LDAP - Сервис директорий

Облегченный протокол доступа к каталогам (LDAP) - это набор протоколов, созданный для доступа и поддержания информационных каталогов. LDAP может использоваться для различных целей, например, управление пользователями и группами, управление системной конфигурацией или управление адресами. В этом разделе будет описание основное объяснение того, каким образом работает LDAP и как управлять данными LDAP с помощью YaST. Поскольку имеется несколько реализаций протокола LDAP, в этом разделе мы будет в основном ориентироваться на реализации openLDAP.

В сетевом окружении важно сохранить ценную информацию структурированной и быстро доступной. Это может быть сделано с помощью такого сервиса каталогов, как общие желтые страницы, хранящие информацию доступной в отличноструктурированной форме, приспособленной для быстрого поиска.

В идеальном случае, центральный сервер хранит данные в каталоге и распределяет их всем клиентам по определенному протоколу. Данные структурированы в таком виде, который позволяет широкому кругу приложений получить доступ к ним. Таким образом, нет необходимости для каждого отдельного календаря или клиента электронной почты хранить его собственную баду данных - вместо этого будет доступен центральный репозиторий. Это значительно сокращает усилия по администрированию информации. Использование открытого и стандартизированного протокола, как LDAP, гарантирует, что многие различные приложения имеют возможность получить доступ к информации.

Каталог в этом контексте - это тип базы данных, оптимизированный для быстрого и эффективного чтения и поиска:

- Делает возможным множественный доступ для чтения, доступ на запись ограничен небольшим числом модификаций, сделанных администратором. Обычные базы данных оптимизированы для доступа к большому объему возможных данных за короткий промежуток времени.
- Поскольку доступ на запись может быть выполнен в ограниченном виде, сервис каталогов используется по большей части для администрирования неизменяемой, статичной информации. Данные в простой базе данных обычно меняются очень часто (**динамичные** данные). Телефонные номера в каталоге компании не меняются так часто, как, например, фигуры, администрируемые в аккаунтинге (так и не сообразил, как перевести по-нормальному)
- Когда управляют статичными данными, модификации текущих данных очень редки. При работе с динамическими данными, особенно, когда данные подобны банковским аккаунтам или управлению аккаунтами, последовательность данных имеется первостепенную важность. Если величину нужно вычесть из одного места и добавить в другое, оба дествия должны производиться параллельно, в пределах **транзакции**, чтобы сохранит баланс над основой данных. Базы данных также поддерживают транзакции. Каталоги нет. Краткосрочные несогласованности данных в каталогах вполне приемлемы.

Дизайн такого сервиса каталогов, как LDAP, не подразумевает поддержки сложных модификаций и механизмов запросов. Все приложения, обращающиеся к этому сервису, должны получать доступ легко и быстро.

LDAP против NIS

Администраторы Unix-систем традиционно используют сервис NIS для разрешения имен и предоставления данных в сети. Данные конфигурации содержатся в файлах в `/etc` и каталогах `group`, `hosts`, `mail`, `netgroup`, `networks`, `passwd`, `printcap`, `protocols`, `rpc` и `services`, которые предоставляются всем пользователям в сети. Эти файлы могут поддерживаться без особых усилий, так как являются простыми текстовыми файлами. Обработка большого количества данных, однако, становится все более и более сложной из-за отсутствия структурирования. NIS разработан только для платформы Unix. И потому не очень подходит для централизованного управления в гетерогенных сетях.

В отличие от NIS, сервис LDAP не ограничен только сетями UNIX. Серверы Windows (начиная с 2000) поддерживают LDAP, как сервис каталогов. Прикладные задачи, описанные выше, дополнительно поддерживаются в не-Unix сетях.

Принцип LDAP может быть применен к структуре любых данных, которая может централизованно администрироваться. Примеры нескоьких приложений:

- Применение в качестве замены сервиса NIS
- Перенаправление почты (postfix, sendmail)
- Адресные книги для таких почтовых клиентов, как Mozilla, Evolution и Outlook
- Администрирование зоны, написанной для сервера имен Bind9
- Аутентификация пользователей совместно с Samba в гетерогенных сетях

Этот список может быть расширен, поскольку LDAP более гибкая вещь, чем NIS. Ясная иерархическая структура данных облегчает администрирование большого количества данных, так как они могут быть найдены значительно проще.

Структура дерева каталога LDAP

Для лучшего понимания, как работает LDAP и как хранятся данные, очень важно понимать, каким образом данные организованы на сервере и как эта структура позволяет LDAP обеспечить быстрый доступ к необходимым данным. Для успешной установки LDAP необходимо ознакомиться с некоторой терминологией LDAP. Эта секция позволяет понять основной вывод дерева LDAP и описывает терминологию в контексте LDAP. Пропустите эту секцию, если вы уже имеете некоторое представление о работе LDAP и просто хотите научиться установке LDAP-окружения в SUSE Linux Enterprise. Прочитайте «Конфигурирования сервера LDAP с помощью YaST» и «Конфигурация сервера с помощью slapd.conf», соответственно.

Каталог LDAP имеет структуру дерева. Все записи (называемые объектами) каталога имеют определенную позицию в этой иерархии. Эта иерархия называется Информационным Деревом Справочника (**directory information tree** — DIT). Полный путь к необходимой записи, который однозначно идентифицирует ее, называется **distinguished name** (характерное имя) или DN. Единый узел вдоль пути к этой записи называется **relative distinguished name** или RDN. В целом, объекты могут относиться к одному из двух типов:

container

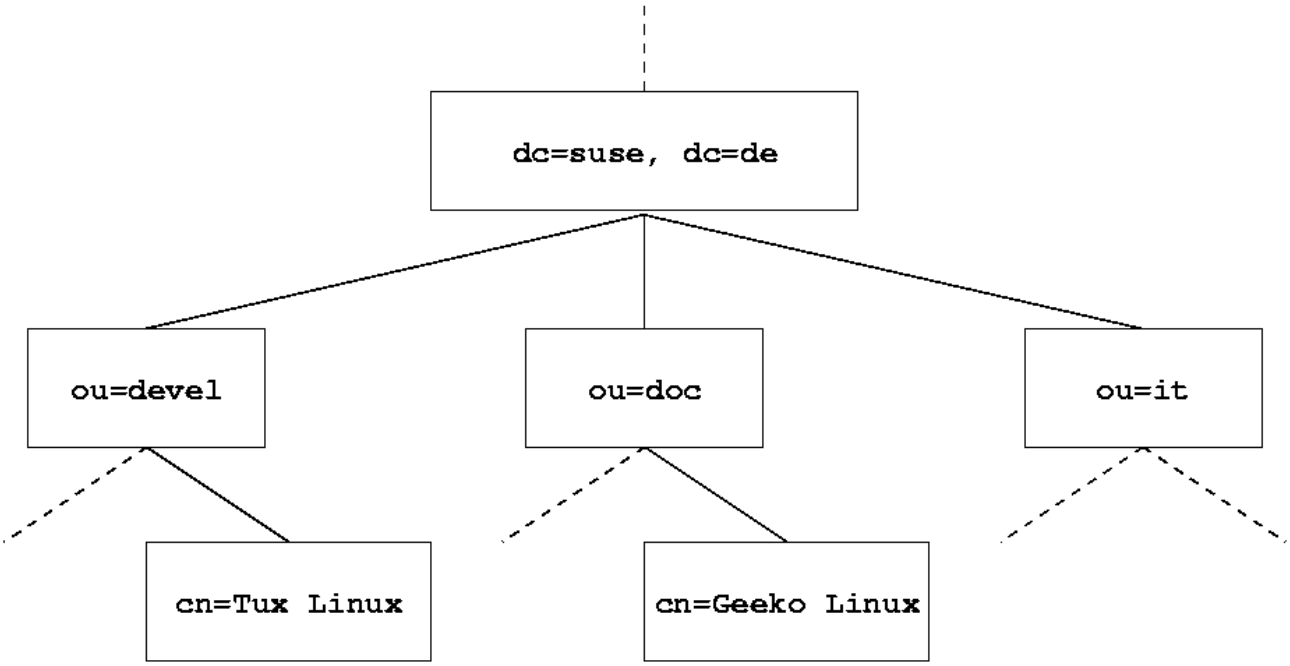
Это объекты, которые в самих себе могут содержать другие объекты. Это такие классы объектов, как `root` (корневой элемент дерева каталогов, который в действительности не существует), `c` (country), `ou` (organizational unit), и `dc` (domain component). Эта модель похожа на директории в файловой системе.

leaf

Эти объекты расположены на конце ветви и не имеют никаких подчиненных объектов. Примеры: `person`, `InetOrgPerson` или `groupofNames`.

В верху иерархии каталогов существует корневой элемент `root`. Он может содержать `c` (country), `dc` (domain component) или `o` (organization), в качестве подчиненных элементов. Отношения элементов в пределах дерева LDAP хорошо видны на следующем рисунке:

Рисунок 1. *Структура дерева LDAP*



Полная диаграмма является вымышленным информационным деревом каталога. Изображены записи на третьем уровне. Каждая запись соответствует одному квадрату на картинке. Полное **distinguished name** для вымышленного пользователя Geeko Linux будет, в данном случае,

`cn=Geeko Linux,ou=doc,dc=example,dc=com`. Оно создается путем добавления RDN `cn=Geeko Linux` к DN предыдущей записи `ou=doc,dc=example,dc=com`.

Типы объектов, которые хранятся в DIT, в общем случае определяются, следуя схеме (**scheme**). Тип объекта определяется классом объекта (**object class**). Класс объекта определяет, какие свойства связанного объекта должны или могут быть назначены. Схема, в свою очередь, должна содержать определения всех классов объекта и свойства, использующиеся в нужном прикладном сценарии. Существует несколько общих схем (смотрите RFC 2252 и 2256). Однако, существует возможность создание собственной схемы или множества схем, которые дополняют друг друга, если это требуется окружением, в котором работает сервер LDAP.

