

Choosing a Directory MATTERS

by Cheryl Walton



According to some industry analysts—including Michael Hoch, Aberdeen Group senior analyst of Internet Infrastructure—many companies think that selecting a directory service is a secondary choice. In fact, Hoch says, the directories companies select are “directly related” to the applications they select. (The Aberdeen Group [www.aberdeen.com] is an IT marketing analysis and positioning services firm.)

In other words, for many companies, choosing a directory takes a back seat to choosing an application. This practice may simply be habit. After all, in the past, most applications—and even network operating systems—included directories that served a single purpose. Companies had little choice but to deploy these directories along with the applications that used them.

Of course, today companies have other options: They can purchase a multipurpose directory service, which can do more than just meet the needs of a single application. A multipurpose directory service can consolidate information residing in scores of single-purpose directories. As a result, companies can use one set of management tools to manage information in one directory, rather than using multiple management tools to manage information in scores of directories. Eliminating this management burden can save companies money. (For more information about saving money, see “The Sound of Saving Money Is Music to Your CEO’s Ears” on p. 8.)

Multipurpose directories can also provide identity management and access control services for commercial and custom e-business applications. Web services applications—which industry leaders such as Novell, Microsoft, and Sun think will comprise the next generation of e-business applications—also require directories that can provide identity management and access control services. (Web services are application components that use Internet standard technologies to integrate existing network resources and to make these resources available over company intranets and the Inter-

net. For more information, see “Web Services: The Next Big Thing?” on p. 6.)

In addition, more and more application vendors are writing applications that use these multipurpose directories, rather than internal, single-purpose directories. Because these applications aren’t tied to a specific directory, your company may soon be shopping for a directory service as a key infrastructure application. How does a company that is unaccustomed to looking at a directory as a key piece of its networking infrastructure select the directory that can best meet its needs?

One tool for measuring the capabilities of directories is the Full Service Directory (FSD) model. This model includes four main categories of directory services—discovery, security, storage, and relationship services—each of which includes several subcategories. (See Figure 1 on p. 11. For more information about the FSD model, see “A Model Directory” on p. 10.)

Of course, you can also measure a directory’s capabilities directly against your company’s needs—provided you understand the directory’s capabilities and your company’s needs. This article relies on the FSD model and other important criteria for evaluating directory services to compare the capabilities of the following popular directories:

- Novell eDirectory 8.7, which will be available later this year (hereafter called *eDirectory*)

In This Article

Access Control	13
Backup Services	18
Cross-Platform Services	20
Discovery Services	6
Distribution and Integrity Services	18
Enforcement Services	17
A Model Directory	10
Namespace	10
New Ways to Search	12
Notification Services	9
Partitioning Services	18
Protocols	20
Relationship Services	19
Saving Money	8
Security Services	12
Storage Services	17
Web Services	6

Please visit our advertiser ACCPAC at www.faxserve.com

Web Services: The Next Big Thing?

Has the term *web services* popped up on your company's radar yet? It has on Novell's and in a big way. Web services are interoperable application components that you make available to users or other web services over your company's intranet or the Internet. What makes web services interoperable isn't the language in which these application components are written. In fact, the underlying application language of a web service doesn't matter.

Web services are interoperable because they are encapsulated using Simple Object Access Protocol (SOAP) and other Internet standard protocols such as eXtensible Markup Language (XML), Web Services Description Language (WSDL), and Universal Description and Discovery Integration (UDDI). SOAP, which is an XML-based protocol that uses HTTP and TCP/IP, enables one web service to communicate with another web service. XML provides a standards-based interface to the underlying application logic of a web service. WSDL is an XML-based language that web services use to describe themselves, and UDDI provides a registry through which web services can advertise themselves and find other web services.

Why is Novell interested in web services, and why should your company be? Novell is interested in web services because these services are quintessentially Net services and can therefore help companies move their business to the Internet. Users can access web services over virtually any kind of network, and web services work with all kinds of operating systems.

Your company should be interested in web services because they can use these services to integrate applications, including applications already running on its network and on its business partners' networks. In other words, web services can help your

company use the resources it already has to provide new services for customers, partners, and employees.

For example, suppose your company has two custom applications—one that tracks customer orders and one that tracks items in stock. Further suppose your company's supplier has a custom application to track your company's orders.

Your company could deploy a web service that receives new customer orders and communicates order information directly to the custom application that tracks customer orders. This web service could also notify a second web service that an order for a particular item had just been placed.

This second web service could then access the items-in-stock application to determine whether or not the requested item was in stock. If the item were not in stock, this second web service could notify a third web service running on the supplier's network. This third web service could then access this supplier's custom application to order the out-of-stock item on behalf of your company.

Of course, if your company is going to use web services, it probably wants to control access to those services through a Full Service Directory (FSD), such as Novell eDirectory. In this way, you can provide comprehensive and cohesive security for all the resources on your company's network, including web services. (For more information about FSDs, see "A Model Directory" on p. 10.)

Incidentally, Novell recently submitted a new specification to the Internet Engineering Task Force (IETF). This proposed standard provides a way to represent UDDI data in a Lightweight Directory Access Protocol (LDAP) directory, which will enable a UDDI server to run on eDirectory via LDAP. (For more information about the proposed standard, visit www.webservices.org/index.php/article/articleview/423/1/8. For more information about web services, visit www.webservices.org.)

- Microsoft Active Directory for Windows 2000 (hereafter called *Active Directory*)
- Sun Open Net Environment (ONE) Directory Server 5.0 (hereafter called *Sun ONE*)

DISCOVERY SERVICES: INDISPENSABLE FOR ANY DIRECTORY

The first main category of services in the FSD model is discovery services, which enable you to browse through, search for, add to, delete, and retrieve information contained within a directory. In FSDs, discovery services must include features—such as Lightweight Directory Access Protocol (LDAP) support—that enable applications to access the directory. LDAP version 3 (v3) is currently the most widely used protocol through which applications access the discovery services within directories. Support for this Internet Engineering Task Force (IETF) standard is *de rigueur* for FSDs.

Support for LDAP

To what extent do eDirectory, Active Directory, and Sun ONE support LDAP v3? eDirectory fully supports LDAP v3 and includes several new LDAP features that enhance applications' options for using eDirectory's basic discovery services. For example, eDirectory supports Persistent Search, which is a proposed extension of the LDAP v3 standard. (For more information about Persistent Search and other LDAP discovery features, see "New Ways to Find and Seek" on p. 12.)

