Applies to **SUSE Linux Enterprise Server 12 SP5**

5 LDAP—A Directory Service

The Lightweight Directory Access Protocol (LDAP) is a set of protocols designed to access and maintain information directories. LDAP can be used for user and group management, system configuration management, address management, and more. This chapter provides a basic understanding of how OpenLDAP works.

In a network environment, it is crucial to keep important information structured and to serve it quickly. A directory service keeps information available in a well-structured and searchable form.

Ideally, a central server stores the data in a directory and distributes it to all clients using a well-defined protocol. The structured data allow a wide range of applications to access them. A central repository reduces the necessary administrative effort. The use of an open and standardized protocol such as LDAP ensures that as many client applications as possible can access such information.

A directory in this context is a type of database optimized for quick and effective reading and searching:

- To make multiple concurrent reading accesses possible, the number of updates is usually very low. The number of read and write accesses is often limited to a few users with administrative privileges. In contrast, conventional databases are optimized for accepting the largest possible data volume in a short time.
- When static data is administered, updates of the existing data sets are very rare. When working with dynamic data, especially when data sets like bank accounts or accounting are concerned, the consistency of the data is of primary importance. If an amount should be subtracted from one place to be added to another, both operations must happen concurrently, within one *transaction*, to ensure balance over the data stock. Traditional relational databases usually have a very strong focus

stock. Traditional relational databases usually have a very strong focus on data consistency, such as the referential integrity support of transactions. Conversely, short-term inconsistencies are usually acceptable in LDAP directories. LDAP directories often do not have the same strong consistency requirements as relational databases.

The design of a directory service like LDAP is not laid out to support complex update or query mechanisms. All applications are guaranteed to access this service quickly and easily.

5.1 LDAP versus NIS

Unix system administrators traditionally use NIS (Network Information Service) for name resolution and data distribution in a network. The configuration data contained in the files `group`, `hosts`, `mail`, `netgroup`, `networks`, `passwd`, `printcap`, `protocols`, `rpc`, and `services` in the `/etc` directory is distributed to clients all over the network. These files can be maintained without major effort because they are simple text files. The handling of larger amounts of data, however, becomes increasingly difficult because of nonexistent structuring. NIS is only designed for Unix platforms, and is not suitable as a centralized data administration tool in heterogeneous networks.

Unlike NIS, the LDAP service is not restricted to pure Unix networks. Windows™ servers (starting with Windows 2000) support LDAP as a directory service. The application tasks mentioned above are additionally supported in non-Unix systems.

The LDAP principle can be applied to any data structure that needs to be centrally administered. A few application examples are:

- Replacement for the NIS service
- Mail routing (postfix)
- Address books for mail clients, like Mozilla Thunderbird, Evolution, and Outlook
- Administration of zone descriptions for a BIND 9 name server
- User authentication with Samba in heterogeneous networks

