2.6.2 no destination-ip

Deletes the previously configured LDAP server address.

Syntax:

```
LDAP config>no destination-ip
```

Example:

```
LDAP config>no destination-ip
LDAP config>
```

2.2.6.3 no name-auth

Deletes the directory object name used by the client to establish the connections. This name can be left blank in the case of anonymous or non-authentication connections.

Syntax:

```
LDAP config>no name-auth
```

Example:

```
LDAP config>no name-auth
LDAP config>
```

2.2.6.4 no port

Resets the default value for the LDAP server TCP port where the client directs the requests. The default value is 389.

Syntax:

```
LDAP config>no port
```

Example:

```
LDAP config>no port
LDAP config>
```

2.2.6.5 no search-options

Clears the attribute list from the entries found in a search operation that the server must include in the response it sends to the client (so that all the attributes for each entry found are included in the response) and resets the default values of a number of different *search*-related parameters:

- Establishes that the referrals or aliases returned by the server in search operations are not to be converted in the search nor when locating the starting point of the search (the so-called *baseObject*), i.e., as if the **never** option is configured.
- Disables the possibility of encoding the empty search base DN, in accordance with ASN.1 *Basic Encoding Rules* (BER), as 04 00 (in hexadecimal), which must be configured as 04 04 4E 55 4C 4C.
- Resets the maximum number of search results that can be returned by the server to the default value 10.
- Sets the search scope (depth of the search in the DIT) default value: only examine the *baseObject* (the search source node). This is the same as configuring the **base** option.
- Sets the maximum number of entries returned by the server in response to search requests to 0. This means that there is no limit to the number of entries that can be included.
- Sets the maximum length of time, in seconds, that a search can last to 0 seconds. This means that there is no time limit for receiving a response.
- Sets the value of the TCP transmission timer to 5 seconds.
- Resets the size of the TCP window that controls the reception of the responses to the transmitted messages to its default value of 536.

Syntax:

```
LDAP config>no search-options
```

Example:

```
LDAP config>no search-options
Default values established.
LDAP config>
```

2.2.6.6 no source-ip

Deletes the source IP address for messages sent by the LDAP client to the LDAP server. If no source IP address is specified, then the source IP address of the output interface is used for these messages when they exit the client.

Syntax:

```
LDAP config>no source-ip
```

Example:

```
LDAP config>no source-ip
LDAP config>
```

2.2.7 PORT

Configures the TCP port for the LDAP server where the client requests are sent. By default the requests go to port 389. Permitted values range between 0 and 65535.

Syntax:

```
LDAP config>port <port_number>
```

Example:

```
LDAP config>port 1122
LDAP config>
```

2.2.8 SEARCH-OPTIONS

Allows you to configure various parameters related to *search* operations:

- How the server should treat any referrals or aliases returned to the client as a result of a search operation.
- Attributes for each found entry that the server includes in the responses to search requests.

- How to encode the search's base DN if it is empty.
- Maximum number of search results that the client can take into account.
- Search scope or depth of the search in the tree (DIT).
- Maximum number of entries returned by the server in the responses to search requests.
- Maximum time in seconds that the search operation can last.
- TCP transmission timer.
- Size of the TCP window that controls the reception of responses to the transmitted messages.

Syntax:

```
LDAP config>search-options ?
  aliases                  Sets deref aliases
  attributes               Sets attributes
  implicit-null            Enables implicit NULL
  max-number-of-results    Sets maximum number of search results
  no                       Negate a command or set its defaults
  scope                    Sets scope: base, single or whole
  size-limit               Sets size limit
  time-limit               Sets time limit
  tout                     Sets timeout
  window-size              Sets window size
```

2.2.8.1 search-options aliases

Allows you to specify how the server should treat the referrals or aliases it returns to the client as a result of the search operations. A referral or alias is a reference to another server where the search information can be found when it is not held in the LDAP server the request is being made to.

Syntax:

```
LDAP config>search-options aliases ?
  always       Sets deref aliases: always
  finding      Sets deref aliases: finding
  never        Sets deref aliases: never
  searching    Sets deref aliases: searching
```

The following alternatives are available:

- Never converts aliases (or referrals) encountered when searching nor in locating the starting point of the search (known as the *baseObject*): **never** option.
- Converts any aliases encountered while searching but not in locating the *baseObject* of the search: **searching** option.
- Converts aliases in locating the *baseObject* of the search, but not when searching: **finding** option.
- Always converts aliases, both in searching and in locating the *baseObject* of the search: **always** option.

By default, aliases are neither converted when searching nor when locating the starting point of the search, i.e., the **never** option is enabled.

Example:

```
LDAP config>search-options aliases finding
LDAP config>
```

This example establishes that the referrals or aliases for finding the point from where to start searching (i.e., the *baseObject Distinguished Name* (DN) or *Directory Information Tree* (DIT) node from where the entries whose attributes have the characteristics specified in the search filters will be searched for) are converted. If the *baseObject* isn't found in the server that was sent the request, it is searched for in the server indicated in the returned referral. However, referrals that are returned as a result of the searches are not translated.