В Таблице 1 показан небольшой обзор классов объекта из `core.schema` и `inetorgperson.schema`, использующиеся в примере, включая свойства и действительные значения свойств.

Таблица 1. *Общие использующиеся классы объектов и атрибуты*

Класс Объекта	Значение	Примерная запись	Необходимые свойства
dcObject	domainComponent (именованые компоненты домена)	example	dc
organizationalUnit	organizationalUnit (организационная единица)	doc	ou
inetOrgPerson	inetOrgPerson (данные, связанные с человеком)	Geeko Linux	sn и cn

В Листинге 1 показана часть из директивы схемы с объяснениями (строки пронумерованы для большей понятности).

Листинг 1. *Выдержка из schema.core*

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

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

...
```

Тип свойства `organizationalUnitName` и переданный класс объекта `organizationalUnit` в данном случае служат примером. В строке отображается имя свойства, его уникальный OID (**oblect identifier**) (цифровой) и абревиатура свойства.

В строке 2 дается краткое описание свойства с помощью DESC. Соответствующий RFC, на котором основано данное определение, также упоминается в этой строке. SUP в строке 3 указывает на тип `superordinate` (так и не смог перевести)) свойства, к которому это свойства принадлежит.

Описание класса объекта `organizationalUnit` начинается со строки 4, как и в описании свойства, с OID и имени класса объекта. Строка 5 показывает краткое описание класса объекта. Строка 6, с ее записью SUP

top, указывает, что этот класс объекта не зависит от другого класса объекта. Строка 7, начинающаяся с MUST, показывает все типы свойств, которые должны использоваться в связке с объектом типа organizationalUnit. В строке 8, начинающейся с MAY, показаны все типы свойств, которые позволено связывать с этим классом объекта.

Очень хорошее введение в описание использования схем можно найти в документации к openLDAP. После установки, она доступна в /usr/share/doc/packages/openldap2/admin-guide/index.html.

Конфигурация сервера с помощью slapd.conf

Ваша установленная система содержит полный файл конфигурации для вашего сервера LDAP в `/etc/openldap/slapd.conf`. Здесь будут кратко описаны отдельные записи и объяснены необходимые настройки.

Глобальные директивы в slapd.conf

***Листинг 2**. slapd.conf: включая директивы для схем*

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/rfc2307bis.schema
include         /etc/openldap/schema/yast.schema
```

Эта первая директива в slapd.conf, показанная в Листинге 2, указывает, в каком каталоге LDAP организованы схемы. Запись `core.schema` обязательна. Дополнительные необходимые схемы добавляются в эту директиву. Больше информации можно найти во включенной документации LDAP.

***Листинг 3.** slapd.conf: pid-файл и args-файл*

```
pidfile /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

Эти два файла содержат PID (ID процесса) и некоторые из аргументов, с помощью которых процесс `slapd` стартует. Нет необходимости вносить в них какие-либо изменения.

***Листинг 4.** slapd.conf: Контроль доступа*

```
# Sample Access Control
#       Allow read access of root DSE
# Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
# access to dn="" by * read
  access to * by self write
              by users read
              by anonymous auth
#
# if no access controls are present, the default is:
#       Allow read by all
#
# rootdn can always write!
```

В Листинге 4 показана часть файла `slapd.conf`, в которой регулируются права доступа для каталога LDAP на сервере. Настройки, сделанные в глобальной секции `slapd.conf` действительны, пока никакие права доступа не объявлены в какой-либо секции базы данных, так как они перезаписывают глобальные объявления. Как указано здесь, все пользователи имеют право на чтение в этой директории, но только администратор (`rootdn`) имеет право записи. Регулирования контроля доступа в LDAP — чрезвычайно сложный процесс. Следующие советы могут вам помочь:

- Каждое правило доступа имеет следующую структуру:

  `access to <what> by <who> <access>`

- *what* является заменителем объекта или свойства, к которому предоставляется доступ. Отдельные ветви каталогов могут быть защищены с помощью отдельных правил. Также есть возможность

обработать разделы дерева каталогов в одном правиле с помощью регулярных выражений. `slapd` определяет правила в том порядке, в котором они указаны в конфигурационном файле. Большинство основных правил должны быть перечислены после большинство специфичных — первое правило `slapd` воспринимает, как единственно определенное, и все следующие записи игнорируются.

- *who* определяет, кому предоставляется доступ к областям, определенным с помощью *what*. Регулярные выражения здесь также могут использоваться. `slapd` также прерывает обработку после первого совпадения, потому более конкретные правила нужно указывать перед более общими. Возможны записи, показанные в Таблице 2:

Таблица 2. *Группы пользователей и их права доступа*

Тэг	**Пределы**
*	Все пользователи без исключения
anonymous	Не аутентифицированные (анонимные) пользователи
users	Аутентифицированные пользователи
self	Пользователи, подключенные к целевому объекту
dn.regex=<regex>	Все пользователи, соответствующие регулярному выражению

- *access* указывает на тип доступа. Используйте опции, перечисленные в Таблице 3:

Таблица 3. *Типы доступа:*

Тэг	**Доступ**
none	Нет доступа
auth	Для подключению к серверу
compare	К объектам для сравнительного доступа
search	Для использования фильтра поиска
read	Доступ на чтение
write	Доступ на запись

`slapd` сравнивает права доступа, запрошенные клиентом, с теми, которые указаны в `slapd.conf`. Клиенту предоставляется доступ в том случае, если правила позволяют получить аналогичные или более высокие права, чем были запрошены. Если клиент запрашивает права выше, чем объявлены в правилах, то для него в доступе будет отказано.

В Листинге 5 показан пример простого контроля доступа, который вполне может быть изменен и дополнен с помощью регулярных выражений.

Листинг 5. *slapd.conf: Пример контроля доступа*

```
access to  dn.regex="ou=([^,]+),dc=example,dc=com"
by dn.regex="cn=Administrator,ou=$1,dc=example,dc=com" write
by user read
by * none
```

Это правило гласит, что только соответствующий администратор имеет право изменять запись ou. Все другие аутентифицированные пользователи имеют доступ на чтение, а у остальной части мира нет доступа вообще.

Hint: Установка правил доступа

Если нет правила `access to` или никакие директивы не совпадают — доступ запрещен. Предоставляются только явно объявленные права доступа. Если вообще никакие правила не объявлены , по умолчанию администратору предоставляются права на запись, а всем остальным — доступ на чтение.

Более подробную информацию и примеры конфигурации прав доступа LDAP можно найти в он-лайн документации из установленного пакета openldap2.

Кроме возможности управлять правами доступа с помощью основного файла конфигурации (slapd.conf), существует информация о контроде доступа (access control information (ACI)). ACI позволяет хранить информацию о доступе для конкретного объекта внутри дерева LDAP. Этот вид контроля доступа редко применяется, и разработчики считают его экспериментальным. За информацией можно обратиться по адресу http://www.openldap.org/faq/data/cache/758.html

Директивы в slapd.conf, специфичные для базы данных

Листинг 6. *slapd.conf: Директивы, специфичные для базы данных*

```
database bdb❶
suffix "dc=example,dc=com"❷
checkpoint      1024    5❸
cachesize       10000❹
rootdn "cn=Administrator,dc=example,dc=com"❺
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret❻
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap❼
# Indices to maintain
index   objectClass     eq❽
overlay ppolicy❾
ppolicy_default "cn=Default Password Policy,dc=example,dc=com"
ppolicy_hash_cleartext
ppolicy_use_lockout
```

❶ - Тип базы данных, Berkeley database в данном случае, установлена в первой строке этой секции.
❷ - `suffix` определяет, за какую чатсь дерева LDAP ответственен этот сервер
❸ - `checkpoint` определяет размер данных (в килобайтах), которые хранятся в логе транзакции до тех пор, пока не будут записаны в дествительную базу данных, и время (в минутах) между двумя записями.
❹ - `cachesize` устанавливает количество объектов, находящихся в кэше базы данных
❺ - `rootdn` определяет, кто владеет правами администратора на этом сервере. Пользователю, объявленному здесь, нет необходимости иметь запись в LDAP или существовать в качестве обычного пользователя.
❻ - `rootpw` устанавливает пароль администратора. Вместо того, чтобы использовать указанный здесь secret, необходимо ввести хэш пароля администратора, созданный с помощью `slappasswd`.
❼ - директива `directory` указывает каталог в файловой системе на сервере, где находится база данных.
❽ - Последняя директива, `index objectClass eq`, приводит к созданию индекса всех классов объектов. Свойства, которые пользователи ищут наиболее часто, могут быть добавлены соответственно сюда для проверки.
❾ - `overlay ppolicy` добавляет уровень механизмов контроля паролей. `ppolicy_default` указывает DN объекта `pwdPolicy`, чтобы использовать в том случае, если не установлена никакая политика лдля записи данного пользователя. Если нет никакой конкретной политики для записи и значение по умолчанию не установлено, никакие политики не применяются. `ppolicy_hash_cleartext` указывает, что пароли в

открытом виде, присутствующие при добавлении или изменении запросов, хэшируются перед созранением в базе данных. Когда используется эта опция, рекомендуется запретить сравнение, поиск и доступ на чтение к свойству `userPassword` для всех пользователей каталогов, потому что `ppolicy_hash_cleartext` нарушает информационную модель X.500/LDAP. `ppolicy_use_lockout` отправляет указанный код ошибки в том случае, когда пользователь пытается подключиться к заблокированному аккаунту. Если ваш сайт критичен в плане безопасности, отключите эту опцию, так как код ошибки может предоставить полезную информацию атакующим.

Правила `Access`, описанные здесь для базы данных, используются вместо глобальных правил `Access`.

Запуск и остановка серверов.

Как только сервер LDAP полностью настроен и все необходимые записи сделаны согласно образцу, описанному в разделе Обработка Данных В Каталоге LDAP, запустите сервер как пользователь root, введя команду `rcldap start`. Для остановки сервера вручную, введите `rcldap stop`. Запросить статус запущенного сервера можно с помощью команды `rcldap status`.

Редактор уровня запуска YaST можно использовать для запуска и остановки сервера при загрузке и остановке системы соответственно. Также есть возможность создать символические ссылки для скриптов старта и остановки с командой `insserv` из командной строки.

Управление данными в каталоге LDAP

OpenLDAP предоставляет серию инструментов для администрирования данными в каталоге LDAP. Четыре наиболее важных инструмента для добавления, удаления, поиска и изменения набора данных кратко описаны ниже.

Вставка данных в каталог LDAP

В том случае, если конфигурация вашего сервера LDAP в `/etc/openldap/slapd.conf` сделана корректно и готова к использованию (что показывают соответствующие записи для `suffix`, `directory`, `rootdn`, `rootpw` и `index`), следует приступить к вводу записей. Для этой задачи openLDAP предоставляет команду `ldapadd`. Если есть возможность, добавляйте объекты в базу данных в связке по вполне практичным причинам. Для этого LDAP имеет возможность для обработки формата LDIF (LDAP data interchange format). Файл LDIF — это простой тестовый файл, содержащий произвольное количество атрибутов и пар значений. Обращайтесь в файлам схем, объявленных в `slapd.conf` за доступными классами объектов и атрибутами. Файл LDIF для создания примерной структуры, например, на Рисунке 1 мог бы выглядеть, как в Листинге 7.

Листинг 7. *Пример файла LDIF*