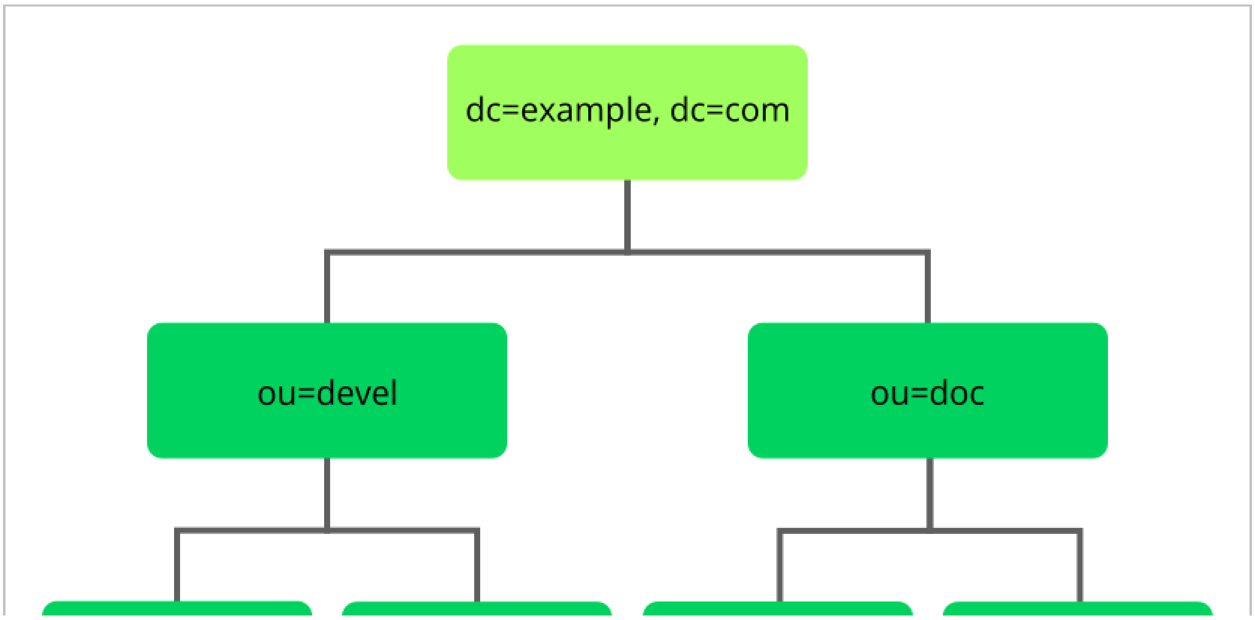
This list can be extended because LDAP is extensible, unlike NIS. The clearly-defined hierarchical structure of the data simplifies the administration of large amounts of data, as it can be searched more easily.

5.2 Structure of an LDAP Directory Tree

To get background knowledge on how an LDAP server works and how the data is stored, it is vital to understand the way the data is organized on the server and how this structure enables LDAP to provide fast access to the data. To successfully operate an LDAP setup, you also need to be familiar with some basic LDAP terminology. This section introduces the basic layout of an LDAP directory tree, and provides the basic terminology used with regard to LDAP. Skip this introductory section if you already have some LDAP background knowledge and only want to learn how to set up an LDAP environment in SUSE Linux Enterprise Server. Read on at Section 5.5, “Manually Configuring an LDAP Server”.

An LDAP directory has a tree structure. All entries (called objects) of the directory have a defined position within this hierarchy. This hierarchy is called the *directory information tree* (DIT). The complete path to the desired entry, which unambiguously identifies it, is called the *distinguished name* or DN. A single node along the path to this entry is called *relative distinguished name* or RDN.

The relations within an LDAP directory tree become more evident in the following example, shown in Figure 5.1, “Structure of an LDAP Directory”.



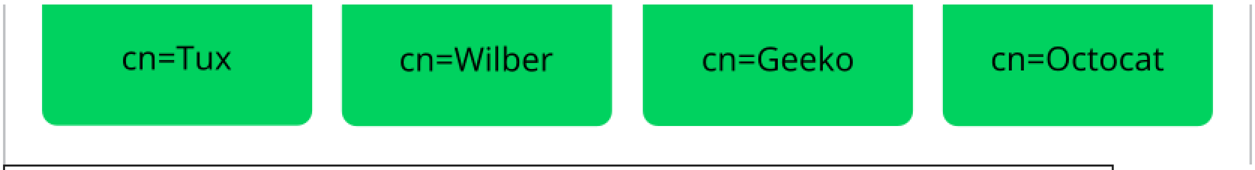


FIGURE 5.1: STRUCTURE OF AN LDAP DIRECTORY

The complete diagram is a fictional directory information tree. The entries on three levels are depicted. Each entry corresponds to one box in the image. The complete, valid *distinguished name* for the fictional employee `Geeko Linux`, in this case, is `cn=Geeko Linux,ou=doc,dc=example,dc=com`. It is composed by adding the RDN `cn=Geeko Linux` to the DN of the preceding entry `ou=doc,dc=example,dc=com`.