2.2.8.2 search-options attributes

Allows you to specify a list of attributes to be returned by the server for each entry found in the search. By default, this list is empty, implying that the server response will include all available attributes for each found entry.

Syntax:

```
LDAP config>search-options attributes ?
  delete    Deletes a previously configured attribute
  new       Adds a new attribute
```

2.2.8.2.1 search-options attributes delete

Deletes an attribute that was added to the list of attributes to be returned by the server for entries found in a search operation. You must specify the name of the attribute to be deleted, which will be a text containing between 1 and 64 characters.

Syntax:

```
LDAP config>search-options attributes delete <attribute>
```

Example:

```
LDAP config>search-options attributes delete cn
Attribute cn successfully deleted.
LDAP config>
```

2.2.8.2.2 search-options attributes new

Adds an attribute to the list of attributes to be returned by the server for entries found in a search operation. You must specify the name of the attribute to add, which will be a text containing between 1 and 64 characters.

Syntax:

```
LDAP config>search-options attributes new <attribute>
```

Example:

```
LDAP config>search-options attributes new cn
LDAP config>
```

2.2.8.3 search-options implicit-null

This parameter allows you to select how the search's base DN will be encoded if empty: if you configure implicit *null*, the empty base DN is encoded, in accordance with ASN.1 *Basic Encoding Rules* (BER), as 04 00 (in hexadecimal); if it is not configured, the empty base DN is encoded as 04 04 4E 55 4C 4C. By default, implicit *null* is enabled (i.e., the short version for encoding).

Syntax:

```
LDAP config>search-options implicit-null
```

Example:

```
LDAP config>search-options implicit-null
LDAP config>
```

2.2.8.4 search-options max-number-of-results

Establishes the maximum number of search results that the client can have. By default this is 10. Permitted values range between 10 and 2147483647.

Syntax:

```
LDAP config>search-options max-number-of-results <max_num_results>
```

Example:

```
LDAP config>search-options max-number-of-results 15
LDAP config>
```

2.2.8.5 search-options no

Deletes the list of attributes to be returned by the server for entries found in a search operation (so all available attributes for each found entry are included in the response) or resets the default values for various parameters related to the *search* operations:

- Establishes that the referrals or aliases returned by the server in search operations are not to be converted when searching nor in locating the starting point of the search (known as the *baseObject*), i.e., as if the **never** option is configured.
- Disables the possibility of encoding the empty search base DN, in accordance with ASN.1 *Basic Encoding Rules* (BER), as 04 00 (in hexadecimal), which must be configured as 04 04 4E 55 4C 4C.
- Sets the maximum number of search results that can be returned to the default value of 10.

- Sets the search scope (depth of the search in the DIT) default value: only examine the *baseObject* (the search source node). This is the same as configuring the **base** option.
- Sets the maximum number of entries returned by the server in response to search requests to 0. This means that there is no limit to the number of entries that can be included.
- Sets the maximum length of time, in seconds, that a search operation can last to 0 seconds. This means that there is no time limit for receiving a response.
- Configures the TCP transmission timer value to 5 seconds.
- Resets the size of the TCP window that controls the reception of responses to the transmitted messages to its default value, 536.

Syntax:

```
search-options no ?
  aliases                  Sets deref aliases
  attributes               Sets attributes
  implicit-null            Enables implicit NULL
  max-number-of-results    Sets maximum number of search results
  scope                    Sets scope: base, single or whole
  size-limit               Sets size limit
  time-limit               Sets time limit
  tout                     Sets timeout
  window-size              Sets window size
```

2.2.8.5.1 search-options no aliases

Restores the default value for the parameter that controls the way in which referrals or aliases (references to other servers containing the requested information) should be treated. The default value is that aliases (or referrals) are neither converted when searching nor when locating the starting point of the search (the so-called *baseObject*), i.e., as if the **never** option is configured.

Syntax:

```
LDAP config>search-options no aliases
```

Example:

```
LDAP config>search-options no aliases
LDAP config>
```

2.2.8.5.2 search-options no attributes

Deletes all the attributes previously added to the list of attributes to be returned by the server for each found entry. If this list is left empty, the server understands that all available attributes for each found entry are to be included in the response.

Syntax:

```
LDAP config>search-options no attributes
```

Example:

```
LDAP config>search-options no attributes
LDAP config>
```

2.2.8.5.3 search-options no implicit-null

Disables the possibility of encoding the empty search base DN, in accordance with ASN.1 *Basic Encoding Rules* (BER), as 04 00 (in hexadecimal), which must be configured as 04 04 4E 55 4C 4C

Syntax:

```
LDAP config>search-options no implicit-null
```

Example:

```
LDAP config>search-options no implicit-null
LDAP config>
```

2.2.8.5.4 search-options no max-number-of-results

Sets the value for the maximum number of search results. The default value is 10.

Syntax:

```
LDAP config>search-options no max-number-of-results
```

Example:

```
LDAP config>search-options no max-number-of-results
LDAP config>
```

2.2.8.5.5 search-options no scope

Sets the search scope (depth of the search in the DIT) default value: only examine the *baseObject* (the search source node). This is the same as configuring the **base** option.