```
# 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
```

> **Важно: Кодировка файлов LDIF**
> LDAP работает с UTF-8 (Уникод). Умляуты должны кодироваться правильно. Используйте редактор, поддерживающий уникод, например, Kate или последние версии Emacs. В другом случае, избегайте умляутов или других специальных символов, или используйте перекодировку в UFT-8.

Сохраните файл с суффиксом `.ldif`, затем отправьте его на сервер с помощью следующей команды:

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

`-x` отключает аутентификацию с помощью SASL в данном случае. `-D` объявляет пользователя, которые вызывает действие. Дествительный DN администратора, введенный здесь, должны быть таким же, как в файле `slapd.conf`. В данном примере, это `cn=Administrator,dc=example,dc=com`. `-W` предотвращает ввод пароля в командной строке (открытым текстом) и активирует отдельное приглашение на ввод пароля. Этот пароль был заранее определен в `slapd.conf` с помощью `rootpw`. `-f` передает имя файла. Детали работы `ldapadd` можно посмотреть в Листинге 8.

Листинг 8. *ldapadd с помощью 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"
```

Данные конкретного пользователя могут быть подготовлены в отдельном файле LDIF. В Листинге 9 пользователь Tux добавляется в новый каталог LDAP.

Листинг 9. *Данные LDIF для пользователя 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
```

Файл LDIF может содержать произвольное количество объектов. Возможно отправлять полные ветви каталога на сервер сразу или только его часть, как показано в примере конкретных объектов. Если есть необходимость изменять отдельные данные достаточно часто, рекомендуется использовать отдельную часть одиночных объектов.

Модификация данных в каталоге LDAP.

Для модификации набора данных предоставляется инструмент `ldapmodify`. Наиболее просто йспособ сделать это — изменить соответствующий файл LDIF, а затем отправить его на сервер. Для изменения телефонного номера коллеги Tux с +49 1234 567-8 на +49 1234 567-10 отредактируйте файл LDIF, как показано в Листинге 10.

Листинг 10. *Изменение файла LDIF tux.ldif*

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

Испортируйте измененный файл в каталог LDAP с помощью следующей команды:

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

По другому, отправьте атрибуты для изменения напрямую в `ldapmodify`. Этот процесс описан ниже:

1. Запустите `ldapmodify` и введите пароль:

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

2. Введите изменения, только внимательно следя за синтаксисом в порядкепредставленном ниже:

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

Более подробную информацию об `ldapmodify` и его синтаксисе смотрите на man-странице `ldapmodify`.

Поиск и чтение данных из каталога LDAP.

OpenLDAP представляет, с помощью `ldapsearch`, интсрумент командной строки для поиска данных внутри каталога LDAP и чтения данных из него. Простой запрос может иметь следующий синтаксис:

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

`-b` определяет базу поиска — раздел дерева, внутри которого будет поизводиться поиск. В данном случае, это `dc=example,dc=com`. Чтобы выполнить более подробный поиск в конкретном подразделе каталога LDAP (например, только в пределах раздела `devel`), отправьте этот раздел в `ldapsearch` с помощью `-b`. `-x` запрашивает активацию простой авторизации. `(objectClass=*)` объявляет, что все объекты, содержащиеся в каталоге, необходимо прочитать. Эта опция команды может использоваться после создания нового дерева каталогов для проверки, что все записи были сделаны правильно и сервер отвечает правильно. Больше информации об использовании `ldapsearch` можно найти на man-странице этой команды (`ldapsearch(1)`).

Удаление данных из каталога LDAP.

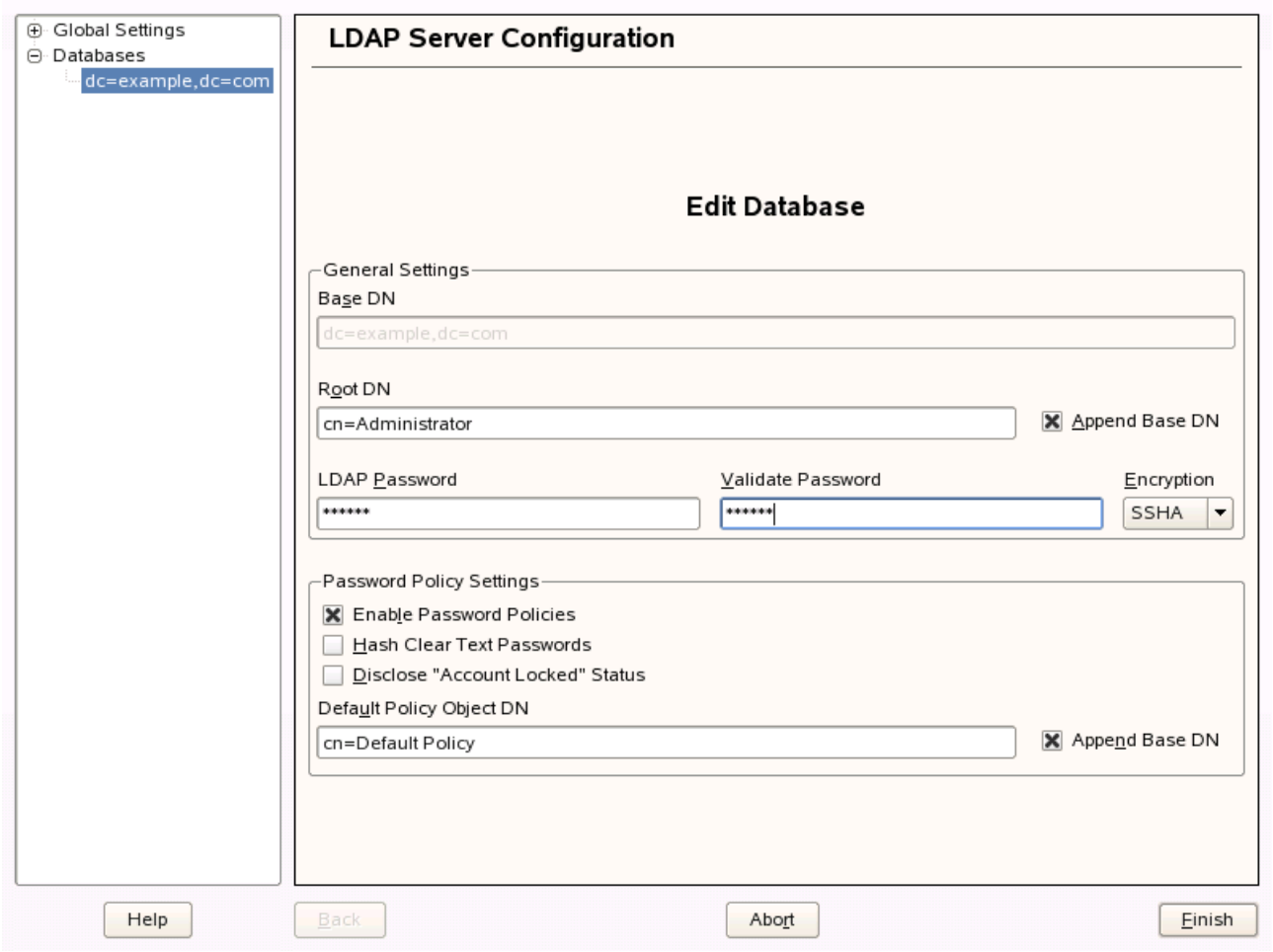
Удалить нежелательные записи можно с помощью `ldapdelete`. Синтаксис этой команды похож на синтаксис остальных команд. Чтобы удалить, например, всю информацию о пользователе Tux Linux, запустите следующую команду:

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

Конфигурирование сервера LDAP с помощью YaST.

Используйте YaST для установки сервера LDAP. Типичные случаи использования серверов LDAP включают в себя управление аккаунтами пользователей и конфигурирование почты, серверов DNS и DHCP.

Рисунок 2. *Конфигурация YaST LDAP Server*



Для установки сервера LDAP для управления аккаунтами пользователей, сделайте следующее:

1. Войдите как пользователь root.
2. Запустите YaST и выберите *Network Services* > *LDAP Server*.
3. Настройте LDAP для старта во время загрузки системы.
4. Если необходимо, чтобы сервер LDAP предоставлял свои сервисы через SLP, выберите *Register at an SLP Daemon*.
5. Выберите *Configure* для настройки *General Settings* и *Databases*.

Чтобы настроить *General Settings* вашего сервера LDAP, сделайте следующее:

1. Примите или измените файлы схем, включенные в конфигурацию сервера, выбрав *Schema Files* в левой части диалога. Выбор файлов схем по умолчанию предполагает наличие возможности управления аккаунтами пользователей с помощью YaST.
2. С помощью Log Level Settings настройте степень активности логгирования (подробность) сервера LDAP. Из встроенного списка выберите или исключите опции логгирования согласно вашим потребностям. Чем больше опций включено, тем больший размер будет у логов.
3. Выберите тип соединения, который будет позволен для сервера LDAP. Выберите из следующего:

   bind_v2
   эта опция позволяет запросы на соединение (bind requests) от клиентов, использующих предыдущую версию протокола (LDAPv2).

bind_anon_cred

Обычно сервер LDAP запрещает любые попытки аутентификации с пустыми регистрационными данными (DN или пароль). Включение этой опции, однако, делает возможным соединение с паролем и без DN для установки анонимного соединения.

bind_anon_dn

Включение этой опции делает возможным соединение без аутентификации (анонимно), используя DN, но без пароля.

update_anon

Включение этой опции позволяет делать неаутентифицированные (анонимные) модификации. Доступ ограничивается согласно ACL и другими правилами (смотрите раздел *Глобальные Директивы в slapd.conf*).

4. Для настройки безопасного соединения между клиентом и сервером продолжите с *TLS Settings*:

a) Установите *TLS Active* в *Yes* для включения шифрования SSL и TLS при соединении клиента/сервера.
b) Кликните *Select Certificate* и определите, как получать правильный сертификат. Выберите *Import Certificate* (импортировать сертификат с внешнего источника) или *Use Common Server Certificate* (использовать сертификат, созданный в процессе установки).