The types of objects that can be stored in the DIT are globally determined following a *Schema*. The type of an object is determined by the *object class*. The object class determines what attributes the relevant object must or can be assigned. The Schema, therefore, must contain definitions of all object classes and attributes used in the desired application scenario. There are a few common Schemas (see RFC 2252 and 2256). The LDAP RFC defines a few commonly used Schemas (see for example, RFC4519). Additionally, Schemas are available for many other use cases (for example, Samba or NIS replacement). It is, however, possible to create custom Schemas or to use multiple Schemas complementing each other (if this is required by the environment in which the LDAP server should operate).

Table 5.1, "Commonly Used Object Classes and Attributes" offers a small overview of the object classes from `core.schema` and `inetorgperson.schema` used in the example, including required attributes (Req. Attr.) and valid attribute values.

TABLE 5.1: COMMONLY USED OBJECT CLASSES AND ATTRIBUTES

Object Class	Meaning	Example Entry	Req. Attr.
`dcObject`	*domainComponent* (name components of the domain)	example	dc
`organizationalUnit`	*organizationalUnit* (organizational unit)	doc	ou
`inetOrgPerson`	*inetOrgPerson* (person-related data for the intranet or Internet)	Geeko Linux	sn and cn

Example 5.1, "Excerpt from schema.core" shows an excerpt from a Schema directive with explanations.

EXAMPLE 5.1: EXCERPT FROM SCHEMA.CORE

```
attributetype (2.5.4.11 NAME ( 'ou' 'organizationalUnitName') ①
       DESC 'RFC2256: organizational unit this object belongs to' ②
       SUP name ) ③

objectclass ( 2.5.6.5 NAME 'organizationalUnit' ④
       DESC 'RFC2256: an organizational unit' ⑤
       SUP top STRUCTURAL ⑥
       MUST ou ⑦
MAY (userPassword $ searchGuide $ seeAlso $ businessCategory ⑧
  $ x121Address $ registeredAddress $ destinationIndicator
  $ preferredDeliveryMethod $ telexNumber
  $ teletexTerminalIdentifier $ telephoneNumber
  $ internationaliSDNNumber $ facsimileTelephoneNumber
  $ street $ postOfficeBox $ postalCode $ postalAddress
  $ physicalDeliveryOfficeName
  $ st $ l $ description) )
  ...
```

The attribute type `organizationalUnitName` and the corresponding object class `organizationalUnit` serve as an example here.

① The name of the attribute, its unique OID (*object identifier*) (numerical), and the abbreviation of the attribute.

② A brief description of the attribute with `DESC`. The corresponding RFC, on which the definition is based, is also mentioned here.

③ `SUP` indicates a superordinate attribute type to which this attribute belongs.

④ The definition of the object class `organizationalUnit` begins—the same as in the definition of the attribute—with an OID and the name of the object class.

⑤ A brief description of the object class.

⑥ The `SUP top` entry indicates that this object class is not subordinate to another object class.

⑦ With `MUST` list all attribute types that must be used with an object of the type `organizationalUnit`.

⑧ With `MAY` list all attribute types that are permitted with this object class.

