

# Too Many Directories?

## Synch 'Em With DirXML

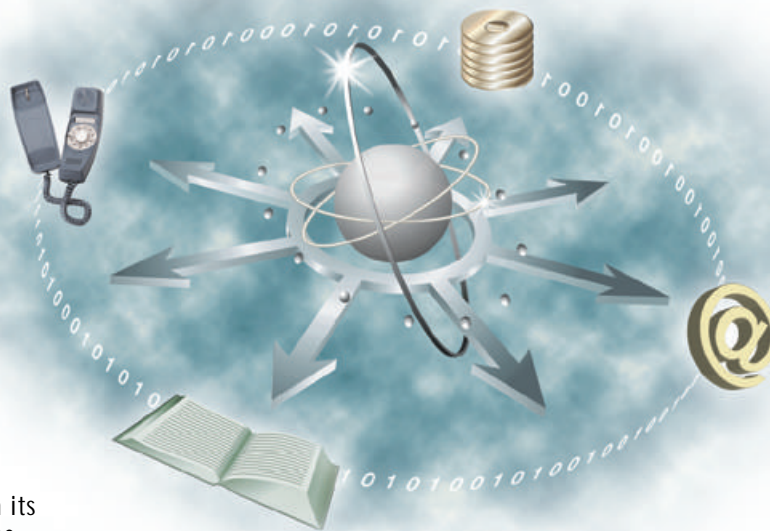
Linda Kennard

**H**ow many directories does your company have on its network? Two? 20? 200? If your company is like the "average company," Forrester Research estimates that it has approximately 180 separate directories.

Of course, the exact number of directories your company has depends on the number of operating systems, databases, and applications your company is running because each system has its own directory. Consequently, your company ends up with separate directories for its enterprise resource planning (ERP) database, e-mail application, white-pages application, telephone system, Microsoft NT segments, and NetWare environments—to name only a few of the systems with directories commonly found in today's enterprise networks.

As a network administrator, you know only too well that managing multiple directories creates irritating—not to mention costly—inefficiencies that increase with the number of directories. For example, suppose your company has 20 directories that contain much of the same data for each of 500 employees. As is the nature of business (and life), stuff happens: People get hired, fired, transferred, promoted, or just get a new telephone number. Changes occur all the time, and when one of those changes affects user-profile data, you and your coworkers in other departments must enter and re-enter much of the same information in 20 different directories. If your company has 200 directories, you might enter and re-enter the same change 200 times—what a waste of time and money.

In addition to the obvious inefficiencies of maintaining multiple copies of the same information, disparate directories should concern you (and upper management) for another reason. Assuming your company is like most, you either have or are beginning to develop an e-business strategy, which means you're thinking about how you can use Internet technologies to improve business processes. As Paul Corriveau, Novell product marketing manager, rightly points out, however, "before you can develop an effective e-business strategy, the first thing you have to do is clean house." What Corriveau means is that an effective e-business strategy begins with the integration of these islands of information. After all, providing partners, suppliers, or customers with Internet access to di-



rectory information is a pointless endeavor if that information isn't current.

So how can you clean house now to prepare for future e-quests? How can you minimize (or even eliminate) redundant data entries in multiple directories, whether your company has two, 20, or 200 directories? Novell's answer to this question is by using DirXML, a metadirectory tool that is due to be released later this summer.

### THE WAY A METADIRECTORY SHOULD BE

DirXML (pronounced Der-X-M-L) is a NetWare Loadable Module (NLM) that you install on any server running NDS eDirectory 8.5, which will also be released later this summer. NDS eDirectory, as you probably already know, is the cross-platform version of NDS. As you would expect, DirXML runs on the same platforms NDS eDirectory runs on, including NetWare 5.1 and NetWare 5, Microsoft Windows NT 4.0, Microsoft Windows 2000, Sun Solaris platforms, and Linux.