Like eDirectory, Sun ONE is fully compliant with the LDAP v3 standard. In fact, Sun ONE is based on LDAP v3 and supports most of the features defined in this standard as well as some proposed features, including Persistent Search. Because eDirectory and Sun ONE so fully support the LDAP v3 standard, commercial LDAP-based applications can use these directories with little or no extra configuration effort on your part.

Although Active Directory supports LDAP v3, this directory doesn't always support the standard interpretation of LDAP v3 features. For example, LDAP v3 defines auxiliary classes that enable you to assign attributes to an object without affecting other objects of the same type. For example, you can add auxiliary class attributes to some User objects without affecting other User objects.

However, Active Directory does not support this LDAP feature. If you add an auxiliary class to any one object in an Active Directory forest, Active Directory adds the attributes defined in this auxiliary class to all other like objects in the forest. In other words, Active Directory enforces schema consistency among all objects within a given Active Directory forest.

As Mike Neuenschwander, analyst at the Burton Group, explains, some LDAP-based applications search for objects that have a specific auxiliary class attribute. This class attribute identifies the particular objects in which the applications are

Please visit our advertiser Novell Education at www.novell.com/education.

The Sound of Saving Money Is Music to Your CEO's Ears

A few years ago, *Network Magazine* published an article describing how ON Semiconductor consolidated information residing in 100 separate directories in one central directory—namely Novell eDirectory. (“Business Case: Cut Costs and Pocket the Savings With Directories,” Dec. 5, 2000. You can download this article from www.networkmagazine.com/article/NMG20001128S0002.) Naturally, managing information in one directory—as opposed to 100—saved this company money. According to the article, before undertaking this directory project, the director of workforce productivity at ON Semiconductor ran the Return On Investment (ROI) numbers. This director estimated how much it cost the company to manage user information in 100 directories and how much it would cost to manage that information in just one directory. As a result of implementing eDirectory as a central store for user information, the director calculated that his company could save U.S. \$3.8 million.

However, this traditional time-equals-money calculation didn't, and couldn't, take into account the serendipitous savings that resulted from having all of the company's user-related information in one directory. For example, because its user information was consolidated, ON Semiconductor was able to reconcile two-way pager bills with user information in eDirectory. The result? ON

Semiconductor saved U.S. \$300,000 on its pager bill simply by identifying which pagers were no longer in use.

ON Semiconductor's story illustrates the money-saving potential of consolidating information in a directory, and this potential is one of the main reasons that Full Service Directories (FSDs) are a topic worthy of your attention. FSDs can act as a central identity store for many, if not all, of your company's network resources. (For more information, see “A Model Directory” on p. 10.)

Among all of the FSDs and FSD wannabes on the market today, eDirectory has the cost-saving advantage of running on all of your network's major operating systems, including NetWare. By using eDirectory as your central directory, you won't need to spend a lot of money overhauling your company's network in order to save money on managing redundant information.

Furthermore, as the most scalable directory on the market, eDirectory is unlikely to run out of room as your company grows. Finally, the number of applications that work with eDirectory just keeps growing.

If you would like a reasonable estimate of the money your company could save by consolidating directory information in eDirectory, you can use the free preview of ROINow, which is an ROI tool from CIOview. (This free preview is available at www.cioview.com/products/index.htm.) ●

interested. According to Neuenschwander, Active Directory's lack of support for the standard interpretation of LDAP auxiliary classes “can cause problems” for applications that use auxiliary classes. (Burton Group [www.burtongroup.com]) is an IT consulting and research firm.)

To demonstrate the disadvantages of Active Directory's interpretation of auxiliary classes, suppose an application were interested only in User objects that include an auxiliary class attribute that identifies a specific subset of users, such as multilingual users. Because Active Directory automatically adds this auxiliary class attribute to every User object in the Active Directory forest, this application's search for multilingual users would return all of the User objects.

In addition, Active Directory does not support inetOrgPerson, which is a proposed and *de facto* standard class for users. Most LDAP-based applications use inetOrgPerson. Because Active Directory does not support inetOrgPerson, however, many off-the-shelf applications that require inetOrgPerson can use Active Directory only after “some unnatural wrangling with the Active Directory schema,” Neuenschwander says.

Although Microsoft provides a support pack for inetOrgPerson, inetOrgPerson is not part of Active Directory's User object class in this support pack. As a result, users

represented by inetOrgPerson objects cannot authenticate to Active Directory domains, nor can these users access the resources available through these domains.

“It is also not possible to associate User objects with inetOrgPerson objects,” Neuenschwander notes. Therefore, you must manage inetOrgPerson objects separately.

Furthermore, using the inetOrgPerson support pack may have long-term implications. Neuenschwander explains, “If you implement the inetOrgPerson support pack that is available today, you'll face a data migration problem when you upgrade to Microsoft .Net server, which will support inetOrgPerson.”

LDAP Is Not the Only Access Protocol on the Block

Although LDAP support is important, LDAP isn't the only standard protocol that applications use to access discovery services in FSDs. For example, Novell DirXML and Novell Account Management use eXtensible Markup Language (XML) to access discovery services in eDirectory. (Novell DirXML [www.novell.com/products/edirectory/dirxml]) is a data-sharing application that can synchronize information across directories. Novell Account Management [www.novell.com/products/edirectory/accountmanagement/details.html] enables you

to manage Active Directory, NT domains, and directories on Solaris and Linux centrally through eDirectory.)

In addition, eDirectory now supports Simple Network Management Protocol (SNMP). As a result, applications—such as Novell iManager 1.5 and Hewlett-Packard OpenView—can use SNMP to make changes in eDirectory. (iManager is a web-based management application that is included with eDirectory. For more information about iManager 1.5, see “Novell iManager: Keeping eDirectory Management Simple,” *Novell Connection*, July 2002, pp. 6-16. You can download this article from www.ncmag.com/past.)

Sun ONE also supports SNMP. However, we could not verify if Active Directory supports SNMP. We did not receive verification from Microsoft before this article went to press. (You may want to contact Microsoft if SNMP support is important for your company.)

The Future Is Closer Than You Think

In the near future, Novell plans to add support for two protocols that will enable web-service applications to access discovery services in eDirectory: Simple Object Access Protocol (SOAP) and Universal Description, Discovery, and Integration (UDDI). SOAP is a World Wide Web Consortium (W3C) standard that defines how applications can use XML-based calls

to access directory services. (For more information about SOAP, visit www.w3.org/TR/soap12-part1.)

eDirectory actually uses SOAP internally to make traditional management utilities, such as the DSREPAIR utility, available through iManager 1.5. eDirectory will soon support SOAP access for web-service applications.