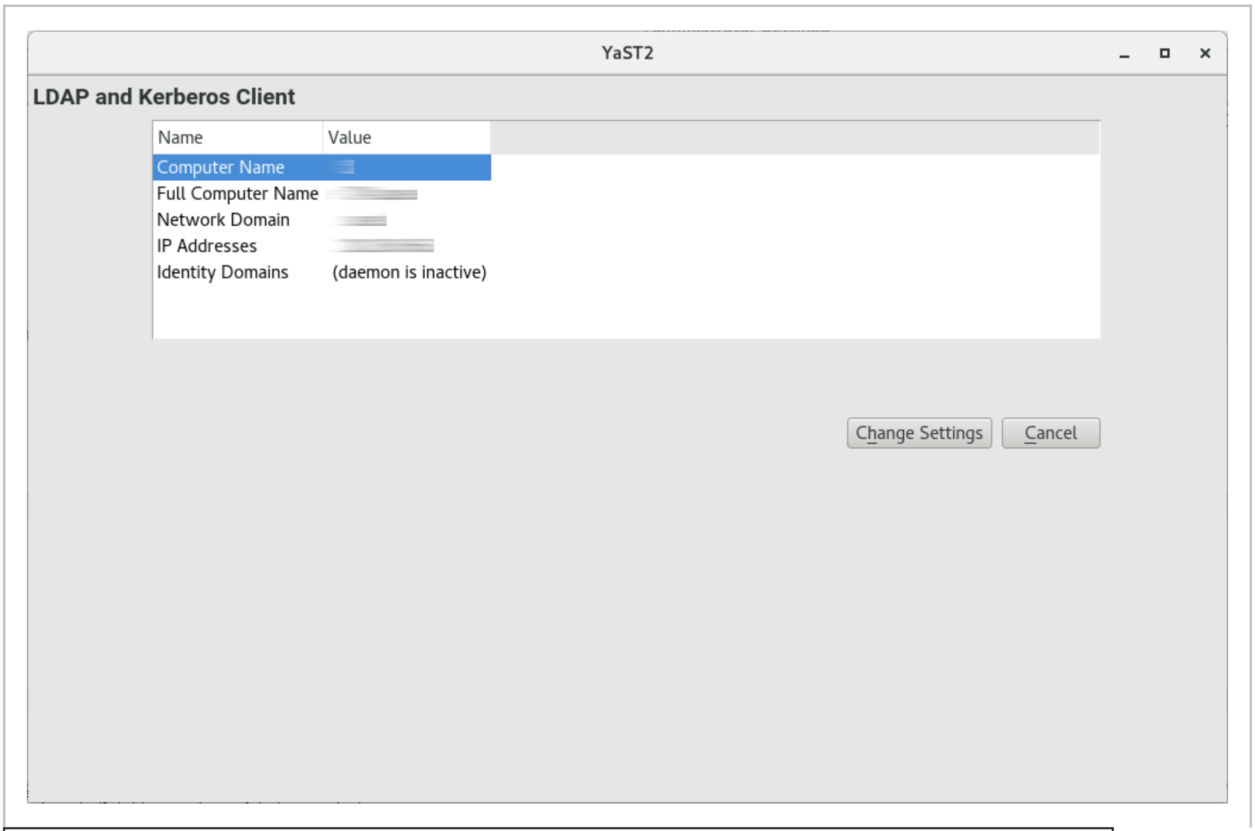
A very good introduction to the use of Schemas can be found in the OpenLDAP documentation (`openldap2-doc`). When installed, find it in `/usr/share/doc/packages/openldap2/adminguide/guide.html`.

5.3 Configuring an LDAP Client with YaST

YaST includes the module *LDAP and Kerberos Client* that helps define authentication scenarios involving either LDAP or Kerberos.

It can also be used to join Kerberos and LDAP separately. However, in many such cases, using this module may not be the first choice, such as for joining Active Directory (which uses a combination of LDAP and Kerberos). For more information, see Section 4.2, “Configuring an Authentication Client with YaST”.

Start the module by selecting *Network Services* › *LDAP and Kerberos Client*.