Platform independence is cool, of course, but certainly isn't the point of DirXML. The real point, or rather "the whole point," of using DirXML, explains Corriveau, "is to integrate the multiple directories on your network." That is, with DirXML, you create a metadirectory like you imagine one should be (not in the nerve-wracking sense the term *metadirectory* might evoke).

The metadirectory you create using DirXML is a figurative directory wheel for which NDS eDirectory (hereafter referred to as NDS) is the central hub. DirXML serves as the figurative spokes that connect each of the separate directories to this hub. (See Figure 1 on p. 10.)

Any other directory for which DirXML provides a driver or for which you (or another third party) write a driver can participate in the metadirectory you create using DirXML. In its initial release, DirXML will ship with a software developer's kit

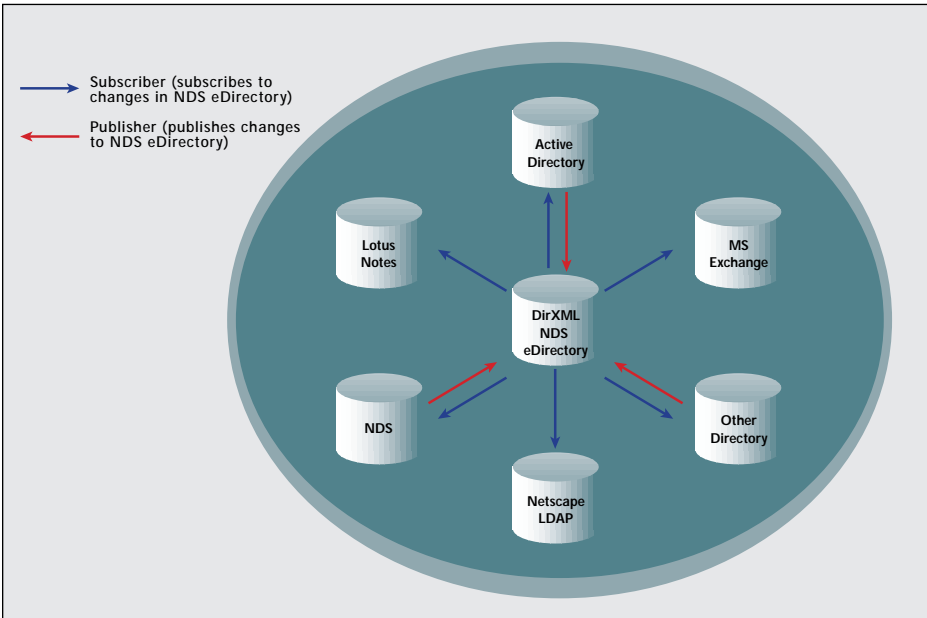


Figure 1. Using DirXML, you can create a figurative directory wheel. NDS eDirectory acts as the wheel's hub from which other directories stem. Depending on how you configure DirXML, it will replicate changes you make in NDS eDirectory to other directories in the wheel and replicate changes you make in other directories to NDS eDirectory.

(SDK) for writing drivers (using C++ or Java). In addition, DirXML will provide drivers for NDS 8 and higher as well as directories from the following applications:

- Lotus Notes
- Microsoft Exchange
- Microsoft Active Directory
- Netscape LDAP

A driver (also called an *Application Shim*) is essentially the key to the gate that DirXML opens between NDS and other directories. As Nick Nikols, DirXML development manager and architect, explains, a driver in a DirXML context is not unlike the drivers you are familiar with in a network context. That is, when you want to connect a hardware device to a network, you first need to install the appropriate driver on the server. This driver, as you well know, enables the network to communicate with the new hardware device and vice versa. Similarly, when you want to connect a directory to the DirXML-powered directory wheel, you must first install (and configure) an application-specific driver.

