This is test textddd

1

Overview of LDAP

Introducing Directories

Directories are designed to help people find their way. We've all entered an unfamiliar building and used the building's directory. Without the directory, we'd have to wander the building in search of our destination. We rely on that directory without thinking much about it, unless the information leads us to the wrong place.
Jkfdlfdl'fldf
sdfdsfdsf

With the advent of computers, there is no end of information that needs organizing so people can easily find it. Computers have always relied on directories. Even early operating systems such as DOS had a file directory so a user could keep track of data files. Directories seem to be everywhere online today, with directories that list contact information for high school graduating classes, directories that list all the movies showing, and so on. All directories have the same goal of helping us eliminate

dsf sdg ds sdgdsg sdgdg ,,, ddf

However, a directory should be more than just an efficient way to find information; it should also provide an efficient means of managing that information. If there are many sources for the information we seek, we may get contradictory or out-of-date information, and sifting through can be just as frustrating as aimless browsing. The directory should be a centrally managed repository. It's important to have a single, authoritative source for a particular type of information. That way, we don't have to search in several places for the information we want, and then painstakingly decide which information is correct.

There are many uses for a directory, beyond the direct interaction a person has when manually looking up information. Application software can leverage the information in a directory to provide a more informed and better experience. Backroom

services that work without our being aware of them can also make use of centralized information. These services provide the foundation that lets us interact in the digital world, identifying us to others, establishing our authority, allowing us to communicate with each other, even protecting us. Each of these foundational services, sometimes called *infrastructure*, must either have its own source of information about identities or rely on a common set of information. Clearly there is a benefit to having only a single set of information to manage, along with a clearly defined method of accessing this data. And there are many uses for the same piece of data, as the example that follows shows.

A directory should enable an organization to manage its business processes better. Imagine the following scenario as an example of why directories are making such an impact.

An important new executive joins your place of business. On her first day, the security officer stops her at the front door to request a long list of information for her security badge. Once she has passed by the security officer, her first visit is to the HR department, where she is asked to fill out a form with her

name, social security number, birth date, home address, department, supervisor, and so forth so she can be added to the payroll system. Then she is shown to her office. There a young technician gives her a user account and password for accesspriate information,
or better yet software could be used to interact directly with the directory and automate the entire process after the executive entered the personal data.

There are as many uses for a directory as there are types of information to organize. The amount of information being stored on computers is increasing at an exponential rate, so finding a good directory solution has become more important than ever. Fortunately for the computer industry, a common standard for directories has emerged in LDAP. This chapter introduces LDAP, highlights its capabilities, and explains why it has garnered widespread support as the best directory solution.

To this point, I have discussed directories through common examples in everyday experience. Now it is time to look at what a

The LDAP standard has been widely accepted

My Company Won't Buy a Directory

Maybe it should. The potential savings over the long run are more substantial than you think. For example, think of all the business processes that are keyed to correct and up-to-date contact information. When my contact information changed recently, I notified all the companies with which I did business. But I still had a difficult time because many businesses didn't use a single, unified repository for tracking that information. In some cases, I stopped doing business with them because I didn't appreciate spending my time troubleshooting their poor business process.

On another track, your company may just as easily end up with a directory because it is a required component for implementing some other essential product. Directories are becoming a common prerequisite. For example, almost all network operating systems require a directory to get the most out of product features.

A lot of server software requires a directory to store its configuration information. So even if your company wouldn't buy a directory to actively solve a business need, you will probably end up with one.

information. For example, my directory could have entries about people (commonly called person entries) that include a person's name along with a phone number, and perhaps other relevant personal information. There would be an entry for each person, and each entry would consist of all the personal information known by the directory about that person. The term "entry" is synonymous with the term *record* or *directory object*; these terms are used interchangeably in the literature on the subject.

The information associated with an entry is called the *attributes* or *properties* of the entry. Again, the literature is not uniform; "attribute" and "property" are used interchangeably. An entry is essentially a collection of attributes. For a person entry, the person's name is one of the attributes, as is the phone number. Depending on how the directory is defined, entries can have a set of mandatory attributes as well as a set of optional attributes. For example, my directory might have entries with mandatory common name (full name) and surname (last name) attributes along with optional phone number, fax number, and e-mail address attributes. The entry is incomplete, and therefore not allowed, without the presence of every mandatory attribute. Figure 1-1 shows an example entry for myself.

Each attribute is composed of a pair of elements. The *attribute type* is a label for the kind of information being stored. The

attribute value is the actual data being stored. For example, cn=Brian Arkills is an attribute pair, where cn (or common name) is the attribute type, and Brian Arkills is the attribute value. Incidentally, some attributes can have multiple values, which is an important feature for maximizing the flexibility of the data structure. The ability to have multiple values is a key advantage that LDAP possesses over common database solutions. Figure 1-2 shows an entry with a multivalued cn attribute.
There is a special attribute that is mandatory to all entries, called the objectclass attribute. This attribute determines what rules the entry follows. These rules govern the content of the entry by specifying the set of attributes that are mandatory and another set that is optional. The objectclass attribute is multivalued, so the set of mandatory and optional attributes for an entry is the union of all the values of the objectclass attribute. The rules may also include the possibility of restrictions on where entries of that object class can be created. At the most basic level, the object class defines what attributes can be used in the entry. The schema of the directory determines which object classes are available in the directory. The *schema* essentially defines the set of rules the directory data must follow.
A directory can have many different types of entries. A directory can have person entries with name attributes, phone attributes, and others. But it can also have entries that represent products