- Если вы выбрали импортирование сертификата, YaST попросит вас указать полный путь до места, где он расположен.
- Если вы выбрали использование общего сертификата сервера и он не был создан в процессе установки, он будет создан.

Чтобы настроить базы данных, управляемые с помощью YaST, проделайте следующее:

1. Выберите пункт *Databases* в левой части диалога.
2. Кликните *Add Database* для добавления новой базы данных.
3. Введите необходимые данные:

Base DN

Введите основной DN вашего сервера LDAP.

Root DN

Введите DN администратора. Если вы отметите Append Base DN, будет необходимо предоставить только cn администратора и система оставит вас в покое.

LDAP Password

Введите пароль администратора базы данных.

Encryption

Определяет алгоритм шифрования, который используется для безопасности пароля Root DN. Выберите *crypt*, *smd5*, *ssha* или *sha*. Диалог также включает в себя выбор использования пароля в открытом виде, но данную опцию не рекомендуется использовать по причинам безопасности. Чтобы подтвердить ваши настройки и вернуться к предыдущему диалогу, выберите *Ok*.

4. Включите возможность политик паролей для обеспечения дополнительной безопасности для сервера LDAP:

- Выберите *Password Policy Settings* для выбора конкретной политики пароля.
- Активируйте *Hash Clear Text Passwords*, чтобы пароли открытым текстом хэшировались до того, как они будут записаны в базу данных, каждый раз, когда они изменяются или добавляются.
- *Disclose Account Locked Status* предоставляет важное сообщение об ошибке для связывания запросов с заблокированными аккаунтами.

Внимание: Заблокированные аккаунты в окружении, чувствительном для безопасности. Не используйте опцию *Disclose Account Locked Status*, если ваше окружение чувствительно к проблемам безопасности, потому что сообщение о заблокированном аккаунте предоставляет важную с точки зрения безопасности информацию, которая может эксплуатироваться потенциальными нападающими.

- Введите DN объекта политики по умолчанию. Для использования DN, отличного от предложенного YaST, введите необходимое вам. Иначе принимаются значения, установленные по умолчанию.

5. Завершите конфигурирование базы данных, кликнув *Finish*.

Если вы не выбирали политики паролей, ваш сервер в данный момент готов к запуску. Если вы включили политики паролей, продолжите конфигурирование политик паролей более подробно. Если вы выбрали объект политики паролей, которого не существует, YaST создаст его:

1. Введите пароль сервера LDAP.
2. Сформируйте политику изменения паролей:

   - Определите количество паролей, хранящихся в истории паролей. Сохраненные пароли могут не использоваться пользователем несколько раз.
   - Определите, будут ли пользователи иметь возможность менять свои пароли и нужно ли им менять им свои пароли после сброса администратором. Дополнительно, при изменении пароля можно затребовать ввод старого пароля.
   - Определите, какого размера должен быть пароль. Устанавливается минимальная длина пароля, которая должна иметь место быть, чтобы пароль считался дествительным. Если вы выберите *Accept Uncheckable Passwords*, пользователям будет позволено использовать зашифрованные пароли, хотя в таком случае качественная их проверка не может быть выполнена. Если вы выберите для *Only Accept Checked Passwords* те пароли, которые проходят тесты на качество, будут считаться действительными.

3. Опеределите политику устаревания паролей:

   - Определите минимальный срок дествия паролей (время, которое должно пройти между двумя сменами паролей) и максимальный возраст пароля.
   - Определите время между предупреждением об истечении времени дествия пароля и его фактическим истечением.
   - Установите количество предупреждений об окончании срока действия пароля перед тем, как срок истечет полностью.

4. Настройте политики блокировок:

   - Разрешите блокировку паролей.
   - Определите количество неудачных попыток, после которых пароль блокируется.
   - Определите на какое время будет блокироваться пароль.
   - Определите, как долго неудачные попытки ввода паролей будут храниться в кэше до того, как будут очищены.

5. Сохраните свои настройки политик паролей, нажав *Accept*.

Чтобы отредактироват ранее созданную базу данных, выберите ее DN в дереве слева. В правой части окна YaST покажет диалог, подобный тому, который использовался при создании новой базы данных — с главным отличием, что запись основного DN — grayed и не может быть изменена.

После того, как вы покинете настройку конфигурации сервера LDAP, выбрав *Finish*, у вас будет готова основная рабочая конфигурация для вашего сервера LDAP. Для тонкой настройки отредактируйте файл `/etc/openldap/slapd.conf`, затем перезапустите сервер.

Конфигурирование клиента LDAP с помощью YaST

YaST включает в себя модуль для установки управления пользователями через LDAP. Если вы не выбрали эту возможность во время установки, запустите модуль, выбрав *Network Services > LDAP Client*. YaST автоматически позволит любые изменения, касающиеся PAM и NSS, которые требуются LDAP, и установит необходимые файлы.

Стандартная процедура.

Знание процессов, действующих на заднем плане клиентской машины, поможет вам понять, как работает клиента LDAP в YaST. Если LDAP активируется для сетевой аутентификации или вызывается модуль YaST, в таком случае пакеты pam_ldap и nss_ldap установлены и два соответствующих файла конфигурации подготовлены. pam_ldap — это модуль PAM, ответственный за переговоры процессов входа в систему и катталогом LDAP, как источником данных аутентификации. Выделенный модуль `pam_ldap.so` установлен и конфигурация PAM подготовлена (смотрите Листинг 11).

Листинг 11. *pam_unix2.conf, подготовленный для LDAP.*

```
auth:       use_ldap
account:    use_ldap
password:   use_ldap
session:    none
```

Когда дополнительные сервисы настраиваются для использования с LDAP, необходимо включить модуль PAM LDAP в конфигурационном файле PAM, ответственном за сервис в `/etc/pam.d`. Файлы конфигурации, уже подготовленные к конкретным сервисам, могут быть найдены в `/usr/share/doc/packages/pam_ldap/pam.d/`. Скопируйте соответствующие файлы в `/etc/pam.d`.

Разрешение имен `glibc` через механизм `nsswitch` адаптировано к использованию LDAP с nss_ldap. Новый, адаптированный файл `nsswitch.conf` создается в `/etc` в момент установки этого пакета. Следующие строки должны присутствовать в `nsswitch.conf` для администрирования пользователей и аутентификации в LDAP. Смотрите Листинг 12.

Листинг 12. *Адаптация в nsswitch.conf*

