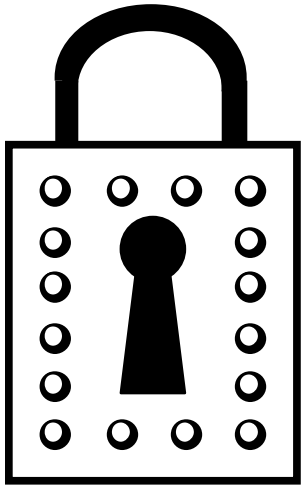
Organization of Data in LDAP



License

LDAP Directory Information Tree (DIT)

- Data in an LDAF server is organized as a hierarchical tree
  - It is usually a tree, but *alias* entries can introduce cyclic loops
  - This tree is called an *LDAP Directory Information Tree* (DIT)
  - Often, *directory information tree* is abbreviated to *directory tree*
- Each entry in the tree can be uniquely addressed by its *distinguished name* (DN):
  - Conceptually similar to /path/to/a/unix/file or C:\path\to\a\windows\file
  - However, there are differences:
    - The separator at each level is a comma (“,”) rather than “/” or “\”
    - Within a level, there is *name*=*value* instead of just *name*
    - A distinguished name is written with the most significant piece first, like in a postal address
  - Example of a distinguished name:

    cn=John Smith,ou=staff,dc=example,dc=com

Attribute names

- Let's consider that example of a distinguished name:

  cn=John Smith,ou=staff,dc=example,dc=com

- What are "cn", "ou" and "dc"?
  - They are names of *attributes*
    (similar to Java fields or C++ instance variables)
  - What follows "=" is the value of the specified attribute

- Many attributes have confusingly short names. Examples:
  - cn = common name
  - sn = surname
  - ou = organizational unit
  - dc = domain component, that is, a component in a DNS domain name

- Attribute names are *not* case sensitive
  - Example: "CN", "cn", "cN" and "Cn" are equivalent

Entries, objectClasses and attributes

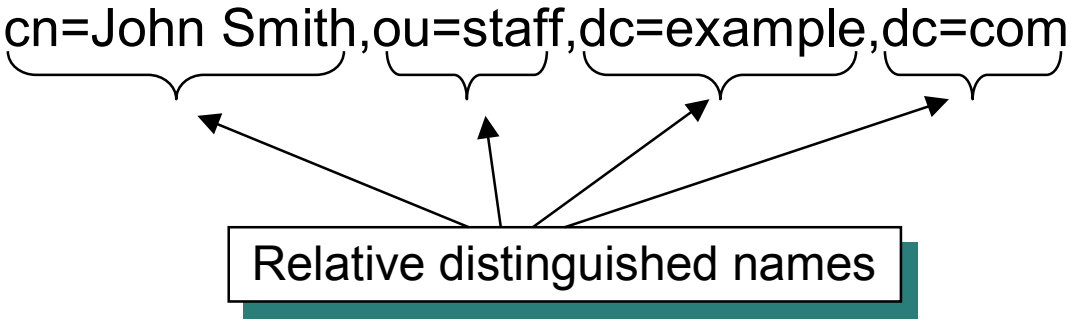
- Each *entry* in an LDAP server is an object
  - An entry (object) can contain many *attribute-name=value* pairs
  - The *objectClass* attribute specifies the entry's class (that is, its type)
- Each objectClass is defined in an LDAP schema:
  - The LDAP schema language supports single inheritance for classes
  - The definition of an objectClass specifies which attributes are optional and which are mandatory
- The schema definition of an attribute specifies if it can have one value or multiple values:
  - Example of an attribute that has multiple values:

    telephoneNumber: +1 555 967-1432

    telephoneNumber: +1 555 967-5634
  - An entry can have multiple values for its *objectClass* attribute

Relative distinguished name (RDN)

- A relative distinguished name (RDN) is a *attribute-name=value* that identifies an entry at one level in the hierarchy
- An an example, consider the following distinguished name:



- An LDAP schema does *not* specify that a particular attribute must be used in the RDN
  - Instead, you can use whatever *attribute-name=value* you prefer (as long as it uniquely identifies one entry)
  - If needed for uniqueness, you can use a “+” separated list of *attribute-name=value*
  - Example: cn=John Smith+telephoneNumber= +1 555 967-1432

LDIF Data

- An LDAP server:
  - May store data in whatever format it wants: text files, relational database, …
  - Must be able to import and export data in LDIF format
- LDIF = LDAP Interchange Format
  - It is a text-file format
- There are typically two ways to enter data into an LDAP server:
  - Use a (proprietary or open-source) GUI client that uses the LDAP protocol
  - Use an LDIF file
- Many administrators prefer using LDIF files instead of GUIs

Example LDIF data

```
dn: uid=jsmith,ou=Marketing,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: jsmith
cn: John Smith
ou: Marketing
# rest of entry deleted for brevity
```

- Notes:
  - Comments lines start with a hash sign (“#”)
  - Blank lines are used to separate entries
  - Attributes are specified as *attribute-name* followed by a colon (“:”) and a space, and then the *value*
  - The *dn* (distinguished name) pseudo-attribute specifies the entry’s location within the directory tree

Suggested reading

- An incomplete but informative online LDAP manual:

  http://www.zytrax.com/books/ldap

- The following book:

  LDAP System Administration by Gerald Carter. O'Reilly, 2003

  - Gives an overview of LDAP
  - Explains how to install and administer OpenLDAP (an open-source implementation)