LDAP and Kerberos Client

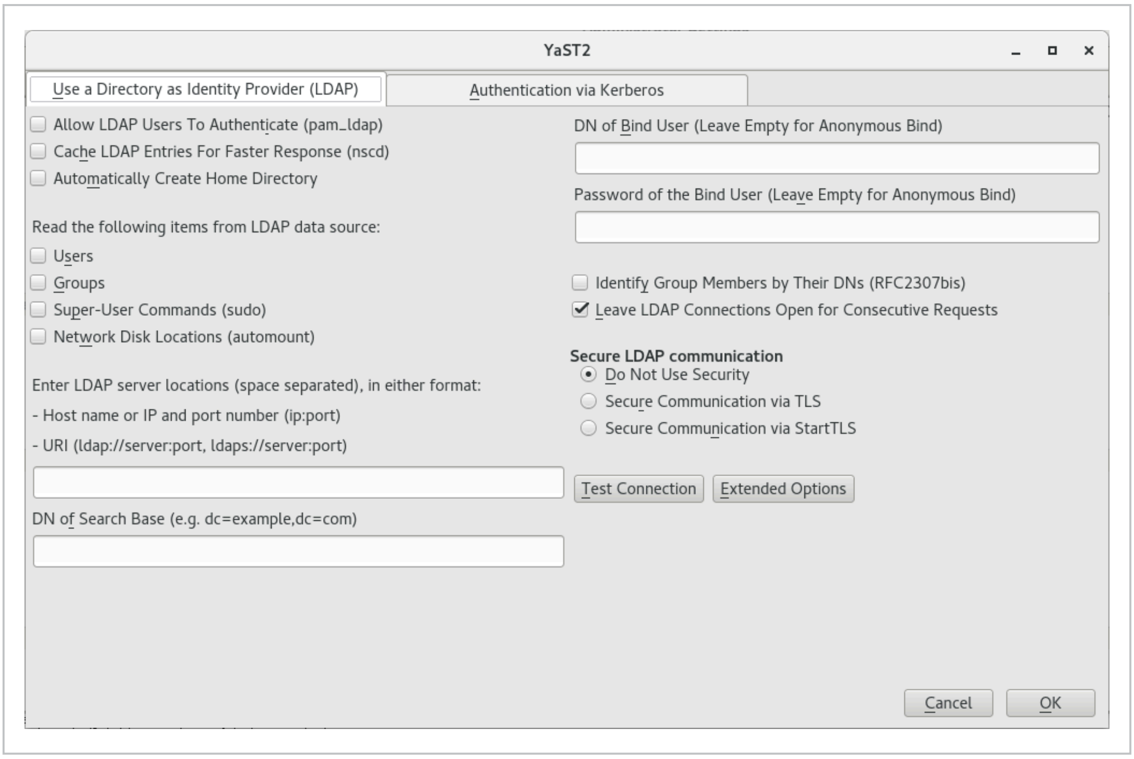
Name	Value
Computer Name	
Full Computer Name	
Network Domain	
IP Addresses	
Identity Domains	(daemon is inactive)

FIGURE 5.2: *LDAP AND KERBEROS CLIENT* WINDOW

To configure an LDAP client, follow the procedure below:

1. In the window *LDAP and Kerberos Client*, click *Change Settings*.

   Make sure that the tab *Use a Directory as Identity Provider (LDAP)* is chosen.



YaST2
Use a Directory as Identity Provider (LDAP)
Authentication via Kerberos
Test Connection
Extended Options
Cancel
OK

□ Allow LDAP Users To Authenticate (pam_ldap)
□ Cache LDAP Entries For Faster Response (nscd)
□ Automatically Create Home Directory

Read the following items from LDAP data source:
□ Users
□ Groups
□ Super-User Commands (sudo)
□ Network Disk Locations (automount)

Enter LDAP server locations (space separated), in either format:
- Host name or IP and port number (ip:port)
- URI (ldap://server:port, ldaps://server:port)

DN of Search Base (e.g. dc=example,dc=com)

DN of Bind User (Leave Empty for Anonymous Bind)

Password of the Bind User (Leave Empty for Anonymous Bind)

□ Identify Group Members by Their DNs (RFC2307bis)
☑ Leave LDAP Connections Open for Consecutive Requests

Secure LDAP communication
● Do Not Use Security
○ Secure Communication via TLS
○ Secure Communication via StartTLS

2. Specify one or more LDAP server URLs, host names, or IP addresses under *Enter LDAP server locations*. When specifying multiple addresses, separate them with spaces.

3. Specify the appropriate LDAP distinguished name (DN) under *DN of Search Base*. For example, a valid entry could be `dc=example,dc=com` .

4. If your LDAP server supports TLS encryption, choose the appropriate security option under *Secure LDAP Connection*.

   To first ask the server whether it supports TLS encryption and be able to downgrade to an unencrypted connection if it does not, use *Secure Communication via StartTLS*.

