- Directory—A database that follows a particular method of organizing data for a directory service

  NOTE ► A Berkeley DB database, which is structured to be accessed via LDAP, is an LDAP directory.

- Data store—A specialized read-optimized database
- Open Directory—Apple's architecture for directory services for Mac OS X and Mac OS X Server
- `DirectoryService`—The process on Mac OS X and Mac OS X Server that handles directory services requests
- API—Code made available to software developers to handle various tasks in a simplified and standardized way
- Node, domain, directory node, directory domain, directory—Interchangeable terms for a directory within directory services

Understanding the BSD/local Node and BSD Flat Files

The /BSD/local node is always active. The files used to store the /BSD/local data are in the same format and file system locations that many other UNIX systems use.

The files in the /BSD/local node are called *flat files* because they are typically line-oriented. Each line of the file is treated as a separate record; an end-of-line character marks the end of a record. A list of record attributes must be specified in a certain defined order, and the attributes must be separated by a certain delimited character, usually a colon. The comma character typically separates multivalue attributes. There is generally no support in flat files for more advanced data structures such as dictionaries or binary data. XML files such as those used by the Local Default node have much more flexibility than flat files.

separated by a certain

The entries in /etc/hosts are used for host name–to-IP resolution and must be formatted with an IP address first, followed by a host name. /etc/passwd contains user records, and this file is readable by all local users. /etc/master.passwd contains the same user records, but because it may contain encrypted passwords, it is readable only with root privileges. /etc/groups contains group information.

/etc/group

MORE INFO ► For a complete list of BSD flat files and what they are used for, see *Mac OS X Server Open Directory Administration for Version 10.5 Leopard, Second Edition.*

Using `dscl`, however, you can successfully see the record:

```
client17:~ cadmin$ dscl /Search -read /hosts/bsdhost.pretendco.com
AppleMetaNodeLocation: /BSD/local
IPAddress: 10.20.1.100
RecordName: bsdhost.pretendco.com
```

Finally, the `ping` command will successfully find the IP address of the name you specified in /etc/hosts, even though there is no response at that IP address. The example uses `-c1` to set a count of only one instead of five packets and `-t1` to set the timeout to just 1 second (so you do not have to wait long for this command to fail):

```
client17:~ cadmin$ ping -c1 -t1 bsdhost.pretendco.com
PING bsdhost.pretendco.com (10.20.1.101): 56 data bytes
--- bsdhost.pretendco.com ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
```

Software such as Safari, Mail, and most command-line utilities (`ftp` and `ssh`, for example) use the entries in /etc/hosts because they use `DirectoryService` for DNS resolution.

> **NOTE ▶** If you add multiple entries to /etc/hosts, you might notice that `dscl` displays your hosts in alphabetical order. If you have multiple IP addresses for the same host name, the `IPAddress` variable can have multiple values depending on the order they appear in /etc/hosts.

You can also use `dscl` to look at the new record:

```
client17:~ cadmin$ dscl /Search read /hosts/bsdhost.pretendco.com
AppleMetaNodeLocation: /BSD/local
IPAddress: 10.20.1.100
RecordName: bsdhost.pretendco.com
```

Open Directory's BSD plug-in is always active, as the example demonstrated earlier by showing that new entries to the BSD file /etc/hosts are immediately available.

NOTE ▶ You cannot store a clear text password in the user record and have it be used for authentication.

Once you change the password with `sudo passwd`, the user's password type is converted from crypt to shadow: The user gets a shadow password hash file and an entry in the local Kerberos Key Distribution Center (LKDC).

```
client17:~ cadmin$ sudo passwd imp3
Password: [typed in cadmin's password but hidden]
Changing password for imp3.
New password: [typed in "new" but hidden]
Retype new password: [typed in "new" but hidden]
client17:~ cadmin$ dirt -u imp3
User password: [typed in "new" but hidden]
2008-03-04 20:14:30.085 dirt[25673:10b] password is : new
Call to dsGetRecordList returned count = 1 with Status : eDSNoErr : (0)
Username: imp3
Password: new
Success
client17:~ cadmin$ dscl . read /Users/imp3 AuthenticationAuthority
AuthenticationAuthority: ;ShadowHash; ;Kerberosv5;;imp3@LKDC:SHA1.07E6D260A31AA81B57C
B6F7528D5E1A0AF160BF9;LKDC:SHA1.07E6D260A31AA81B57CB6F7528D5E1A0AF160BF9;
```

dsimport

When you use ~~dslocal~~ to import users into the Local Default node, remember that the record description is quite important, but cumbersome. There are three ways of specifying this:

- Including it as the first line in your import file
- Including it as an option in the `dsimport` command
- Using the `-T <xDSStandardUser>` option and including the standard seven attributes, and then using `passwd` to set up the `AuthenticationAuthority` attribute values

Creating a Local User Record by Copying a Record File

Workgroup Manager and `dsimport` are great tools, but they call upon `DirectoryService` to make changes. What do you do if you need to make changes when `DirectoryService` is not running, perhaps when you're in single-user mode?

Understanding the Attributes of Group Records

As with local user records, each group attribute may have a differently named attribute for `dsAttrTypeStandard` and `dsAttrTypeNative`. The attribute names that you see depend on how you look at the data. If you are using `dscl` or another tool that is mediated by `DirectoryService`, you will see the `dsAttrTypeStandard` attribute names. However, if you look at the straight text files in /var/db/dslocal/nodes/Default/, you will see the `dsAttrTypeNative` attribute names. Table 1.4 lists some of the key attributes that define a group record, their names, and a brief explanation of how each is used. Note that the different attributes `GroupMembership` and `GroupMembers` have very similar names.

Table 1.4 Attributes of a Local Group Record

dsAttrTypeStandard	dsAttrTypeNative for /Default/Local	Explanation
RecordName	name	Short name for the group (for example, `admin`)
RealName	realname	Long name for the group (for example, `Administrators`)
PrimaryGroupID	gid	Numerical ID to identify the group (for example, `80`)
GroupMembership	users ~~members~~	Short names of users that are members of the group
GeneratedUID	generateduid	128-bit value guaranteed unique across space and time
SMBSID	smb-sid	SMB Primary Group Security ID
Password	passwd	Usually an asterisk
GroupMembers	groupmembers	GUIDs of users that are members of the group
NestedGroups	nestedgroups	GUIDs of groups that are members of the group

```
GeneratedUID: B5621F0A-7E38-43B4-A4BF-C0A4402660A7
GroupMembers: 2151240B-09E9-47F0-8F4D-C4007C6C391F
GroupMembership: imp1
NestedGroups: B37BDB51-DBCA-46A3-AF2E-456528324722
PrimaryGroupID: 500
RecordName: newgroup2
RecordType: dsRecTypeStandard:Groups
```

You can use the `checkmember` operation to verify that a user is a member of a group. The previous examples made user `imp1` a member of the `newgroup2` group, and the two commands below confirm this fact:

```
client17:~ cadmin$ dseditgroup -o checkmember -m imp1 newgroup1
no imp1 is NOT a member of newgroup1
client17:~ cadmin$ dseditgroup -o checkmember -m imp1 newgroup2
yes imp1 is a member of newgroup2
```

You can also use the `dsimport` command to import groups from a file. If you use the `-T xDSStandardGroup` option, your delimited import file should contain the following fields in order:

1. `RecordName`
2. `Password`
3. `PrimaryGroupID`
4. `GroupMembership`

(See the section on using `dsimport`, earlier in this chapter, for more details.)

Creating and Editing Local Groups with a Text Editor

Tools such as Workgroup Manager, `dscl`, and `dseditgroup` use calls to `DirectoryService` as a mediator for working with groups. However, you ~~cannot~~ can, but should not, circumvent `DirectoryService` by directly creating, removing, or editing files that contain records.

You can edit group records with a text editor. Each group in the Local Default node has an XML file to define it, and these files are located in /var/db/dslocal/nodes/Default/groups. Group records for the /BSD/local node are located in the flat file `/etc/group`. /etc/group.