```
passwd: compat
group: compat

passwd_compat: ldap
group_compat: ldap
```

Эти строки предписывают сначала обращаться к библиотеке распознавания `glibc` для оценки соответствующих файлов в `/etc`, и дополнительно обращаются к серверу LDAP, как источнику аутентификации и данных пользователей. Протестируйте этот механизм, например, путем чтения содержимого базы данных пользователей с помощью команды `getent passwd`. Ответ должен содержать обзор локальных пользователей вашей системы точно так же, как и пользователей, хранящихся на сервере LDAP.

Чтобы запретить обычным пользователям, управляемых через LDAP, возможность входить на сервер, используя `ssh` или просто логин, необходимо в файлы `/etc/passwd` и `/etc/group` дополнительную строки. Это строка `+::::::/sbin/nologin` в файле `/etc/passwd` и `+:::` в `/etc/group`.

Конфигурирование клиента LDAP.

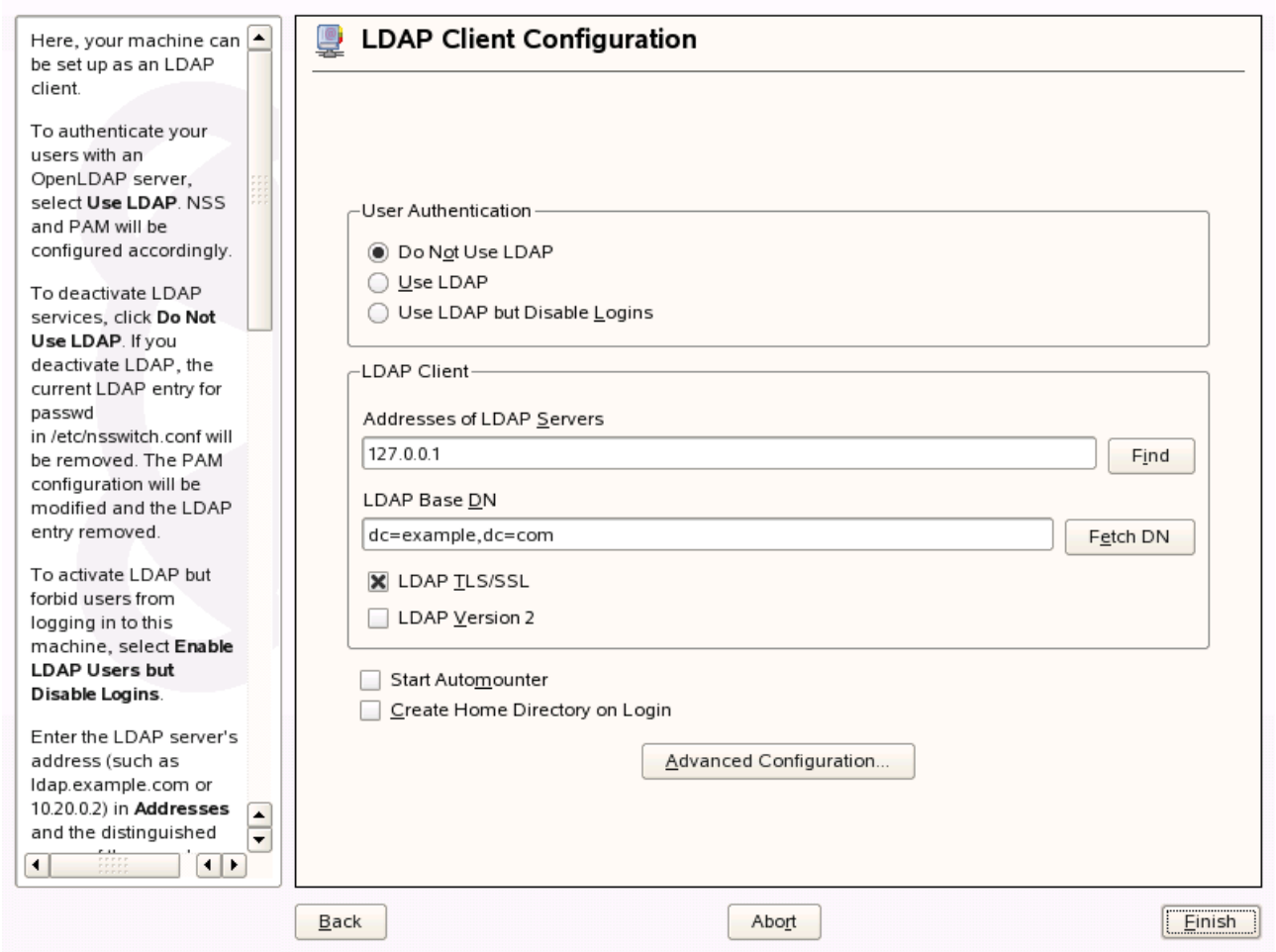
После того, как начальные настройки `nss_ldap`, `pam_ldap`, `/etc/passwd` и `/etc/group` были сделаны через YaST, вы можете подключить своего клиента к серверу и позволить YaST управлять пользователями через LDAP. Эта основная настройка описаны в *Основной Конфигурации*.

Используя YaST клиент LDAP в дальнейшем настраивает модули управления группами и пользователями YaST. Оно включает в себя манипулирование значениями по умолчанию для новых пользователей и групп и количество и природа атрибутов для пользователей и групп. Управление пользователями LDAP позволяет вам назначать гораздо больше и различные атрибуты для пользователей и групп, чем традиционные способы управления. Это подробно описывается в Конфигурирование Модулей YaST По Управлению Пользователями и Группами.

Основная конфигурация

Диалог основной конфигурации клиента LDAP (Рисунок 3) открывается в процессе установки в том случае, если вы выбрали управление пользователями через LDAP или когда вы выберите *Network Services* > *LDAP Client* в YaST Control Center на установленной системе.

Рисунок 3. *YaST: Конфигурация клиента LDAP*

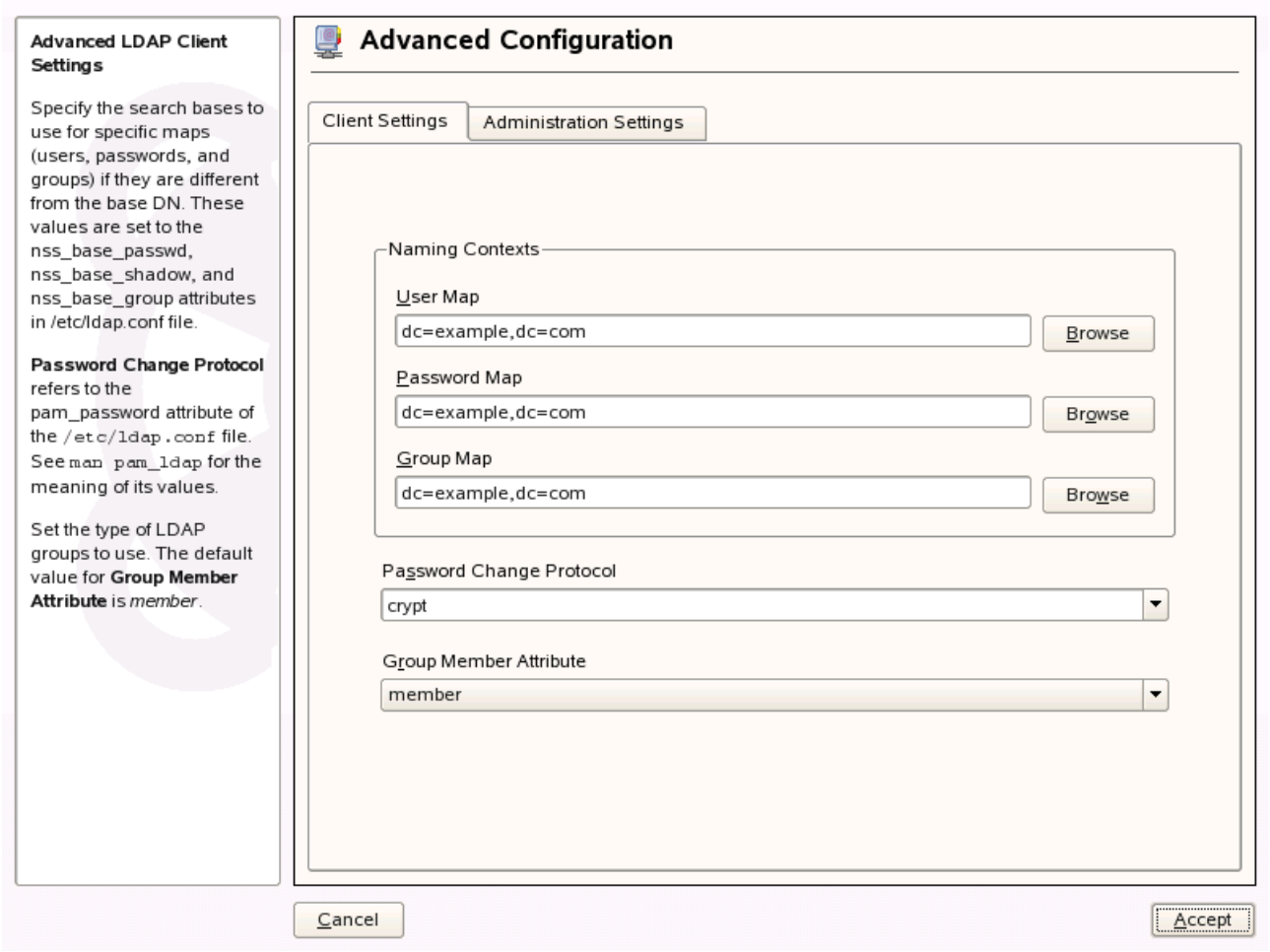


Чтобы включить возможность управления пользователями машины через openLDAP, проделайте следующее:

1. Кликните *Use LDAP* для использования LDAP. Выберите *Use LDAP but Disable Logins* вместо этого, если вы хотите использовать LDAP для аутентификации, но не хотите, чтобы другие пользователи логинились на этом клиенте.
2. Введите IP-адрес сервера LDAP, который используете.
3. Введите *LDAP base DN* для выбора базы поиска на сервере LDAP. *Для* получения base DN автоматически, кликните *Fecth DN*. После этого YaST проверит каждую базу данных LDAP на адресе сервера, указанном выше. Выберите необходимое base DN из результатов поиска, предоставленных YaST.
4. Если сервер требует, чтобы соединение защищалось с помощью TLS или SSL, выберите *LDAP TLS/SSL*.
5. Если сервер LDAP до сих пор использует LDAPv2, укажите явно использовать эту версию протокола, выбрав *LDAP Version 2*.
6. Выберите *Start Automounter* для монтирования удаленных каталогов на вашем клиенте, таких как, удаленно управляемый `/home`.

7. Выберите *Create Home Directory on Login*, чтобы домашний каталог пользователя автоматически создавался при первом входе в систему.
8. Кликните *Finish* для принятия сделанных настроек.