Syntax:

```
LDAP config>search-options no scope
```

Example:

```
LDAP config>search-options no scope
LDAP config>
```

2.2.8.5.6 search-options no size-limit

Restores the default value 0 for the parameter controlling the maximum number of entries the server returns in response to a search request. A 0 value means that there is no limit to the number of entries that can be included.

Syntax:

```
LDAP config>search-options no size-limit
```

Example:

```
LDAP config>search-options no size-limit
LDAP config>
```

2.2.8.5.7 search-options no time-limit

Restores the default value for the maximum length of time, in seconds, that a search operation can last. The default value is 0 seconds. This means that there is no time limit for receiving a response.

Syntax:

```
LDAP config>search-options no time-limit
```

Example:

```
LDAP config>search-options no time-limit
LDAP config>
```

2.2.8.5.8 search-options no tout

Resets the TCP transmission timer to its 5-second default value.

Syntax:

```
LDAP config>search-options no tout
```

Example:

```
LDAP config>search-options no tout
LDAP config>
```

2.2.8.5.9 search-options no window-size

Restores the size of the TCP window that controls the reception of responses to the transmitted messages to its default value of 536.

Syntax:

```
LDAP config>search-options no window-size
```

Example:

```
LDAP config>search-options no window-size
LDAP config>
```

2.2.8.6 search-options scope

Search scope: specifies the depth of the search in the DIT. There are three options:

base: only the *baseObject* (the search source node) is considered.

single: all nodes immediately beneath and connected to the *baseObject* (without including the base object itself) are examined.

whole: the *baseObject* and all of its subordinates are examined.

The default value is **base**.

Syntax:

```
LDAP config>search-options scope ?
  base      Sets scope: base object
  single    Sets scope: single level
  whole     Sets scope: whole subtree
```

Example:

```
LDAP config>search-options scope whole
LDAP config>
```

2.2.8.7 search-options size-limit

Sets a size limit for responses to search operations, restricting the maximum number of entries the server can return in response to search requests. The default is 0, which means that no limit has been set and the valid values are between 1 and 65535. To restore the default value, enter **search-options no size-limit.**

Syntax:

```
LDAP config>search-options size-limit <entry_number>
```

Example:

```
LDAP config>search-options size-limit 10
LDAP config>
```

2.2.8.8 search-options time-limit

Sets a time limit, in seconds, for receiving a response to a search request. The default value is 0, which indicates no time limit is set, and the permitted values are between 0s and 65535s, i.e., between 0 seconds and 18 hours, 12 minutes and 15 seconds (0s…18h12m15s).

Syntax:

```
LDAP config>search-options time-limit <max_time>
```

Example:

```
LDAP config>search-options time-limit 10s
LDAP config>
```

2.2.8.9 search-options tout

Value of the timer controlling the TCP transmission. The default is 5 seconds, and the permitted values are between 5s and 2147483647s, i. e., between 5 seconds and 3550 weeks, 5 days, 3 hours, 14 minutes and 7 seconds (5s…3550w5d3h14m7s).

Syntax:

```
LDAP config>search-options tout <timeout>
```

Example:

```
LDAP config>search-options tout 10s
LDAP config>
```

2.2.8.10 search-options window-size

The size of the TCP window that controls the reception of responses to the transmitted messages. The default value is 536 and the permitted values are between 128 and 65535.

Syntax:

```
LDAP config>search-options window-size <size>
```

Example:

```
LDAP config>search-options window-size 128
LDAP config>
```

2.2.9 SOURCE-IP

Sets the source IP address for messages sent by the LDAP client to the LDAP server. If no source IP address is specified, then the source IP address of the output interface is used for these messages when they exit the client.

Syntax:

```
LDAP config>source-ip <ip_address>
```

Example:

```
LDAP config>source-ip 192.168.1.1
LDAP config>
```

2.2.10 EXIT

Use this command to exit the LDAP client configuration menu and return to the main configuration menu (*Config*>).

Syntax:

```
LDAP config>exit
```

Example:

```
LDAP config>exit
Config>
```

Chapter 3 Monitoring

3.1 Accessing the LDAP monitoring

To access the monitoring menu associated with the LDAP client, you must enter the following commands:

```
*monitor
Console Operator
+feature ldap
LDAP client monitor
LDAP+
```

Or

```
*monitor
Console Operator
+feature ldap
LDAP client monitor
LDAP+
```

Once you've accessed the LDAP client monitoring menu, you will have access to the following commands:

Command	**Function**
? (HELP)	Lists the available commands and their options.
BIND	The client initiates an LDAP session with the server by sending a *bindRequest* message.
DISABLE	Disables the viewing of the bytes exchanged with the LDAP server and/or the viewing of the state changes that occur.
ENABLE	Enables the viewing of the bytes exchanged with the LDAP server and/or the viewing of the state changes that occur.
LIST	Displays the established LDAP sessions, the enabled or disabled state for viewing bytes exchanged with the server and the state changes, as well as detailed client state information and the success or error results obtained when requesting server operations.
SEARCH	The client sends the server a request to search in the directory.
UNBIND	Ends a session between the client and the server. The client sends an *unbindRequest* message to the server.
EXIT	Exits the LDAP client monitoring menu and returns to the general monitoring menu (+).