The UDDI specification defines how you can publish information about available web services and access this information from client applications. (For more information about UDDI, visit www.uddi.org. For more information about SOAP, UDDI, and web services, see "Web Services: The Next Big Thing?" on p. 6.)

According to a Microsoft spokesperson, Active Directory in Windows 2000 supports SOAP through Directory Services Markup Language version 2 (DSML v2), which is an XML-based specification for formatting information in directory services. Active Directory will support UDDI in its .Net version through Enterprise UDDI Services (a new feature for web

services discovery in .Net servers). Sun also plans to add support for UDDI (and for SOAP) in future versions of Sun ONE.

The Bottom Line

In general, the more access protocols a directory supports, the more applications can access the directory's discovery services. eDirectory supports more protocols for accessing discovery services—14 protocols at present—than do Active Directory and Sun ONE combined. (For more information about the access protocols these three directories support, see "Let Me Count the Protocols" on p. 20.)

NOTIFICATION SERVICES: KEEPING YOU IN THE LOOP

The discovery services outlined in the FSD model also include notification services, which—as the name suggests—can notify applications when changes are made within the directory. Notification services are important because many applications depend on notification services to perform their tasks.

eDirectory provides notification ser-

vices through an event engine, which can dynamically notify applications that have registered when something within the directory changes. Applications can use this event engine to obtain up-to-the-second information from eDirectory. For example, NetVision Policy Management Suite and Blue Lance LT Auditor+ 8.0 use eDirectory's event engine to notify you when changes occur in the directory. (NetVision Policy Management Suite [www.netvision.com] and LT Auditor+ 8.0 [www.bluelance.com] are monitoring and alerting applications that include directory-based intrusion detection systems.)

Many of Novell's directory-enabled applications also take advantage of this event engine in one way or another. For example, Novell DirXML uses this event engine to synchronize information in various directories with information in eDirectory. iManager 1.5 uses this event system to enable web-based access to directory monitoring and repairing utilities (such as DSTRACE and DSREPAIR) that heretofore have been available only

Please visit our advertiser St. Bernard Software at www.openfilemanager.com.

A Model Directory

The Full Service Directory (FSD) model was developed in 2000 as a tool for identifying directories that are capable of being a central data and identity store for today's network resources. According to the FSD model, an FSD must offer several key services in the following four categories:

- Discovery services
- Security services
- Relationship services
- Storage services

DISCOVERY SERVICES

- **Publication.** These services provide the ability to write information to the directory.
- **Search.** These services provide the ability to find information in the directory.
- **Identification.** These services provide the ability to uniquely identify objects in the directory.
- **Retrieval.** These services provide the ability to read information from the directory.
- **Notification.** These services enable directories to notify applications about events that occur within the directory.
- **Indirection.** These services enable you to refer to objects in the directory by their common names. Users can then recognize these objects even if the location of these objects changes.

SECURITY SERVICES

- **Authentication.** These services enable the directory to identify users and applications that access the directory.
- **Key Management.** These services enable the directory to provide Public Key Infrastructure (PKI) support.
- **Qualification.** These services enable directories to provide varying levels of access to resources, as opposed to all-or-nothing access.
- **Authority.** These services enable you to determine who is authorized to do what with directory-controlled resources.
- **Enforcement.** These services enable directories to enforce access controls.
- **Audit.** These services enable you to discover security breaches.

RELATIONSHIP SERVICES

- **Federation.** These services enable you to establish a relationship between separate directories.
- **Organization.** These services enable you to organize logically the objects a particular directory holds.
- **Collection.** These services enable you to define collections of related objects.
- **Registration.** These services enable you to provide unique identifiers—such as Global Unique Identifiers (GUIDs)—for objects within the directory. (eDirectory and Active Directory use GUIDs to ensure that links between objects don't break when objects are renamed or moved. Because Sun ONE does not use GUIDs, broken links in Sun ONE are commonplace unless you use a special plug-in—which is called a *referential integrity plug-in*—to help alleviate this problem.)
- **Reference.** These services enable information in one object to refer to information in other objects.
- **Policy.** These services enable you to manage your network through policies that are linked to objects within the directory.
- **Subscription.** These services enable users to subscribe to objects within the directory.
- **Inference.** These services enable you to create inferred relationships between one or more objects.

STORAGE SERVICES

- **Persistence.** These services keep your company's data uncorrupted and safe from hardware failures.
- **Integrity.** These services keep your company's directory tree synchronized.
- **Segmentation.** These services enable you to break up large directories into smaller, more manageable portions while still maintaining a single, cohesive namespace.
- **Distribution.** These services enable you to distribute a directory tree across two or more servers.
- **Indexing.** These services speed up the process of finding commonly requested information.
- **Caching.** Like indexing services, caching services increase the speed with which the directory can access and return information.
- **Classification.** These services enable you to extend the number of objects and attributes in your company's directory schema. ●

at the NetWare server console. Novell also recently made eDirectory's event engine available for LDAP-based applications that rely on notification services.

If directories don't have event engines, they can provide notification services through change logs, as Active Directory and Sun ONE do. Change logs are files in which directories log internal changes. Applications that need to know about changes within Active Directory and Sun ONE poll these directories at regular intervals to gather information from these files.

Depending on the length of the polling interval, the information these applications receive may be outdated, which is

acceptable if the timeliness of the information isn't critical. Sometimes, however, the timeliness of information that applications use is critical. For example, a security application that monitors unsuccessful attempts to log in to the directory needs up-to-the-second information so that this application can notify you about suspicious login attempts.

NAMESPACE: A NAME AND A PLACE FOR EVERYTHING

A directory's namespace determines more than simply the means by which you can uniquely identify the objects within the directory (although a directory's

namespace does provide this valuable discovery service). The namespace also affects the way you can organize the objects within a directory and defines the relationship an object has to other objects.

Name That eDirectory Object

The namespace eDirectory uses is based on the X.500 naming model, which the International Telephone and Telegraph Consultative Committee (CCITT) standard defines for creating hierarchical, distributed, global directories. (CCITT is now called the *International Telecommunications Union-Telecommunications Standards Sector*.) Because the namespace

eDirectory uses a hierarchical namespace, the objects within eDirectory are uniquely identified by distinguished names, which reflect the relationship these objects have to other objects within the eDirectory tree.

Furthermore, because eDirectory's namespace enables you to define hierarchical relationships between objects, you can create an eDirectory tree that mirrors the structure of your organization. That is, this namespace enables you to logically organize the objects within your organization's eDirectory tree. This organization, in turn, can help you determine which resources to allocate to which objects.

For example, suppose all of your organization's employees need to access its corporate white pages, which is an application running on a server located at corporate headquarters. You can create a Server object that represents this server within an Organization object that represents your entire company. You can also create within this Organization object User objects that represent the top executives who work at your company's corporate headquarters.