**SUBSCRIBERS AND PUBLISHERS: COMING AND GOING**

Conceptually, all information shared between NDS and other directories is either coming from or going to NDS, depending upon whether the connected directory subscribes to NDS changes or publishes its changes to NDS, respectively, or both. For each driver, you configure a subscriber channel, a publisher channel, or both using the DirXML snap-in module for ConsoleOne. (See Figure 2.)

If a directory subscribes to NDS changes and you update certain NDS object classes and attributes, DirXML pushes the updates to this directory. These updates flow over the subscriber channel that you configure for this particular directory. (See Figure 1.) Between NDS and this other directory, DirXML synchronizes only the information that you configure DirXML to synchronize.

Similarly, if a directory publishes its changes to NDS and you update certain object classes and attributes in this directory, DirXML pushes the resulting changes to NDS. These changes flow over the publisher channel that you configure for this particular directory. (See Figure 1.) Again, DirXML pushes through only the changes that you configure it to push through. (For more information, see the related article, "Check Out That DirXML Engine" on p. 16.)

**SYNCHING IS A SNAP**

With a DirXML-powered directory wheel, updating information in multiple directories is simple. For example, sup-

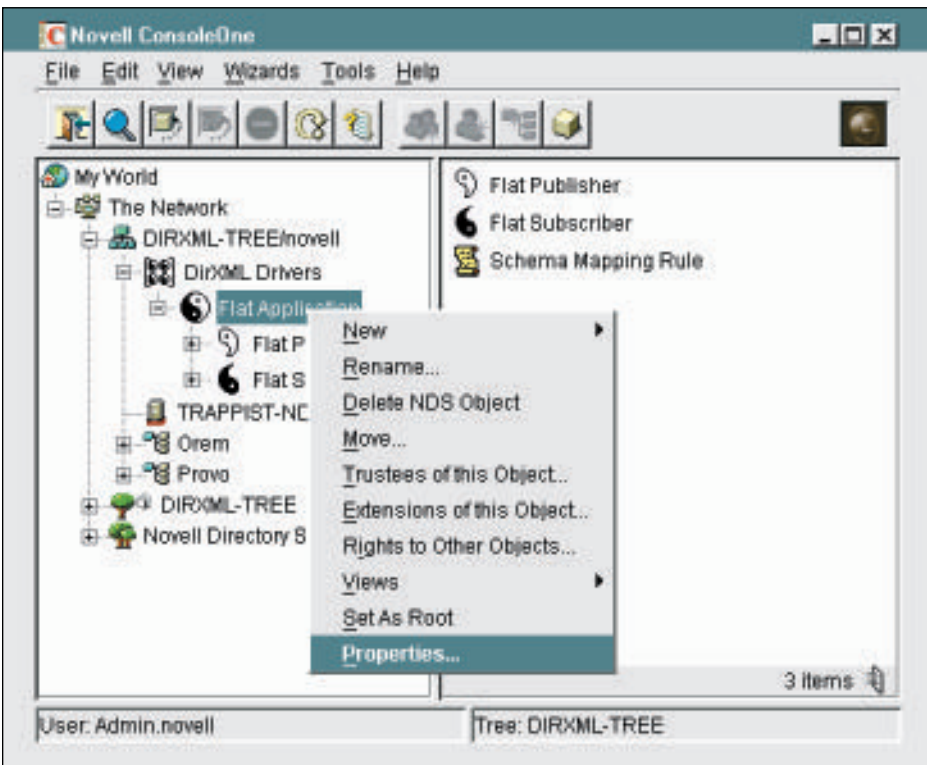


Figure 2. Using a snap-in module for ConsoleOne, you can use DirXML to configure directories to subscribe to NDS changes. If a directory subscribes to NDS changes and you update certain NDS object classes and attributes, DirXML pushes the updates to this directory.

pose that your company runs Lotus Notes, Netscape SuiteSpot, PeopleSoft, and NDS. Further suppose that each user on your company's network has an account in each of these directories and that these accounts contain roughly the same information. "Having all of that user information in all of those directories," Corriveau says, "is a nightmare to manage."