5. Activate other options as necessary:

   - You can *Allow users to authenticate via LDAP* and *Automatically Create Home Directories* on the local computer for them.
   - Use *Cache LDAP Entries For Faster Response* to cache LDAP entries locally. However, this bears the danger that entries can be

slightly out of date.

- Specify the types of data that should be used from the LDAP source, such as *Users* and *Groups*, *Super-User Commands*, and *Network Disk Locations* (network-shared drives that can be automatically mounted on request).
- Specify the distinguished name (DN) and password of the user under whose name you want to bind to the LDAP directory in *DN of Bind User* and *Password of the Bind User*.

  Otherwise, if the server supports it, you can also leave both text boxes empty to bind anonymously to the server.

  > **Warning: Authentication Without Encryption**
  >
  > When using authentication without enabling transport encryption using TLS or StartTLS, the password will be transmitted in the clear.

Under *Extended Options*, you can additionally configure timeouts for BIND operations.

6. To check whether the LDAP connection works, click *Test Connection*.
7. To leave the dialog, click *OK*. Then wait for the setup to complete.

   Finally, click *Finish*.

5.4 Configuring LDAP Users and Groups in YaST

The actual registration of user and group data differs only slightly from the procedure when not using LDAP. The following instructions relate to the administration of users. The procedure for administering groups is analogous.

1. Access the YaST user administration with *Security and Users › User and Group Management*.
2. Use *Set Filter* to limit the view of users to the LDAP users and enter the password for Root DN.
3. Click *Add* to enter the user configuration. A dialog with four tabs opens:

   a. Specify the user's name, login name, and password in the *User Data* tab.

   b. Check the *Details* tab for the group membership, login shell, and home directory of the new user. If necessary, change the default to values that better suit your needs.

   c. Modify or accept the default *Password Settings*.

   d. Enter the *Plug-Ins* tab, select the LDAP plug-in, and click *Launch* to configure additional LDAP attributes assigned to the new user.

4. Click *OK* to apply your settings and leave the user configuration.

The initial input form of user administration offers *LDAP Options*. This allows you to apply LDAP search filters to the set of available users. Alternatively open the module for configuring LDAP users and groups by selecting *LDAP User and Group Configuration*.

5.5 Manually Configuring an LDAP Server

YaST uses OpenLDAP's dynamic configuration database (`back-config`) to store the LDAP server's configuration. For details about the dynamic configuration back-end, see the `slapd-config(5)` man page or the OpenLDAP Software 2.4 Administrator's Guide located at `/usr/share/doc/packages/openldap2/guide/admin/guide.html` on your system if the `openldap2` package is installed.

> **Tip: Upgrading an Old OpenLDAP Installation**
>
> YaST does not use `/etc/openldap/slapd.conf` to store the OpenLDAP configuration anymore. In case of a system upgrade, a copy of the original `/etc/openldap/slapd.conf` file will get created as `/etc/openldap/slapd.conf.YaSTsave`.

To conveniently access the configuration back-end, you use SASL external authentication. For example, the following **`ldapsearch`** command executed as `root` can show the complete **`slapd`** configuration:

```
ldapsearch -Y external -H ldapi:/// -b cn=config
```

Note: LDAP Server Is Part of the Authentication Server

Basic LDAP Server initialization and configuration can be done within the Authentication Server YaST module. For more information, see Section 4.1, “Configuring an Authentication Server with YaST”.

When the LDAP server is fully configured and all desired entries have been made according to the pattern described in Section 5.6, “Manually Administering LDAP Data”, start the LDAP server as `root` by entering **`sudo systemctl start slapd`**. To stop the server manually, enter the command **`sudo systemctl stop slapd`**. Query the status of the running LDAP server with **`sudo systemctl status slapd`**.

Use the YaST *Services Manager*, described in Section 14.4, “Managing services with YaST”, to have the server started and stopped automatically on system bootup and shutdown. You can also create the corresponding links to the start and stop scripts with the **`systemctl`** commands as described in Section 14.2.1, “Managing Services in a Running System”.