On the other hand, only your company's New York branch office handles accounting tasks for your company. Therefore, you can create an Organizational Unit (OU) object to represent this branch office and then place all of the objects that represent your company's accounting resources in that OU object.

Because you can logically organize objects in relationship to other objects, you can easily find and, therefore, manage objects in eDirectory. Furthermore, this hierarchical namespace enables eDirectory to infer certain things about objects based on the location of these objects within the directory tree. This inference capability is called *inheritance*, and inheritance can further simplify the task of managing objects within an eDirectory tree.

For example, because all of your organization's employees need to access its corporate white pages, you can specify access rights to this application at the Tree object, which is a container object representing the top level of the eDirectory tree. eDirectory then automatically grants these rights to all of the objects below the Root object. You don't have to specify these rights for each User object. Instead, these User objects inherit these rights based on their relationship to the Root object.

Namespaces That Fall Flat

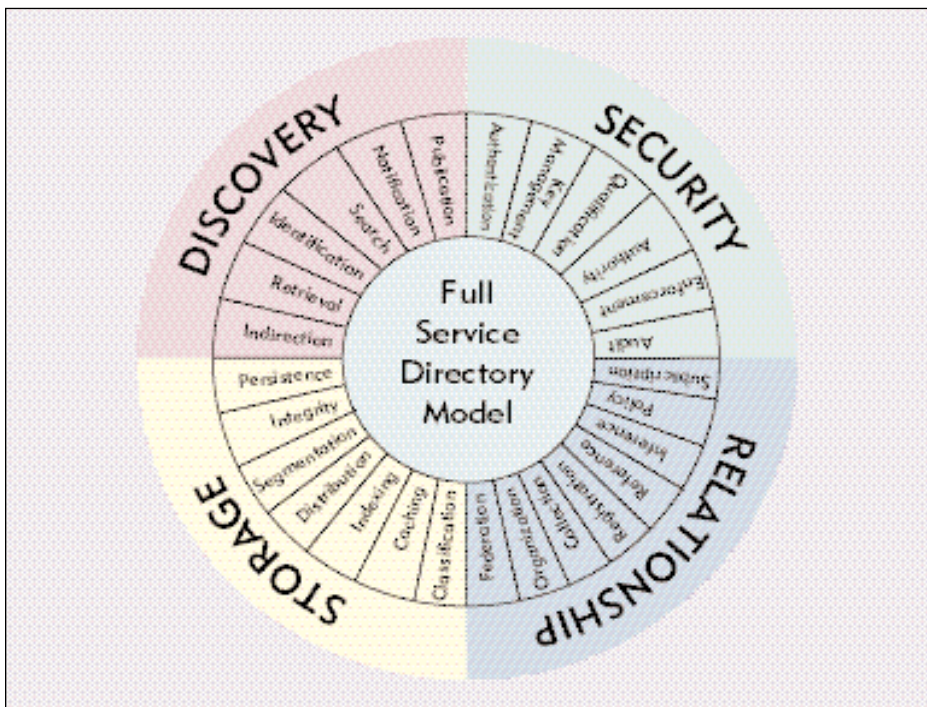


Figure 1. The Full Service Directory (FSD) model can help you select a directory that functions as a key infrastructure component of your company's network. Directories that closely fit this model can provide services for a variety of directory-enabled applications.

Please visit our advertiser Biscom at

www.biscom.com

New Ways to Find and Seek

With Novell eDirectory 8.7, which will be released later this year, eDirectory now includes support for three Lightweight Directory Access Protocol version 3 (LDAP v3) features. These features will make it easier for you and your company's directory-enabled applications to find information that is stored within the directory:

- Persistent Search
- Extensible Match
- Superior Referrals

PERSISTENT SEARCH

Persistent Search is a proposed extension of LDAP v3 that enables applications to maintain a connection to an LDAP-compliant directory even after that directory has returned the results of a search request. To illustrate this feature's usefulness, suppose you are using an application that supports Persistent Search. You want this application to provide a list of all User objects for which the common name attribute has a value that begins with the letter A. However, this application supports increments of only 100 search results, and thousands of User objects in your company's eDirectory tree have a common name that begins with A.

Using Persistent Search, this application could receive search results for this request in increments of 100 common names without having to establish a new connection to the directory for each increment. Without Persistent Search, on the other hand, this application would have to establish a new connection and issue a new search request for every increment. (For more information about Persistent Search, visit www.ietf.org/internet-drafts/draft-smith-psearch-ldap-01.txt.)

EXTENSIBLE MATCH

Extensible Match is an LDAP v3 search filter that enables you to define your own matching rules for LDAP searches. For example, suppose you want a list of all administrative assistants who work in sales, and your company's eDirectory tree includes three

organizational unit (OU) container objects, each of which includes a Sales container object. Further suppose that each of these Sales container objects contain User objects and that some of these User objects include an attribute (organizationalRole), the value of which identifies these particular users as administrative assistants.

Using the Extensible Match filter, you can build an LDAP search request that returns the names of all the administrative assistants in all of the Sales container objects in all of the OU objects—all in one fell swoop. To obtain these same names without using the Extensible Match filter, you would first need to construct a search request to locate the Sales container objects in each of the OU objects. You would then need to construct search requests to locate administrative assistants within each of these Sales container objects. Without using the Extensible Match filter, you would need to perform four separate searches to obtain the same results that you could obtain when performing a single search using the Extensible Match filter. (For more information about Extensible Match, see Request for Comments [RFC] 2251, p. 27. You can download this document from www.ietf.org.)

SUPERIOR REFERRALS

A Superior Referral is an LDAP v3 extension that enables an eDirectory tree to refer directory requests to other directories. This capability enables eDirectory to function as part of a larger directory tree that comprises two or more separate directory trees. In other words, this feature can help companies federate two or more directory trees to function as a single directory tree.

For example, suppose your company acquired a subsidiary that is using eDirectory. Your company is not using eDirectory but is using an LDAP v3-compliant directory. Your company wants to add the subsidiary's directory tree as a branch of its corporate tree.

Using the Superior Referrals feature, you can configure eDirectory to consult your company's corporate directory to fulfill requests for information that resides in that directory. You can also use Superior Referrals to configure your company's corporate directory to consult eDirectory to fulfill requests for information located in eDirectory. ●

Although both Active Directory and Sun ONE have object-oriented namespaces that superficially resemble the X.500-based namespace, these directories don't actually create in the underlying database hierarchical relationships between objects, as eDirectory does. Instead, Active Directory and Sun ONE store objects in flat-file databases (which are single database files that contain lists of objects and object attributes).