3.2 LDAP client monitoring commands

This section describes the monitoring parameters available for the LDAP client. In addition to all the usual commands - for listing valid commands (**help**), listing operating information (**list**) and returning to the main monitoring menu (**exit**) -, the LDAP client monitoring also includes commands for enabling or disabling the viewing of the bytes exchanged with the LDAP server (**enable**/**disable frame**), enabling or disabling the viewing of the state changes that occur (**enable**/**disable trace**), and for asking the server to perform various LDAP operations (**bind**, **unbind**, **and search**).

3.2.1 ? (HELP)

Use this command to list the valid commands at the level where the router is programmed. You can also use this command after a specific command to list its available options.

Syntax:

```
LDAP+?
```

Example:

```
LDAP+?
  bind
  disable
  enable
  list
  search
  unbind
  exit
LDAP+
```

3.2.2 BIND

Starts an LDAP session between the configured client and server. The client sends a *bindRequest* message establishing the LDAP version to be used (version 2 in the client present in the router) and giving authentication information if necessary. In the router simple authentication is used with the password configured in clear and with the administrator name for the directory that has been established. If no password has been configured, an attempt is made to establish an unauthenticated or anonymous session.

An operation may be completed correctly or an error may occur. If an error occurs, an error message is displayed on screen with the cause of the error.

Example 1:

```
LDAP+bind
Connecting ...172.24.0.201 389...
LDAP+
```

In this case, the operation completed successfully and the LDAP session between the client and the server with IP address 172.24.0.201 on TCP port 389 has been established.

Example 2:

```
LDAP+bind
Connecting ...172.24.0.201 389...
 ERROR.
[BIND]: LDAPSTATUS_INVALIDCREDENTIALS
LDAP+
```

In this case, the LDAP client's request to establish a session with the server with IP address 172.24.0.201 on TCP port 389 resulted in an error because the name and/or the password sent by the client to authenticate were not admitted by the server.

3.2.3 DISABLE

Disables the viewing of the bytes exchanged with the LDAP server and/or the viewing of the LDAP client state changes.

Syntax:

```
LDAP+disable ?
  frame
  trace
```

3.2.3.1 disable frame

Disables the viewing of the bytes exchanged with the LDAP server (through an event). By default this option is disabled.

Syntax:

```
LDAP+disable frame
```

Example:

```
LDAP+disable frame
LDAP+
```

3.2.3.2 disable trace

Disables the viewing of the LDAP client state changes that occur when performing server requests. By default this option is disabled.

Syntax:

```
LDAP+disable trace
```

Example:

```
LDAP+disable trace
LDAP+
```

3.2.4 ENABLE

Enables the viewing of the bytes exchanged with the LDAP server and/or the viewing of the LDAP client state changes.

Syntax:

```
LDAP+enable ?
  frame
  trace
```

3.2.4.1 enable frame

Enables the viewing of the bytes exchanged with the LDAP server (through an event). By default this option is disabled.

Syntax:

```
LDAP+enable frame
```

Example:

```
LDAP+enable frame
LDAP+
```

3.2.4.2 enable trace

Enables the viewing of the LDAP client state changes that occur when performing server requests. By default this option is disabled.

Syntax:

```
LDAP+enable trace
```

Example:

```
LDAP+enable trace
LDAP+
```

3.2.5 LIST

Displays the enabled or disabled state of the viewing of the bytes exchanged with the server and the state changes, the established LDAP sessions, detailed client state information and the success or error results obtained upon requesting the server to perform operations.

Syntax:

```
LDAP+list ?
  enabled-events
  session
  trace
```

3.2.5.1 list enabled-events

Displays the enabled or disabled state of the viewing of the bytes exchanged with the server and the LDAP client state changes.

Syntax:

```
LDAP+list enabled-events
```

Example:

```
LDAP+list enabled-events
Trace States      : NO
Trace Sent Frames : NO
LDAP+
```

3.2.5.2 list session

Displays the established LDAP sessions.