5.6 Manually Administering LDAP Data

OpenLDAP offers a series of tools for the administration of data in the LDAP directory. The four most important tools for adding to, deleting from, searching through and modifying the data stock are explained in this section.

5.6.1 Inserting Data into an LDAP Directory

Once your LDAP server is correctly configured (it features appropriate entries for `suffix`, `directory`, `rootdn`, `rootpw` and `index`), proceed to entering records. OpenLDAP offers the **`ldapadd`** command for this task. If possible, add the objects to the database in bundles (for practical reasons). LDAP can process the LDIF format (LDAP data interchange format) for this. An LDIF file is a simple text file that can contain an arbitrary number of attribute and value pairs. The LDIF file for creating a rough framework for the example in Figure 5.1, “Structure of an LDAP Directory” would look like the one in Example 5.2, “An LDIF File”.

Important: Encoding of LDIF Files

LDAP works with UTF-8 (Unicode). Umlauts must be encoded correctly. Otherwise, avoid umlauts and other special characters or use **`iconv`** to convert the input to UTF-8.

convert the input to UTF-8.

EXAMPLE 5.2: AN LDIF FILE

```
# The Organization
dn: dc=example,dc=com
objectClass: dcObject
objectClass: organization
o: Example dc: example

# The organizational unit development (devel)
dn: ou=devel,dc=example,dc=com
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=example,dc=com
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=example,dc=com
objectClass: organizationalUnit
ou: it
```

Save the file with the `.ldif` suffix then pass it to the server with the following command:

```
ldapadd -x -D DN_OF_THE_ADMINISTRATOR -W -f FILE.ldif
```

`-x` switches off the authentication with SASL in this case. `-D` declares the user that calls the operation. The valid DN of the administrator is entered here, as it has been configured in `slapd.conf`. In the current example, this is `cn=Administrator,dc=example,dc=com`. `-W` circumvents entering the password on the command line (in clear text) and activates a separate password prompt. The `-f` option passes the file name. See the details of running **`ldapadd`** in Example 5.3, “ldapadd with example.ldif”.

EXAMPLE 5.3: LDAPADD WITH EXAMPLE.LDIF

```
ldapadd -x -D cn=Administrator,dc=example,dc=com -W -f example.ldif

Enter LDAP password:
adding new entry "dc=example,dc=com"
adding new entry "ou=devel,dc=example,dc=com"
adding new entry "ou=doc,dc=example,dc=com"
adding new entry "ou=it,dc=example,dc=com"
```

The user data of individuals can be prepared in separate LDIF files

The user data of individuals can be prepared in separate LDIF files. Example 5.4, “LDIF Data for Tux” adds `Tux` to the new LDAP directory.

EXAMPLE 5.4: LDIF DATA FOR TUX

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=example,dc=com

objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@example.com
uid: tux
telephoneNumber: +49 1234 567-8
```

An LDIF file can contain an arbitrary number of objects. It is possible to pass directory branches (entirely or in part) to the server in one go, as shown in the example of individual objects. If it is necessary to modify some data relatively often, a fine subdivision of single objects is recommended.

5.6.2 Modifying Data in the LDAP Directory

The tool **`ldapmodify`** is provided for modifying the data stock. The easiest way to do this is to modify the corresponding LDIF file and pass the modified file to the LDAP server. To change the telephone number of colleague Tux from `+49 1234 567-8` to `+49 1234 567-10`, edit the LDIF file like in Example 5.5, “Modified LDIF File tux.ldif”.

EXAMPLE 5.5: MODIFIED LDIF FILE TUX.LDIF

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Import the modified file into the LDAP directory with the following command:

```
ldapmodify -x -D cn=Administrator,dc=example,dc=com -W -f tux.ldif
```

Alternatively, pass the attributes to change directly to **`ldapmodify`** as follows:

1. Start **`ldapmodify`** and enter your password:

   ```
   ldapmodify -x -D cn=Administrator,dc=example,dc=com -W
   Enter LDAP password:
   ```

2. Enter the changes while carefully complying with the syntax in the order presented below:

> der presented below.
>
> ```
> dn: cn=Tux Linux,ou=devel,dc=example,dc=com
> changetype: modify
> replace: telephoneNumber
> telephoneNumber: +49 1234 567-10
> ```

For more information about **`ldapmodify`** and its syntax, see the **`ldapmodify`** man page.

5.6.3 Searching or Reading Data from an LDAP Directory

OpenLDAP provides, with **`ldapsearch`**, a command line tool for searching data within an LDAP directory and reading data from it. This is a simple query:

```
ldapsearch -x -b dc=example,dc=com "(objectClass=*)"
```

The `-b` option determines the search base (the section of the tree within which the search should be performed). In the current case, this is `dc=example,dc=com`. To perform a more finely-grained search in specific subsections of the LDAP directory (for example, only within the `devel` department), pass this section to **`ldapsearch`** with `-b`. `-x` requests activation of simple authentication. `(objectClass=*)` declares that all objects contained in the directory should be read. This command option can be used after the creation of a new directory tree to verify that all entries have been recorded correctly and the server responds as desired. For more information about the use of **`ldapsearch`**, see the `ldapsearch(1)` man page.

5.6.4 Deleting Data from an LDAP Directory

Delete unwanted entries with **`ldapdelete`**. The syntax is similar to that of the other commands. To delete, for example, the complete entry for `Tux Linux`, issue the following command:

```
ldapdelete -x -D cn=Administrator,dc=example,dc=com -W cn=Tux \
Linux,ou=devel,dc=example,dc=com
```

5.7 New negation feature in sudoers.ldap

If you are using sudoers.ldap, there is a useful change in `sudo` versions 1.9.9

and up. In versions older than 1.9.9, negation in sudoers.ldap does not work for the `sudoUser`, `sudoRunAsUser`, or `sudoRunAsGroup` attributes. For example:

```
# does not match all but joe
# instead, it does not match anyone
sudoUser: !joe


# does not match all but joe
# instead, it matches everyone including Joe
sudoUser: ALL
sudoUser: !joe
```

In **`sudo`** version 1.9.9 and higher, negation is enabled for the `sudoUser` attribute, so you may exclude individual users. See **`man 5 sudoers.ldap`** for more information.

5.8 For More Information

More complex subjects (like SASL configuration or establishment of a replicating LDAP server that distributes the workload among multiple slaves) were omitted from this chapter. Find detailed information about both subjects in the *OpenLDAP 2.4 Administrator's Guide*—see at OpenLDAP 2.4 Administrator's Guide.

The Web site of the OpenLDAP project offers exhaustive documentation for beginner and advanced LDAP users:

OpenLDAP Faq-O-Matic

A detailed question and answer collection applying to the installation, configuration, and use of OpenLDAP. Find it at http://www.openldap.org/faq/data/cache/1.html (http://www.openldap.org/faq/data/cache/1.html) ↗.

Quick Start Guide

Brief step-by-step instructions for installing your first LDAP server. Find it at http://www.openldap.org/doc/admin24/quickstart.html (http://www.openldap.org/doc/admin24/quickstart.html) ↗ or on an installed system in Section 2 of `/usr/share/doc/packages/openldap2/guide/admin/guide.html`.

OpenLDAP 2.4 Administrator's Guide

A detailed introduction to all important aspects of LDAP configuration, including access controls and encryption. See http://www.openldap.org/doc/admin24/ (http://www.openldap.org/doc/admin24/) ↗ or, on an installed system,

`/usr/share/doc/packages/openldap2/guide/admin/guide.html`.

Understanding LDAP

A detailed general introduction to the basic principles of LDAP: http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf (http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf) ↗.

Printed literature about LDAP:

- *LDAP System Administration* by Gerald Carter (ISBN 1-56592-491-6)
- *Understanding and Deploying LDAP Directory Services* by Howes, Smith, and Good (ISBN 0-672-32316-8)

The ultimate reference material for the subject of LDAP are the corresponding RFCs (request for comments), 2251 to 2256.