Because these objects are not related to one another within these databases, Active Directory and Sun ONE uniquely identify objects by common names, rather than by distinguished names. Therefore, these directories enforce unique common names within each flat-file database.

Requiring a unique common name for every object in your company's directory

tree may seem more like an asset than a liability. After all, Active Directory and Sun ONE enforce such a naming policy for you. eDirectory, on the other hand, requires you to search the entire directory tree to ensure that a particular common name is unique. Because Active Directory and Sun One do not use a hierarchical namespace, however, you forego other important management benefits—notably inheritance. Only eDirectory provides inheritance, which relies on a hierarchical database.

SECURITY SERVICES: A DIRECTORY ESSENTIAL

A recent online newspaper article explains how Experian, a large credit-reporting agency, learned the hard way that if it offers confidential services over

the Internet, it must provide ironclad access control to those services. Using an illicitly gained access code that identified them as an employee of Ford Motor's credit company in Grand Rapids, Michigan, hackers gained access to 13,000 consumer credit reports, including credit card numbers, social security numbers, and credit ratings. ("13,000 Credit Reports Stolen by Hackers," *The New York Times on The Web*, May 17, 2002. Past articles are available for a fee from www.nytimes.com.)

Obviously, Experian didn't adequately control access to its confidential information. Not coincidentally, the type of ironclad access control that may have prevented this situation is a hallmark of FSDs.

As a first step to controlling access to your company's network resources, FSDs

provide authentication services, which ask and verify the answer to this simple but very important question: Who are you?

eDirectory

Of course, eDirectory, Active Directory, and Sun ONE ask and verify the answer to this question. What differentiates the authentication services these three directories provide is the number of authentication methods each directory supports and the number of authentication methods these directories can require for one login.

eDirectory supports a multitude of authentication methods, including several new options for LDAP authentication. In addition to simple username-password authentication for LDAP users, eDirectory now supports Simple Authentication and Security Layer (SASL) authentication. SASL authentication methods include SASL External, which enables LDAP users to authenticate using the following:

- X.509 Public Key (PK) certificates
- Message Digest 5 (MD5)
- NMAS_LOGIN

For X.509 PK connections, eDirectory supports Transport Layer Security (TLS) and Secure Sockets Layer (SSL). (TLS is an open-standards implementation of SSL 3.0. You can download the proposed IETF standard for using TLS with LDAP v3 at www.ietf.org/rfc/rfc2830.txt.)

MD5 enables LDAP-based applications to send passwords securely over a clear-text connection. By using the LDAP bind operation, NMAS_LOGIN enables applications to use any of the authentication methods available in Novell Modular Authentication Service (NMAS) 2.0 Enterprise Edition. NMAS authentication methods include digital certificate, smart card, token, and biometric authentication.

NMAS 2.0 Enterprise Edition is available as a separate product. However, eDirectory ships with NMAS 2.02 Standard Edition, which enables you to strengthen security for LDAP username and password authentication. NMAS 2.02 includes modules for simple password and enhanced password support. The simple password module enables eDirectory to use advanced encryption algorithms such as Secure Hash Algorithm 1 (SHA1) and MD5 to secure traditional LDAP passwords as they cross the network (or the Internet). Enhanced password sup-

port, on the other hand, enables you to implement password policies for LDAP passwords. For example, you can create a policy that requires complex case-sensitive passwords.

In other words, eDirectory now provides for LDAP users all of the authentication methods it provides for traditional eDirectory users. Furthermore, using NMAS Enterprise Edition 2.0, eDirectory supports graded authentication, which enables you to require different and multiple authentication methods for individual resources.

For example, using NMAS 2.02, eDirectory can require a simple username and password combination to access resources that require only minimal access control. In contrast, eDirectory can require biometric authentication or a username-password and biometric authentication for confidential resources that require more stringent access control.

Active Directory and Sun ONE

In comparison, Active Directory supports three methods of authentication:

- Username password authentication
- X.509 PK certificates
- Smart card

To secure the authentication process, Active Directory uses a proprietary implementation of Kerberos 5 encryption and SSL (for certificate authentication).

Because Sun ONE is wholly LDAP-based, it supports only authentication methods defined in the LDAP v3 specification. Specifically, Sun ONE supports simple password, simple password over SSL, X.509 PK certificate (using SSL), and SASL.

Like eDirectory, Sun ONE supports graded authentication. However, to configure graded authentication for resources that are under the control of a Sun ONE directory, you must manually configure Access Control Instructions (ACIs). ACIs are statements that list the rights an object has when accessing specific resources.

In contrast, to configure graded authentication for resources to which eDirectory controls access, you can use a web-based utility with a graphical user interface—namely iManager 1.5. Furthermore, because LDAP can bind a user to an LDAP directory server using only one method of authentication, Sun ONE cannot require two or more authentication

methods for access to confidential resources, as eDirectory can.

ACCESS CONTROL: WHAT CAN YOU DO?

Just because a user authenticates and receives rights to access a particular resource, that user shouldn't have *carte blanche* to do what he or she wants to the resource. eDirectory, Active Directory, and Sun ONE all enable you to assign to users levels of access that are based on these users' needs. For example, a department manager may need rights to add a new employee's name and department to a white pages application. That new employee, on the other hand, may need only the right to view entries in this application.

Of course, users are not the only entities that need access to network resources. For example, e-business applications often need to access backend resources. Like the needs of users, the needs of these applications vary. That is, some applications need read-only access to information, and other applications need read-write access.

**Please visit our
advertiser DSI
Consulting Inc. at
www.dsi-consulting.com**

Please visit our advertiser Novell Inc. at www.novell.com.

Please visit our advertiser Novell Inc. at www.novell.com.

eDirectory

You can assign any object in the eDirectory tree varying levels of access to any other object. This ability to assign rights to any object is important for two reasons:

- You can use this ability, eDirectory's hierarchical structure, and eDirectory's unique inheritance capabilities to create implicit rights assignments based on users' needs. For example, you can assign a departmental container object the rights to access departmental resources. The objects below this container object (User objects that represent departmental employees) then inherit these rights. Objects above this container object, on the other hand, do not have rights to these departmental resources.

As a result, you don't need to explicitly assign rights to each individual User object. Furthermore, if you move one of the User objects to another container, that object automatically loses the inherited rights associated with the container object from whence it came and automatically receives the inherited rights associated with its new container object. This ability to assign access rights based on objects' location in the eDirectory tree simplifies the management of these objects.