Syntax:

```
LDAP+list session
```

Example:

```
LDAP+list session

SESSION 4
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         194.56.114.71   0

Last result information:
Last result: OK (0)
Last result DN: (null)
DN: CN=S-TRUST Qualified Root CA 2005-001:PN, O=Deutscher Sparkassen Verlag GmbH
, L=Stuttgart, ST=Baden-Wuerttemberg (BW), C=DEionPoint

TCP connection information:

Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------- ------
1         Initial                                   BindRequest 0x18E4C10

SESSION 5
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         195.55.117.103  0

Last result information:

Last result: OK (0)
Last result DN: (null)
DN: CN=CRL,CN=AC RAIZ DNIE,OU=DNIE,O=DIRECCION GENERAL DE LA POLICIA,C=ESag GmbH
, L=Stuttgart, ST=Baden-Wuerttemberg (BW), C=DEionPoint

TCP connection information:

Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------- ------
1         Initial                                   BindRequest 0x197CE10

SESSION 6
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         194.56.114.71   0

Last result information:
Last result: OK (0)
Last result DN: (null)
DN: CN=S-TRUST Qualified Root CA 2005-001:PN, O=Deutscher Sparkassen Verlag GmbH
```

```
, L=Stuttgart, ST=Baden-Wuerttemberg (BW), C=DEionPoint

TCP connection information:

Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------ ------
1         Initial                                 BindRequest 0x197C610

SESSION 1
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         195.55.117.103  0

Last result information:
Last result: OK (0)
Last result DN: (null)
DN: CN=CRL,CN=AC RAIZ DNIE,OU=DNIE,O=DIRECCION GENERAL DE LA POLICIA,C=ESag GmbH
, L=Stuttgart, ST=Baden-Wuerttemberg (BW), C=DEionPoint

TCP connection information:

Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------ ------
1         Initial                                 BindRequest 0x1960410

SESSION 2
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         194.56.114.71   0

Last result information:
Last result: OK (0)
Last result DN: (null)
DN: CN=S-TRUST Qualified Root CA 2005-001:PN, O=Deutscher Sparkassen Verlag GmbH
, L=Stuttgart, ST=Baden-Wuerttemberg (BW), C=DEionPoint

TCP connection information:
Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------ ------
1         Initial                                 BindRequest 0x1960210

SESSION 3
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         195.55.117.103  0

Last result information:
Last result: OK (0)
Last result DN: (null)
DN: CN=CRL,CN=AC RAIZ DNIE,OU=DNIE,O=DIRECCION GENERAL DE LA POLICIA,C=ESag GmbH
, L=Stuttgart, ST=Baden-Wuerttemberg (BW), C=DEionPoint

TCP connection information:
Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------ ------
1         Initial                                 BindRequest 0x1960C10

SESSION 7
```

```
-------------
Bind information:
IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         172.24.0.201    389

Last result information:
Last result: OK (0)
Last result DN: (null)
DN:

TCP connection information:
Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------- ------
2         BindCompleted                                    OK 0x197CC10

LDAP+
```

The following is displayed for each session:

- Session connection or establishment data:

  - Source IP address configured for the LDAP client (if 0.0.0.0 has been configured, the messages take the IP address of the output interface as source address).

  - IP address for the LDAP server with which the client is going to establish the sessions.

  - TCP port of the server to which the client addresses the LDAP messages.

- Data on the results of the last operation executed through this session

  - Results of the operation (OK or error).

  - *Last result DN*: *Distinguished Name* (DN) returned in the last search operation.

  - *DN*: *Distinguished Name* (DN) configured as base for the search operations (*baseObject*) is the DIT node where the search operation initiates.

  - Error message (if the last operation doesn't complete correctly).

- TCP connection data:

  - *Sequence*: LDAP request sequence number (*messageID*).

  - *State*: LDAP session state. This can be one of the following:

    - "Initial"
    - "TCPOpened"
    - "BindCompleted"
    - "SearchResponse"
    - "Searching"
    - "SearchNext"
    - "UnbindSend"
    - "Error100"
    - "Error101"
    - "Error102"
    - "Error103"
    - "Error104"
    - "Error105"
    - "Error106"
    - "Unkown"

  - *Status*: Request status. This can be in progress, completed correctly or with an error.

  - *Socket*: Identifier for the connection between client and LDAP server.

3.2.5.3 list trace

Displays detailed client state information together with the success or error results obtained when sending LDAP messages to the server whose IP address is given.

Syntax:

```
LDAP+list trace
```

Example:

```
LDAP+list trace

Time ****************************** 0h0m12s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m22s
172.24.0.201 BindCompleted OK
172.24.0.201 TCPOpened OK
172.24.0.201 Initial BindRequest
Time ****************************** 0h0m5s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
```

```
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
194.56.114.71 Initial BindRequest
Time ****************************** 0h0m4s
194.56.114.71 UnbindSend TCPClose
194.56.114.71 UnbindSend BindRequest
194.56.114.71 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
Time ****************************** 0h0m27s
195.55.117.103 Initial BindRequest
Time ****************************** 0h0m4s
195.55.117.103 UnbindSend TCPClose
195.55.117.103 UnbindSend BindRequest
195.55.117.103 Initial LDAPSTATUS_RECEIVE_TIMEOUTOUT. NO ANSWER
LDAP+
```

3.2.6 SEARCH

Induces the LDAP client to send a search operation request o the LDAP server. On entering this command, messages appear on the console requesting the following:

- Base DN (*baseObject*) for the search operation. When this is blank, the indicated DN is used (which is the one used in the last search), and if the “NULL” value is selected, the previous DN is deleted and the search is performed with a blank DN.
- Search filters, i.e., requirements that an entry must comply with to be returned by the search operation.
- Whether you want to queue more than one search request.