Рисунок 4. *YaST: Расширенная конфигурация*



Для изменения данных на сервер, в качестве администратора, кликните Advanced Configuration. Следующий диалог разделен на две вкладки. Смотрите Рисунок 4.

В табе *Client Settings* измените следующие параметры по необходимости:

a) Если база поиска для пользователей, паролей и групп отличается от глобальной базы поиска, указанной в *LDAP base DN*, введите эти другие имена в *User Map*, *Password Map* и *Group Map*.
b) Укажите протокол изменения пароля. Стандартный метод, который используется каждый раз при изменении пароля — crypt, что означает, что используются хэши паролей, созданные `crypt`. За более подробной информацией об этой и других опциях обращайтесь на man-страницу `pam_ldap`.
c) Укажите группу LDAP, которая используется совместно *Group Member Attribute*. Здесь значение по умолчанию — `member`.

В *Administration Settings* измените следующие параметры:

a) Установите базу для хранения ваших данных управления пользователями через *Configuration Base DN*.
b) Введите соответствующее значение для *Administrator DN*. Это DN должно быть идентично значению `rootdn`, указанном в `/etc/openldap/slapd.conf`, чтобы дать обычным пользователям манипулировать данными, хранящимися на сервере LDAP. Введите полное имя DN (например, `cn=Administrator,dc=example,dc=com`) или активируйте *Append Base DN*, чтобы base DN автоматически добавлялось, когда вы введете `cn=Administrator`.
c) Отметьте *Create Default Configuration Objects* для создания основных объектов конфигурации на

сервере, чтобы разрешить управление пользователями через LDAP.

d) Если ваша клиентская машина должна служить файловым сервером для домашних каталогов в вашей сети, отметьте *Home Directories on This Machine*.

e) Используйте раздел *Password Policy* для выбора, добавления, удаления или изменения параметров настройки использующихся политик паролей. Конфигурация политики паролей с помощью YaST — часть установки сервера LDAP.

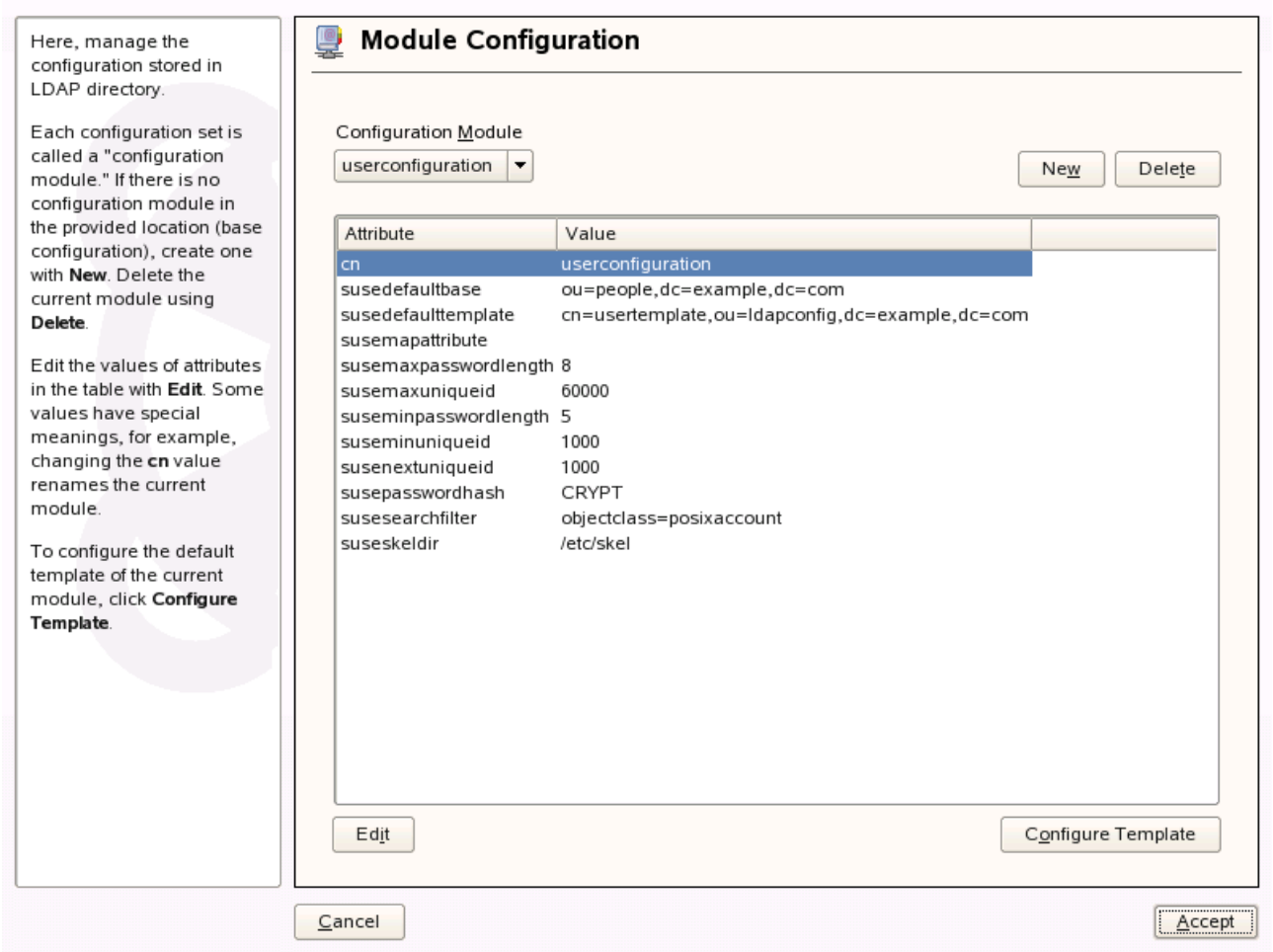
f) Кликните *Accept*, чтобы выйти из *Advanced Configuration*, затем *Finish* для принятия ваших настроек.

Используйте *Configure User Management Settings* для редактирования записей на сервере LDAP. Затем предоставляется доступ к модулям конфигурации, согласно ACL и ACI, находящимся на сервере. Следуйте процедурам, описанным в Конфигурирование Модулей YaST Для Управления Пользователями и Группами.

Конфигурирование Модулей YaST Для Управления Пользователями и Группами.

Используйте клиент YaST LDAP для адаптации модулей YaST для управления пользователями и группами и для расширения их по необходимости. Определите шаблон со значениями по умолчанию для конкретных атрибутов, чтобы сделат регистрацию данных более простой. Преднастройки, сделаные здесь, хранятся, как объекты LDAP в каталоге LDAP. Регистрация данных пользователей до сих пор делается с помощью обычных модулей YaST для управления пользователями и группами. Зарегистрированные данные хранятся в виде объектов LDAP на сервере.

***Рисунок 5**. YaST: Конфигурация модулей*



Диалог конфигурации модулей (Рисунок 5) позволяет создать новые модули, выбирать и модифицировать уже существующие модули конфигурации, а также проектировать и изменять шаблоны для таких модулей.

Для создания нового модуля конфигурации проделайте следующее:

1. Кликните *Next* и выберите тип модуля, который необходимо создать. Для модуля конфигурации пользователя выберите `suseuserconfiguration`, а для конфигурации группы —

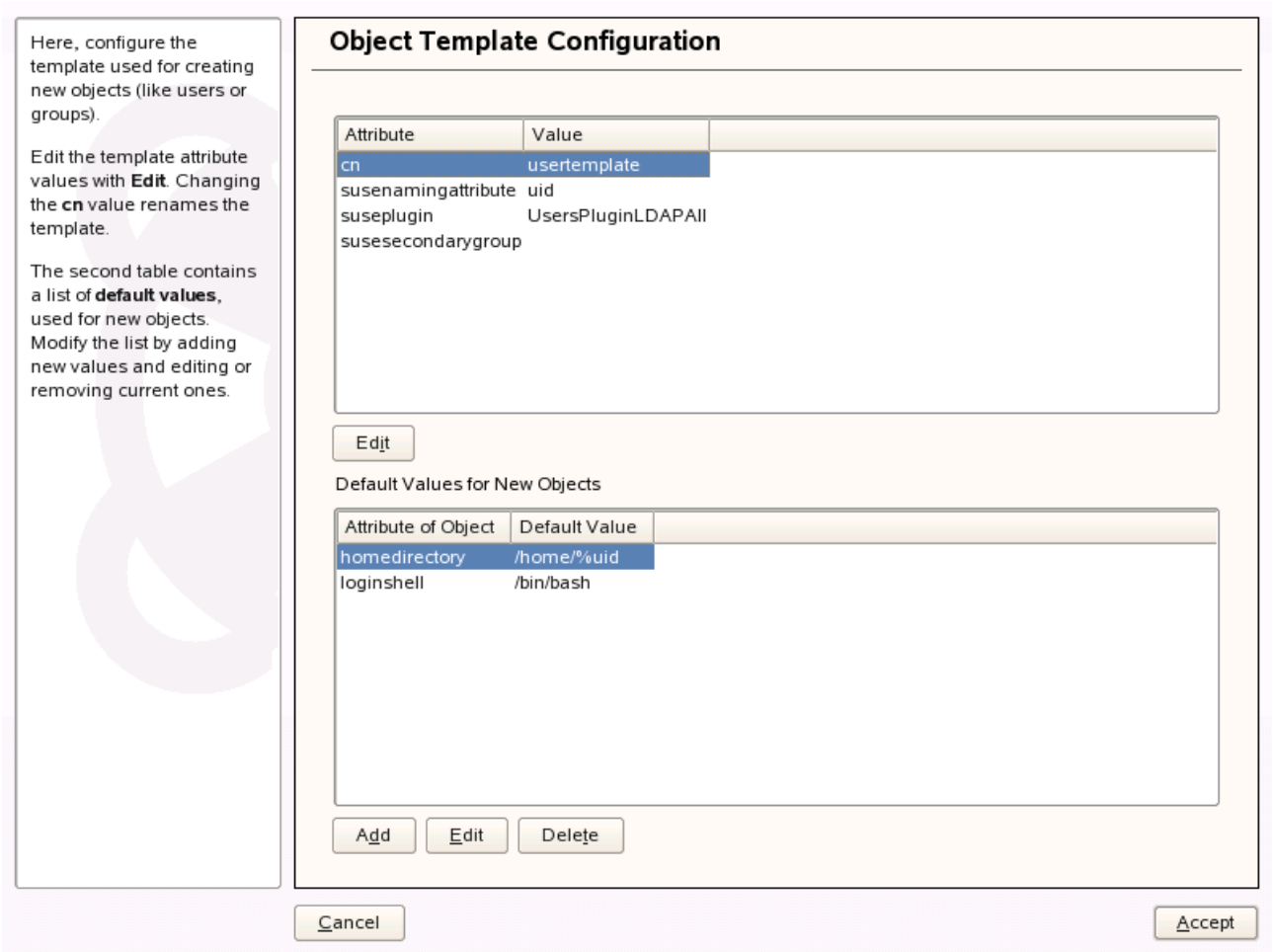
`susegroupconfiguration`.

2. Выберите имя для нового шаблона. Затем содержимое будет выглядеть в виде таблицы с перечислением всех атрибутов, разрешенных в этом модуле, с их установленными значениями. Отдельно от всех наборов атрибутов, в списке также показаны все другие атрибуты, позволенные текущей схемой, но в настоящее время не испоьзующиеся.
3. Приминте предустановленные значения или измените значения по умолчанию для использования конфигурации пользователей и групп, выбрав соответствующий атрибут, нажав *Edit* и введя новое значение. Переименуйте модуль, просто изменив атрибут cn модуля. Кликните *Delete* для удаления выбранного в данный момент модуля.
4. После того, как вы кликните *Accept*, новый модуль будет добавлен в меню выбора.

В модуль YaST для администрирования пользователей и групп встроены шаблоны с разумными стандартными значениями. Чтобы отредактировать шаблон, связанный с модулем конфигурации, сделайте следующее:

1. В диалоге *Module Configuration* кликните *Configure Template*.
2. Определите значения общих атрибутов, относящихся к этого шаблону как вам необходимо, или оставьте некоторые их них пустыми. Пустые атрибуты удаляются на сервере LDAP.
3. Измените, удалите или добавьте новые значения по умолчанию для новых объектов (объекты конфигурирования пользователей и групп в дереве LDAP).

Рисунок 6. *YaST: Конфигурация шаблона объекта*



Объединит ешаблон с его модулем, установив значение атрибута susedefaulttemplate модуля в значение DN соответствующего шаблона.

> **Hint:** Значения по умолчанию для атрибутов можно создать из других атрибутов, используя переменную вместо абсолютного значения. Например, при создании нового пользователя `cn=%sn %givenName` создается автоматически из значений атрибутов для `sn` и `givenName`.

Как только все модули и шаблоны настроены правильно и готовы к запуску, новые группы и пользователи могут быть зарегистрированы обычным способом через YaST.

Конфигурация пользователей и групп LDAP в YaST.

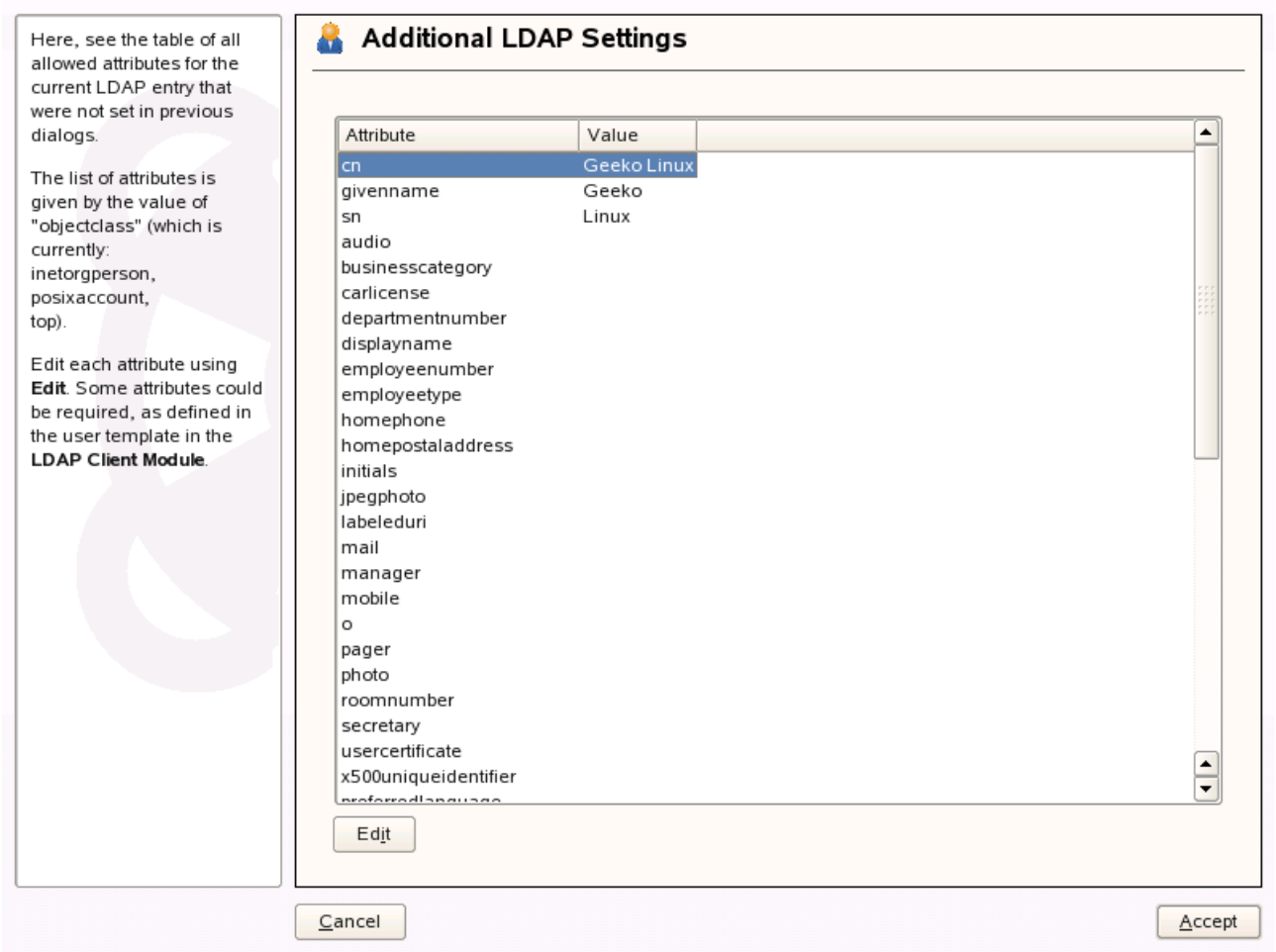
Фактическая регистрация данных пользователей и групп лишь слегка отличается от процедуры без использования LDAP. Следующие инструкции имеют отношение к администрированию пользователей. Процесс администрирования групп аналогичен.

1. Перейдите к администрированию пользователей YaST: *Security & Users > User Administration*.
2. Используйте *Set Filter*, чтобы ограничить просмотр пользователей пользователями LDAP и введите пароль Root DN.
3. Кликните *Add* и введите конфигурацию нового пользователя. Откроется диалог с четырьмя вкладками:

   - Укажите имя пользователя,логин и пароль во вкладке *User Data*.
   - На вкладке Details проверьте членство в группах, login shell и домашний каталог нового пользователя. Если необходимо, измените значения по умолчанию на значения более подходящие в данной ситуации. Значения по умолчанию так же, как и настройки пароля, могут быть определены с помощью процедуры, описанной в *Конфигурирование модулей YaST администрирования пользователей и групп*.