- You can assign non-User objects the rights to perform only the tasks that these objects need to perform. That is, if a particular application needs only to read information from backend resources, you can assign to this application read-only access rights to these resources. Furthermore, like User objects, non-User objects must authenticate to eDirectory before eDirectory grants these rights.

Because eDirectory enables you to assign rights to any eDirectory object, you don't need to use specialized User objects to enable applications to access resources on your company's network. With Active Directory and Sun ONE, however, you must use specialized User objects, which often have higher-than-normal access rights and therefore can present a tempting target for hackers. (These specialized User objects are discussed later in this article.)

To assign access rights in eDirectory, you use standard eDirectory management utilities—iManager 1.5 or Con-

soleOne. As mentioned earlier, iManager enables you to manage eDirectory over the Internet.

Active Directory

Using Active Directory, you can assign access rights only to User and Group objects. You must also explicitly grant these rights in Access Control Lists (ACLs), which can apply only to the User and Group objects within a given domain. If you want an ACL to apply to all of the User and Group objects in an Active Directory forest, you must copy this ACL to

You can assign any object in the eDirectory tree varying levels of access to any other object.

each domain. You must then manage this ACL separately in each domain.

Incidentally, when you create an ACL to control access to a resource that is available through Active Directory, Active Directory stores this ACL on the object that represents this resource. Active Directory also stamps the access rights described in this ACL on each User and Group object specified within this ACL. Depending on the number of resources to which you need to control access, this practice can cause the Active Directory database to grow disproportionately to the size of the network.

Like rights for commonplace users and groups, administrative rights in Active Directory apply only to a particular domain. You must also assign and manage these rights in every domain within an Active Directory forest. Furthermore, because Active Directory supports access control only for User and Group objects, you must use a specialized user object—called a *Service Account object*—to grant access privileges to non-User objects, such as Application objects. As mentioned earlier, these specialized User objects have higher-than-normal levels of authority and therefore present a security risk.

Active Directory includes a management utility—the Microsoft Manage-

ment Console (MMC) utility—to simplify the task of creating and managing ACLs in Active Directory. (MMC is Microsoft's utility for managing services in Windows 2000, including Active Directory services.)

Sun ONE

Like eDirectory, Sun ONE enables you to assign to all of the objects within the directory tree rights to access other objects. To do this, you create ACIs that specify these rights. However, Sun ONE does not support access rights equally for all Sun ONE objects.

Specifically, non-User objects—such as Application objects—cannot log in to the Sun ONE directory as themselves to receive their own access rights. Instead, these objects must receive access rights through specialized User objects. One such Sun ONE object, the Root User object, poses a particularly great security risk.

In fact, as Neuenschwander explains, the Root User object is a “built-in,” unrestricted Sun ONE User object that creates several security risks. For example, you cannot place access controls on this object nor can you delete it. In addition, the password for this object is stored in a text file on the hard drive of the server upon which Sun ONE is running. Consequently, the Root User account can be an especially tempting target for hackers—particularly if you don't rename this object.

Although Sun ONE does not support inherited rights, you can approximate some of the management benefits of inherited rights by including LDAP search filters in Sun ONE ACIs. For example, suppose you want all of your company's accounting users to have read-write access to its accounting resources. To grant these privileges without having to specify them for each user in your company's accounting department, you can create an ACI that uses a search filter to grant these privileges to users whose departmental title is accounting. Of course, to take advantage of this feature, your company's Sun ONE directory must include information (in this case, departmental information) that makes this feature useful.

Whether or not you use search filters with ACIs, Sun ONE includes only rudimentary management tools to help you create these ACIs. Furthermore, Sun ONE provides no tools for testing rights assignments, as eDirectory does. Therefore, you must thoroughly understand how

to configure LDAP searches, and you must also understand your company's tree design to create ACIs. In fact, because creating and maintaining ACIs can be confusing, Sun recommends that you create only a few general ACIs at or near the top of the Sun ONE directory tree.

ENFORCEMENT SERVICES: MAKE MY DAY!

Because all three directories enable you to assign users and applications the rights to access network resources, it's reasonable to assume that these directories can also enforce these rights, and all three directories can. As you might expect, however, the way these directories enforce access controls can significantly affect the security of the resources being protected.

eDirectory's enforcement services check users' rights to resources each time users exercise those rights. For example, suppose a company manager has administrative rights to several applications: Financial 1, Financial 2, and Financial 3. When this user logs in to eDirectory and clicks to select Financial 1, eDirectory calculates this user's right to access Financial 1 from all of the ACLs and trustee assignments that apply to Financial 1. If this user then opens Financial 2, eDirectory computes this user's rights to access Financial 2, and so on.

Active Directory and Sun ONE approach enforcement quite differently. When users log in to these directories, Active Directory and Sun ONE read from all applicable ACLs and ACIs to determine all of the rights users have to directory-controlled resources. These directories then create an electronic ticket upon which they stamp these rights.

As long as a user retains his or her connection, these directories pass this ticket to the resources he or she requests. These resources then grant to this user the rights that are stamped on this ticket.

Admittedly, these enforcement services demand less of the directory server's processing power than eDirectory's enforcement services. However, eDirectory's enforcement services are much more effective.

For example, suppose a user who has administrative rights logs in to the directory and then is told that his or her position has been terminated. As the network administrator, you are notified of this event, and immediately upon notification, you disable this user's account.

If your company is using Active Directory or Sun ONE, this user retains administrative rights to your company's network until he or she closes his or her network connection—even though this user's account is disabled. If your company is using eDirectory, in contrast, the instant this user requests a resource that is controlled by the directory, eDirectory discovers that this user's account is no longer valid and terminates the user's connection.

STORAGE SERVICES: THE FOUNDATION UPON WHICH DIRECTORIES ARE BUILT

The FSD model also includes storage services, which rely on the databases that underlie directories. The database that underlies a directory can provide a number of important services, including services that provide persistence.

Persistence refers to a directory's ability to protect the data stored therein. The three directories discussed in this article practice the virtue of persistence in many ways. For example, all three directories use logs to ensure that intended changes are completed, even if the server experiences a failure while the directory is in the process of making these changes.

Sometimes Even Persistent Directories Need Help

However, a directory's underlying database can't always diagnose and repair data-related problems on its own. Even the strongest underlying databases sometimes need outside intervention to maintain persistent data. When you have to provide that outside help, having the right tools for the job can make a big difference.

Of the three directories discussed in this article, eDirectory has the most comprehensive and easiest-to-use set of tools for diagnosing problems with and repairing its database. You are probably familiar with tools such as DSTRACE and DSREPAIR, which you can use to locate and fix problems in eDirectory. If you have used these tools, you know that they work while eDirectory is running.