Syntax:

```
LDAP+search
```

In LDAP version 2, the client-server connection must already be established before performing an operation, i.e., the ‘**bind**’ command must have been previously executed. If you try to perform a search operation without an established LDAP session, the following error is returned:

```
LDAP+search
 ERROR.
```

```
[SEARCH]: Unable to perform this operation. Bind First
CLI Error: Command error
LDAP+
```

Example:

```
LDAP+search
Empty value will search with () as DN
"NULL" value will clean DN
DN : []? CN=Certificate Revocation 1, O=Telekom-Control-Kommission,C=AT
Data Type  String: [objectclass]? certificaterevocationlist;binary
Data Value String: []? *

Do you want to queue another search(Yes/No)? n
Searching ....
 OK.
Object Name: CN=Certificate Revocation 1,O=Telekom-Control-Kommission,C=AT
ATTRIBUTE                         VALUE
------------------------------------------------------------
objectclass                     :pkiUser
objectclass                     :organization
objectclass                     :top
objectclass                     :pkiCA
cn                              :Certificate Revocation 1
cacertificate;binary            : 30820433 3082031B A0030201 02020105
                                  300D0609 2A864886 F70D0101 05050030
                                  5D310B30 09060355 04061302 41543123
                                  30210603 55040A13 1A54656C 656B6F6D
                                  2D436F6E 74726F6C 2D4B6F6D 6D697373
                                  696F6E31 29302706 03550403 13205465
                                  6C656B6F 6D2D436F 6E74726F 6C2D4B6F
                                  6D6D6973 73696F6E 20546F70 2031301E
                                  170D3032 30393234 31363038 30305A17
                                  0D303530 39323431 36303830 305A3055
                                  310B3009 06035504 06130241 54312330
                                  21060355 040A131A 54656C65 6B6F6D2D
                                  436F6E74 726F6C2D 4B6F6D6D 69737369
                                  6F6E3121 301F0603 55040313 18436572
                                  74696669 63617465 20526576 6F636174
                                  696F6E20 31308201 22300D06 092A8648
                                  86F70D01 01010500 0382010F 00308201
                                  0A028201 0100C864 EB8F891F B930324C
                                  954AC409 7A5455BA 054E19BC 55CFC9B9
                                  EDB6D09E 3573011B 00E44D57 83E717BA
                                  389F39C8 AB476067 62B8AB20 F3E614D7
                                  0355E362 F41DE2F1 09F7CF90 C47F3764
                                  77975961 B34D4A5D 6F12B9A1 DA1F084A
                                  C72CA514 75DAB1CE 1706B492 767898E9
                                  974EE0BA 39A0047A 3467FAB2 415DDFC3
                                  659FC7EA 0CE2A76B C59AFAF5 18C5B297
                                  7936C421 FE66B977 3B38B414 2F40D2AB
                                  93699E6A 9D470C6F 361F6A4F B3506264
                                  5A1FDC34 19A142B0 5C8558A9 4E411356
                                  07D5B1CC 4C3EF77A 2DDF501D 34109598
                                  EC652EE1 C15687D2 D2A47BF3 FB3078DB
                                  618AB075 8663DD5F F018EF83 DC2AD61B
                                  56F78662 BC8D8505 704BA9F9 D08AD7E3
                                  6B24B0C5 92610203 010001A3 82010430
                                  82010030 1F060355 1D230418 30168014
                                  65CD574E BB9DD3FF A6BCBB79 209DE9E0
                                  4653ED0B 301D0603 551D0E04 16041412
                                  1BAE8479 927CE6B8 504203AA 42B4B38E
                                  DBEAA130 37060355 1D1F0430 302E302C
                                  A02AA028 86266874 74703A2F 2F777777
                                  2E736967 6E617475 722E7274 722E6174
                                  2F637572 72656E74 2E63726C 300C0603
                                  551D2404 05300380 01003056 0603551D
                                  20044F30 4D304B06 092A2800 15000100
                                  0100303E 303C0608 2B060105 05070201
                                  16306874 74703A2F 2F777777 2E736967
                                  6E617475 722E7274 722E6174 2F64652F
                                  64697265 63746F72 792F6370 732E6874
                                  6D6C300F 0603551D 130101FF 04053003
                                  0101FF30 0E060355 1D0F0101 FF040403
                                  02010230 0D06092A 864886F7 0D010105
                                  05000382 01010074 E8C30D13 B0C13073
                                  ED55FB12 7C6D8E8E EC7F32B8 B7D5EC61
                                  43FA811C C98D3B45 B0CDADA2 0E3D6073
                                  9AF4921A C175BEC0 BA6A976E 701923D4
                                  C21AAC02 365D0C02 B0C1F538 B9F162E2
                                  ED597347 DF000E45 9173764E 1EBC4756
                                  ACC238D4 32BD9BE6 4CF51841 73B99EB0
                                  78BD6ED6 D91D44EC 7B12EBAC 589F390C
                                  2AD68AC7 86A1EC92 C9C9943E F79F9795
                                  A922E160 D74C4FBA 4874C8EB 0EC12066
                                  EEA702E3 A13924E5 62A8C71A 4611F5AC
                                  AD357766 41A2503C 734F8326 5DADBE2C
                                  32F5E7D8 A9FFE20D EEE3D9A9 65E8EF1A
                                  7D2DB320 6CC94E93 4A1B0198 54EDDA75
                                  C4A8053E F468B139 CC17F0C0 46558F3B
                                  F3E0E0B7 968AC2B8 B1EF8A6A 87731E1A
                                  873B8FD5 9C7B3A
cacertificate;binary            : 30820433 3082031B A0030201 0202010C
                                  300D0609 2A864886 F70D0101 05050030
                                  5D310B30 09060355 04061302 41543123
                                  30210603 55040A13 1A54656C 656B6F6D
                                  2D436F6E 74726F6C 2D4B6F6D 6D697373
                                  696F6E31 29302706 03550403 13205465
                                  6C656B6F 6D2D436F 6E74726F 6C2D4B6F
                                  6D6D6973 73696F6E 20546F70 2031301E
                                  170D3035 30393133 31363230 30305A17
                                  0D313030 39313331 36323030 305A3055
                                  310B3009 06035504 06130241 54312330
                                  21060355 040A131A 54656C65 6B6F6D2D
                                  436F6E74 726F6C2D 4B6F6D6D 69737369
                                  6F6E3121 301F0603 55040313 18436572
                                  74696669 63617465 20526576 6F636174
                                  696F6E20 31308201 22300D06 092A8648
                                  86F70D01 01010500 0382010F 00308201
                                  0A028201 0100C864 EB8F891F B930324C
                                  954AC409 7A5455BA 054E19BC 55CFC9B9
                                  EDB6D09E 3573011B 00E44D57 83E717BA
                                  389F39C8 AB476067 62B8AB20 F3E614D7
                                  0355E362 F41DE2F1 09F7CF90 C47F3764