   - Измените или согласитесь с настройками по умолчанию в *Password Settings*.
   - Выберите вкладку *Plug-Ins*, выберите плагин LDAP и кликните *Launch* для настройки дополнительных атрибутов LDAP для нового пользователя (смотрите Рисунок 7).

4. Кликните *Accept* для сохранения настроек и выхода из конфигурирования пользователя.

Рисунок 7. *YaST: Дополнительные настройки LDAP*



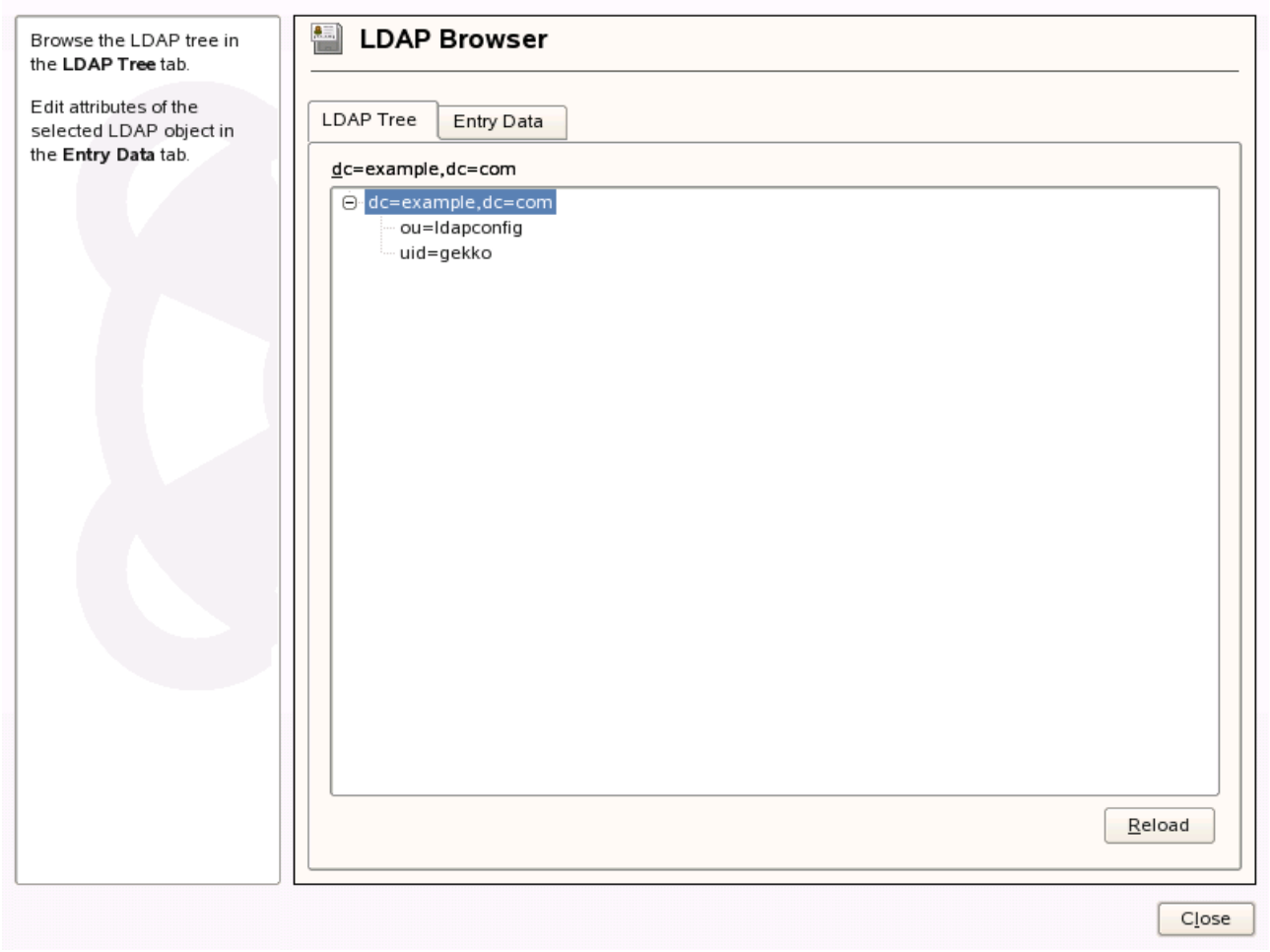
В начальной форме входа администрирования пользователей предлагается *LDAP Options*. И предоставляется возможност применять фильтры поиска в LDAP для выбора доступных пользователей или перейти к модулю для конфигурирования пользователей и групп LDAP, выбрав *LDAP User and Group Configuration*.

Просмотр дерева каталогов LDAP.

Для просмотра дерева каталогов LDAP и всех его записей удобно использовать YaST LDAP Browser:

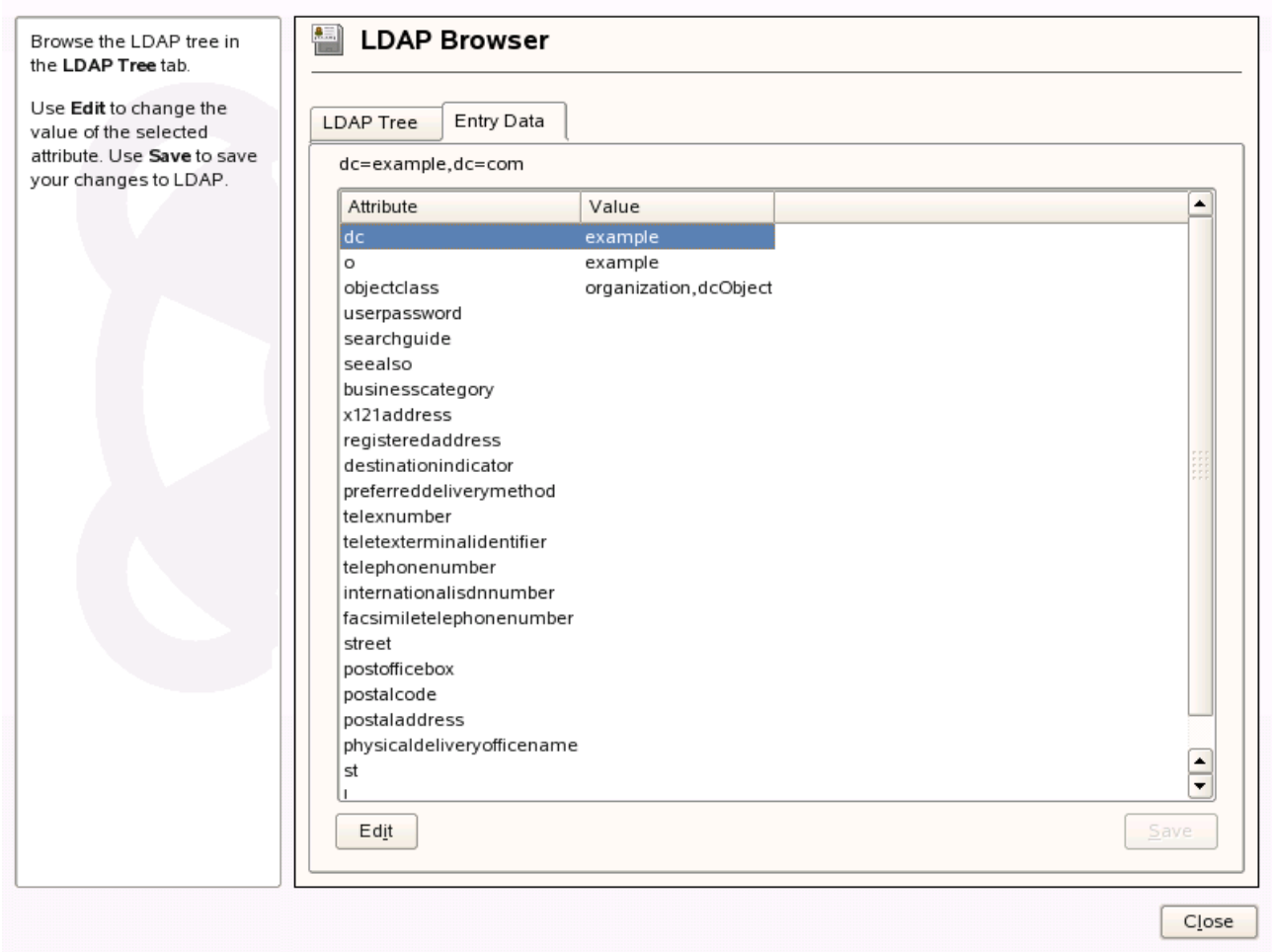
1. Зайдите под пользователем root.
2. Запустите *YaST > Network Services > LDAP Browser*.
3. Введите адрес сервера LDAP, AdministratorDN и пароль RootDN этого сервера, если вам нужен доступ к данным, хранящимся на сервере как на чтение, так и на запись.
   Или, выберите *Anonymous Access*, и не вводите пароль для получения доступа на чтения к каталогу.
   Во вкладке *LDAP Tree* отображается содержимое каталога LDAP, к которому подключена ваша машина.

   ***Рисунок 8**. Просмотр дерева каталогов LDAP.*



4. Для просмотра любых записей более подробно выберите в *LDAP Tree* просмотр и откройте вкладку *Entry Data*.
   Будут отображены все атрибуты и значения, относящиеся к этой записи.

Рисунок 9. *Просмотр данных записей.*



5. Чтобы изменить значение любого из этих атрибутов, выберите атрибут, кликните *Edit*, введите новое значение, кликните *Save* и введите пароль *RootDN*, когда он будет запрошен.
6. Выйдите из браузера LDAP, кликнув *Close*.

Дополнительная информация.

Более сложные темы, вроде конфигурирования SASL или установки реплицирующего сервера LDAP, который распределяет нагрузку на множество подчиненных, намеренно не вошли в этот раздел. Детальная информация об этих темах может быть найдена в **OpenLDAP 2.2 Administrator's Guide**.

Веб-сайт проекта openLDAP предоставляет исчерпывающую информацию для начинающих и продвинутых пользователей.

OpenLDAP Faq-O-Matic

> Очень богатая коллекция с вопросами и ответами на них по поводу установки, настройки и использования openLDAP. Посетите http://www.openldap.org/faq/data/cache/1.html.

Quick Start Guide

> Краткие пошаговые инструкции по установке вашего первого сервера LDAP. Ищите их на http://www.openldap.org/doc/admin22/quickstart.html или на установленной системе в `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`.

OpenLDAP 2.2 Administrator's Guide

> Детальное введение во все важные аспекты конфигурации LDAP, в том числе, контроль доступа и шифрование. Смотрите на http://www.openldap.org/doc/admin22/ или на установленной системе `/usr/share/doc/packages/openldap2/admin-guide/index.html`.

Understanding LDAP

> Детальное общее введение в основные принципы LDAP : http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf.

Печатная литература о 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)

Послений справочный материал по теме LDAP — соответствующие документы RFC, с 2251 по 2256.