You also know that in the past, you had to access these tools through the eDirectory server console. You can now access and use these and other directory maintenance tools through iManager 1.5. You can also access DSTRACE via the version of Novell Remote Manager that is included in NetWare 6 Support Pack 1. Like iManager 1.5, Novell Remote Manager is a web-based management tool.

Whether you use iManager or Novell Remote Manager, you can now access DSTRACE and DSREPAIR from any computer that has a standard web browser and an Internet connection—provided, of course, that you have rights to do so. (For more information, see "Novell iManager: Keeping eDirectory Management Simple." For more information about the version of Novell Remote Manager included in NetWare 6 Support Pack 1, see "Beyond the Basics: New Features in the NetWare Remote Manager Utility as Found in NetWare 6 Support Pack 1," *Novell Appnotes*, June 2002. You can download this article from <http://developer.novell.com/research/appnotes/2002/a0206.htm>.)

To repair the Active Directory database, in contrast, you use a command-line tool, which you access by restarting the Active Directory server in a special directory services repair mode. In other words, to use this tool, you must bring down the server upon which Active Directory is running. This server remains unavailable on the network while you are using this

**Please visit our
advertiser Intermine at
www.intermine.com.**

tool to make repairs.

As Neuenschwander notes, this Active Directory repair tool is not only inconvenient to use but also presents a potential security risk. In directory services repair mode, Neuenschwander says, "there are no controls over what you can and cannot do with the directory; it's laid bare." Of course, a potential mal-doer would need physical access to the server to use this tool. Although Neuenschwander knows of no attacks on Active Directory through this repair tool, this potential security problem "should at least raise some eyebrows," he says.

Unfortunately, Sun ONE includes no repair tools to maintain the persistence of Sun ONE's underlying database. The only way to repair this database is to use Sun ONE's backup and restore services, which include batch files, UNIX shell scripts, and Perl scripts.

BACKUP SERVICES: RECOVERING FROM DISASTERS

eDirectory and Active Directory also include backup and restore services. eDirectory's new continuous backup services are available through iManager 1.5. Using these backup services, you can back up the eDirectory database while eDirectory is running (and available). If you need to restore eDirectory from backup, these services also enable you to restore the eDirectory database to its state just preceding the last synchronization. (For more information, see "Some Like It Hot: Backing Up Novell eDirectory 8.7," *Novell Connection*, July 2002, pp. 18-20. You can download this article from www.ncmag.com/past.)

Using Active Directory's backup and restore services, on the other hand, entails closing the Active Directory database.

PARTITIONING SERVICES: BREAKING UP THE DATABASE

When a directory's database becomes extremely large, you may want to break it up into manageable pieces—provided, of course, that these pieces can belong to the single, cohesive namespace that defines the directory tree. The ability to create partitions can improve a directory's scalability. After all, a directory can more easily find an object in a small database than in a large database—that is, if the directory knows in which small database to look. Maintaining a single namespace across all of a directory's partitions enables the directory to know which partition

holds the data it is seeking.

eDirectory enables you to define partitions at any point in a tree and maintains a single namespace within the tree. Regardless of the number of partitions you create within an eDirectory tree, you manage that tree as a single entity.

eDirectory also supports pruning, grafting, splitting, and renaming. These operations enable you to change the structure of your company's eDirectory tree as the structure of your company changes, ensuring that objects within the tree are logically related to the resources those objects represent. You can even merge two eDi-

If a directory can be partitioned but those partitions must reside on a single server, the directory probably can't scale to meet the demands of large, geographically distributed companies.

rectory trees to form a combined tree.

You can also create partitions in an Active Directory forest, but you can do so only at the domain level, which limits the extent to which you can partition an Active Directory forest. To provide a cohesive namespace across Active Directory domains (which define discrete management boundaries), Active Directory uses a Global Catalog that contains information about the objects in each domain. Each domain controller within an Active Directory forest must contain a complete copy of this Global Catalog. In other words, the Active Directory Global Catalog does not support partitioning.

Sun ONE supports partitioning at the container level and maintains a cohesive namespace for these partitions using index files. Neither Active Directory nor Sun ONE support pruning, grafting, splitting, renaming, or merging—limiting your ability to restructure your company's directory.

DISTRIBUTION AND INTEGRITY SERVICES: SPREAD IT AROUND

You should also consider how a directory database can be stored. If a directory can be partitioned but those partitions must reside on a single server, the directory probably can't scale to meet the demands of large, geographically distributed companies. If you can distribute partitions to multiple servers, you can place segments of your company's directory tree near the users who need to access those segments. Distributing partitions saves bandwidth, increases directory performance, and provides load-balancing.

eDirectory

Using eDirectory's distribution services, you can distribute partitions and replicas of partitions to any server on your company's eDirectory tree. (Replicas allow you to replicate the data within a partition.) In fact, eDirectory can support more than 100 partitions on a single server.

Note. The number of partitions eDirectory can support depends on the platform upon which eDirectory is running. For example, eDirectory can support significantly fewer partitions on Windows than it can on Solaris.

To conserve bandwidth between the servers upon which eDirectory is running, eDirectory supports both multimaster and master-slave replication. A master replica of an eDirectory partition is a replica to which you can write changes and from which you can create partitions. Because eDirectory supports multiple master replicas, you can update information in the master replica that is geographically closest to you. To keep data synchronized throughout your company's directory tree, this master replica then communicates these changes to other replicas, including slave replicas, which are read-only replicas.

To further conserve bandwidth, eDirectory uses delta and transitive replication to communicate changes between replicas. With delta replication, only the information that has changed is replicated. With transitive replication, a partition can replicate changes without having to contact each replica to communicate those changes. For example, suppose replicas of a partition are stored on three servers: Server A, Server B, and Server C. Further suppose that an attribute value in the replica on Server A has just been changed. Transitive replication enables Server A to replicate this change only to

Server B, as opposed to replicating this change to Servers B and C. After Server A replicates its change to Server B, Server B replicates this change to Server C.

Transitive replication can accelerate the replication process. Using transitive replication, a downed server in a replica ring can't hold up the replication process, as it can without transitive replication.

Active Directory

Like eDirectory, Active Directory enables you to replicate and distribute partitions on one or more servers. However, Active Directory supports only one partition—or domain—per server, which can make distributing Active Directory partitions a hardware-intensive (and therefore expensive) proposition.