                                  77975961 B34D4A5D 6F12B9A1 DA1F084A
                                  C72CA514 75DAB1CE 1706B492 767898E9
                                  974EE0BA 39A0047A 3467FAB2 415DDFC3
                                  659FC7EA 0CE2A76B C59AFAF5 18C5B297
                                  7936C421 FE66B977 3B38B414 2F40D2AB
                                  93699E6A 9D470C6F 361F6A4F B3506264
                                  5A1FDC34 19A142B0 5C8558A9 4E411356
                                  07D5B1CC 4C3EF77A 2DDF501D 34109598
                                  EC652EE1 C15687D2 D2A47BF3 FB3078DB
                                  618AB075 8663DD5F F018EF83 DC2AD61B
                                  56F78662 BC8D8505 704BA9F9 D08AD7E3
                                  6B24B0C5 92610203 010001A3 82010430
                                  82010030 1F060355 1D230418 30168014
                                  65CD574E BB9DD3FF A6BCBB79 209DE9E0
                                  4653ED0B 301D0603 551D0E04 16041412
                                  1BAE8479 927CE6B8 504203AA 42B4B38E
                                  DBEAA130 37060355 1D1F0430 302E302C
                                  A02AA028 86266874 74703A2F 2F777777
                                  2E736967 6E617475 722E7274 722E6174
                                  2F637572 72656E74 2E63726C 300C0603
                                  551D2404 05300380 01003056 0603551D
                                  20044F30 4D304B06 092A2800 15000100
                                  0100303E 303C0608 2B060105 05070201
                                  16306874 74703A2F 2F777777 2E736967
                                  6E617475 722E7274 722E6174 2F64652F
                                  64697265 63746F72 792F6370 732E6874
                                  6D6C300F 0603551D 130101FF 04053003
                                  0101FF30 0E060355 1D0F0101 FF040403
                                  02010230 0D06092A 864886F7 0D010105
                                  05000382 01010018 2C98DD82 9085CAE0
                                  2949B086 AF7801EA 7282F74C 7AD9479E
                                  62199DEC 18C58E2D E08E8100 781AECCD
                                  08F785CA 6082A2CE A6DD57CE A867628F
                                  8736430E B6F299CF 2D32CAD2 D7B6D5BD
                                  0D9D24E5 F9A75148 9B619489 2ED50593
                                  E51EE748 57D36A2F A9B6CB71 723F22AC
                                  2942B8B2 20013B75 77F331C7 9BC22C63
                                  B6267951 F481E5FC 79A023CE 7AAB9588
                                  A894C038 B7BE6A2D 84ACB7C0 DA48D2B9
                                  0DF08D03 CF29CA9D 88F5A0ED 5FB688A6
                                  4CD0CF75 A72596F8 71950606 297B6E16
                                  D57222D2 90F35C7B 36B0CAF2 2F3F4EE4
                                  DBD82430 04A3FC50 1C2F91A5 C52F9890
                                  0901ADDF 7BE59D58 35C8FC29 8A08F76F
                                  F4BFD8E9 12E7ECF4 CC366B57 4583931C
                                  E8A5CFFE CC31CE
certificaterevocationlist;binary: 30820364 3082024C 02010130 0D06092A
                                  864886F7 0D010105 05003055 310B3009
                                  06035504 06130241 54312330 21060355
                                  040A131A 54656C65 6B6F6D2D 436F6E74
                                  726F6C2D 4B6F6D6D 69737369 6F6E3121
                                  301F0603 55040313 18436572 74696669
                                  63617465 20526576 6F636174 696F6E20
                                  31170D30 38313032 38313430 3030375A
                                  170D3038 31303239 31343030 30375A30
                                  82017C30 81870201 11170D30 34313031
                                  32313335 3131335A 30733065 0603551D
                                  1D0101FF 045B3059 A4573055 310B3009
                                  06035504 06130241 54312330 21060355
                                  040A131A 54656C65 6B6F6D2D 436F6E74
                                  726F6C2D 4B6F6D6D 69737369 6F6E3121
                                  301F0603 55040313 18436572 74696669
                                  63617469 6F6E2053 65727669 63657320
                                  31300A06 03551D15 04030A01 05302002
                                  0115170D 30353031 31333136 32303235
                                  5A300C30 0A060355 1D150403 0A010430
                                  2002011A 170D3034 31303231 31323336
                                  32375A30 0C300A06 03551D15 04030A01
                                  04302002 011D170D 30353031 31333136
                                  31393036 5A300C30 0A060355 1D150403
                                  0A010430 81890201 01170D30 32303932
                                  37313435 3933325A 30753067 0603551D
                                  1D0101FF 045D305B A4593057 310B3009
                                  06035504 06130241 54312F30 2D060355
                                  040A1326 52756E64 66756E6B 20756E64
                                  2054656C 656B6F6D 20526567 756C6965
                                  72756E67 732D476D 62483117 30150603
                                  55040313 0E525452 20536572 76696365
                                  73203130 0A060355 1D150403 0A0104A0
                                  43304130 11060355 1D1C0101 FF040730
                                  05A40301 01FF301F 0603551D 23041830
                                  16801412 1BAE8479 927CE6B8 504203AA
                                  42B4B38E DBEAA130 0B060355 1D140404
                                  0202458C 300D0609 2A864886 F70D0101
                                  05050003 82010100 80B0094C 28C9DE3F
                                  8C7BA3FE 7870FB3D FFF2086C 54C918B4
                                  6B6E8C7A F59122AE 6C8C882A F9B98B44
                                  DEF16CBD A9E93A68 AC8B58CF 062BAABE
                                  DD83EB28 D62B3966 9F6B29C6 AEB86EC5
                                  AB9DA253 D5363A2D 73687CB7 D9CB265C
                                  87B68054 2FFCB218 5578BF69 D2CB2C6F
                                  AB0C50BB 336FED35 CFA997EA 6734C34C
                                  1C171CCE 78D48122 42EB9A45 7FEE7487
                                  84D2FCBB BA7762E8 2E4859E0 CFEE33E3
                                  06298575 76B32EB1 6547B902 45D2CCD6
                                  D39308A6 862615ED DE9ED1EA 5B34F7DA
                                  564C3E21 E81AC69E 7C03764B EF329F0A
                                  C2304B29 420DD7D4 AC0A0338 DFD1ED12
                                  EB235343 2E8CB76F 9316E306 137AFCD8
                                  F10F4510 51B136FA C59160A7 F2567C05
                                  B682D483 C7A99E9C
postaladdress                   :A-1060 Wien, Mariahilfer StraÃŸe 77-79
labeleduri                      :http://www.signatur.rtr.at/
mail                            :signatur@signatur.rtr.at
telephonenumber                 :+43 1 58 058 0
description                     :Telekom-Control-Kommission
----------------------------------
========================================
LDAP+
```