Active Directory also supports multi-master, master-slave, transitive, and delta replication. However, whereas eDirectory replicates changes at the value level of an attribute, Active Directory replicates changes only at the attribute level. For example, suppose the value of user Jane's telephone attribute is changed from 801-843-2000 to 801-443-2000. In this case, eDirectory would replicate only the changed number—the number four in the first digit of the prefix. Active Directory would replicate the entire telephone number. Therefore, replication in Active Directory consumes more bandwidth than does replication in eDirectory.

Sun ONE

Although you can partition a Sun ONE directory tree, distributing individual partitions among multiple servers creates significant management overhead, particularly with master replicas of partitions. Although Sun ONE supports multi-master replication, providing two-way communication between master replicas requires manual configuration. You must configure each master replica to act as both a consumer and a supplier to another master replica. (A *consumer replica* is a replica that accepts changes from another replica, and a *supplier replica* is a replica that communicates these changes.) To enable communications to flow between two Sun ONE master replicas, you must perform four configurations.

Because configuring more than two master replicas is complicated and confusing, Sun guarantees multimaster support for only two master replicas. This limitation ultimately affects the scalability of

Sun ONE.

To conserve bandwidth, Sun ONE supports cascading master-slave and delta replication at the attribute level. Using cascading master-slave replication, read-only replicas that are configured as consumers can also act as supplier replicas for other read-only replicas. Of course, you must manually configure cascading master-slave replication.

RELATIONSHIP SERVICES: PROVIDING THE TIE THAT BINDS

In the FSD model, relationship services help you create and manage relationships between the objects within a directory tree and between two or more directory trees. For example, when you create a traditional Group object in eDirectory, Active Directory, or Sun ONE, you are creating a relationship between the collection of User objects listed within that Group object and the Group object itself. Such relationships can simplify the task of managing objects.

Dynamic Groups

In traditional Group objects, membership is static, which means membership doesn't change unless you manually add or delete members. However, eDirectory and Sun ONE also enable you to create Dynamic Group objects and Role objects, respectively. Active Directory does not support these types of objects.

Unlike traditional Group objects, which are based on static membership lists, Dynamic Group and Role objects include search filters that enable you to base membership on certain criteria, such as the location in which users work. When you access Dynamic Group objects in eDirectory or Role objects in Sun ONE, these directories calculate group membership on-the-fly based on these criteria. Because you don't need to manually add and delete individual members, dynamic groups are easier to manage than static groups.

You can manually add users who do not meet the criteria that define a particular Dynamic Group or Role object. Similarly, you can exclude specific users who do meet these criteria. (For more information about managing dynamic groups in eDirectory, see the "Get a Group" sidebar on p. 10 of "Novell iManager: Keeping eDirectory Management Simple.")

Relationships With Class

Sun ONE also includes another rela-

tionship service that Active Directory and eDirectory do not offer. Sun ONE's Class of Service enables you to refer an attribute of one object to an attribute of another object. For example, suppose your company's directory tree includes a fax number attribute within a departmental container. The value of this attribute is, of course, the number of the department's fax machine.

You can refer this container object's fax number attribute to the fax number attribute of each User object included in this container object, and vice versa. After you create these references, you no longer need to update the departmental fax number in these individual User objects. If this department's fax number changes, you change the fax number only once in the departmental container object. In this way, Class of Service simplifies the task of managing objects within the directory.

Now You're Role-ing!

eDirectory also supports a relationship service that neither Active Directory nor Sun ONE support: Role-based administration simplifies the task of managing eDi-

**Please visit our
advertiser Beginfinite at
www.beginfinite.com.**

Let Me Count the Protocols

The greater the number of access protocols a directory supports, the greater the number of applications that can use that directory for access control and the storage of user identity information. The following table lists the access protocols Novell eDirectory, Active Directory, and Sun ONE support. (In the Full Service Directory [FSD] model, these access protocols are called *discovery methods*. For more information, see "A Model Directory" on p. 10.)

Directory Server Applications	Database	Other Applications Browsers	Internet
Novell eDirectory (JDBC) and Open Database	Java Database Connectivity (ADSI), an application program Connectivity (ODBC)	Active Directory Services Interface JNDI, HTTP interface for Active Directory that supports several programming languages, including ActiveX, C, C++, Visual Basic, and scripting languages; eXtensible Markup Language (XML); Java Naming and Directory Interface (JNDI); Lightweight Directory Access Protocol (LDAP); Novell Directory Access Protocol (NDAP), an API for eDirectory that supports several programming languages, including ActiveX, C, C++, and Visual Basic.	LDAP, XML,
Microsoft Active Directory		ADSI (Through ADSI, Active Directory supports LDAP, NetWare APIs, NT, and Windows OS applications.)	LDAP (through ADSI)
Sun ONE Directory Server		LDAP	LDAP

rectory for you and for the users to whom you delegate specific administrative tasks.

To implement role-based administration, you create Role-based container objects. These container objects represent organizational and functional roles within your company and include management tasks that you want to assign to these roles. Role-based container objects also contain references to User objects, which represent the particular users to whom you want to assign these tasks.

To perform these tasks, users access iManager 1.5, which reads from the directory the specific tasks that these users are authorized to perform and displays only these tasks. Users do not have to wade through scores of tasks in ConsoleOne to find the specific task these users are authorized to perform. (For more information, see "Novell iManager: Keeping eDirectory Management Simple.")

CROSS-PLATFORM SERVICES: THINKING OUTSIDE THE MODEL

Although cross-platform support isn't a

criterion for directory services in the FSD model, it is an important consideration. After all, if your company's network is like most, it includes a variety of server operating systems. Therefore, your company probably needs a directory service that can run on multiple operating systems.

How do eDirectory, Active Directory, and Sun ONE compare on cross-platform support?

- eDirectory runs on NetWare, IBM AIX, Linux, Solaris, and Windows server operating systems.
- Active Directory supports the Windows NT operating system in only a limited fashion and fully supports only Windows 2000.
- Sun ONE runs on HP-UX, IBM AIX, Tru64 UNIX (available only through Compaq) and Windows.

CONCLUSION

Despite the observation that directories are increasingly becoming application independent, and vice versa, the truth is

this: The relationship between directories and applications is a symbiotic one and will probably remain so. However, this doesn't suggest that forward-thinking companies will continue to select applications first and directories second.

On the contrary, the new relationship between applications and directories suggests that selecting the right directory is at least as important as selecting the right application. In fact, a poor directory choice today may limit the number of applications from which your company can choose tomorrow. At the very least, a poor choice today will mean adding to the number of directories already running on your company's network tomorrow.

If your company recognizes the need to evaluate and select a directory as carefully as it does the applications that use this directory, evaluating your company's needs and the FSD model can help you make that selection. Although this article cannot address your company's present needs, it does demonstrate that of the three popular directories available today,