In this example, we have set the search base to be the object whose DN is “CN=Certificate Revocation 1, O=Telekom-Control-Kommission,C=AT”, the search scope was configured to only search the base object, i.e., this DN, the list of attributes to be included in the search was left empty, so the server returns the attributes for each found entry, and the search filter is that the “certificaterevocationlist;binary” attribute takes any value.

On receiving the search request, the server begins executing the search in the directory and returns the result to the client. After receiving the response message, the entries found in said search are shown on the console (only 1 in this case as only the base object was searched) with, in this case, all their attributes since we didn't define a list of desired attributes.

If you list the session, you can see whether the operation was successful or produced an error:

```
LDAP+list session

SESSION 1
-------------
Bind information:

IP Source       IP Destination  Server Port
--------------- --------------- -----------
0.0.0.0         81.15.158.132   389

Last result information:
Last result: OK (0)
Last result DN: (null)
DN: CN=Certificate Revocation 1, O=Telekom-Control-Kommission,C=AT

TCP connection information:

Sequence  State                 Status                          Socket
--------  --------------------  ------------------------------ ------
4         BindCompleted                                    OK 0x192F210
LDAP+
```

3.2.7 UNBIND

Ends a session between the client and server. The client sends an *unbindRequest* to the server. This operation does not require confirmation from the server to the client: the client sends the *unbindRequest* message and assumes that the session has ended, while the server, on receiving said message, also assumes the session is terminated and drops any pending requests. If an error occurs, it is shown on the screen together with the cause of the error.

Syntax:

```
LDAP+unbind
```

Example 1:

```
LDAP+unbind

Disconnecting ...
LDAP+
```

This example shows that the previous established LDAP session has terminated correctly.

Example 2:

```
LDAP+unbind
 ERROR.
[UNBIND]: Unable to perform this operation. Bind First
LDAP+
```

In this example, an attempt to terminate a session has generated an error because the session is not established.

3.2.8 EXIT

Use this command to exit the LDAP client monitoring menu and return to the main monitoring menu (+).

Syntax:

```
LDAP+exit
```

Example:

```
LDAP+exit
+
```