You probably know from experience precisely what Corriveau means. When you make a change to a user account in Lotus Notes, you have to make the same change in Netscape, PeopleSoft, and NDS. This process is inefficient and makes it difficult, if not impossible, to maintain current information in all of these directories all of the time.

In other words, without DirXML, updating information in four directories is a four-step process:

1. Update the Lotus Notes directory.
2. Update NDS.
3. Update the Netscape directory.
4. Update PeopleSoft.

With DirXML, in sharp contrast, updating four directories (or 200) is a one-step process:

1. Update the Lotus Notes directory.

With that one step, you're done. DirXML does the rest.

How does DirXML do it? The answer to that question is quite lengthy and, to many people, boring. In fact, to many of you, reading about bridging directories is a bit like reading about bridging teeth: The results of the process are dazzling, but reading about the process can be dull. However, if you're interested in learning more about how DirXML works, see the related article, "Check Out That DirXML Engine" on page 16.

#### **A PENNY SAVED IS A PENNY EARNED**

Admittedly, you won't get this directory wheel rolling without effort. After all, although DirXML is a relatively easy metadirectory tool, synchronizing directories is never truly easy. In fact,

Richard Reid, an IT manager for True North Communications, describes directory synchronization as "one of the worst tasks in the world," an opinion you may share.

*If a directory subscribes to NDS changes and you update certain NDS object classes and attributes, DirXML pushes the updates to this directory.*

Nevertheless, even Reid, who was one of a handful of customers who tested the private beta of DirXML, had to admit that DirXML—even in beta—helped ease the struggle of synchronizing directories. (For more information about Reid's plans for DirXML, see "True North Communications' Dir(XML)ection" on p. 12.)

Please visit our advertiser

**Allegro**

<http://www.allegro.com>

## True North Communications' Dir(XML)ection

When Richard Reid, manager of True North Communications' Worldwide Messaging and Directory Team, received his copy of Novell's NDS eDirectory and DirXML private beta near the close of the last century, he already had big plans. True North Communications is a leading global communications holding company with more than 300 operations in 83 countries and comprising advertising agencies, such as FCB Worldwide, Bozell Group, and Temerlin McClain. (For more information about True North Communications, visit their web site at <http://www.truenorth.com>.)

Not surprisingly, True North Communications and the companies that fall under what Reid calls the True North Communications' umbrella have hundreds of directories in dozens of locations—all connected by a private WAN. True North Communications alone has a wide range of databases, including ODBC and SQL databases, and application directories, such as Netscape SuiteSpot and PeopleSoft directories. In addition, several of the advertising agencies under the True North Communications' umbrella run their own e-mail systems on GroupWise, Netscape SuiteSpot, or Microsoft Exchange. Furthermore, "depending on how you define directory," says Reid, "we also have printed address books." Basically, Reid concludes, "We have directories everywhere."

Eventually, Reid would like to synchronize all of his company's major directories and hopes to do so using DirXML. Reid started this endeavor by testing the DirXML beta—using it to synchronize user-profile data between NDS and his Netscape SuiteSpot directory. With approximately 15,000 user objects to synchronize, the job was not a small one but, Reid felt, was

necessary. "Netscape's directory hasn't been able to replicate the way we'd like it to," Reid explains. As a result, Reid and his team ultimately plan to convert from Netscape to NDS eDirectory. Synchronization is the first step in this conversion. If all goes well in the test environment, Reid plans to go live with the synchronization—even before DirXML ships. (In fact, he has probably already done so.)

After Novell ships DirXML, Reid plans to use DirXML and its Exchange driver to synchronize affiliated agencies' Exchange systems with NDS eDirectory. Each affiliate office will maintain its own directory. Reid and his team will simply "maintain synchronization between [the affiliates'] e-mail systems and NDS [eDirectory]."

Reid also plans to use DirXML to complete True North's move from paper-based white pages to electronic white pages. For years, True North Communications' worldwide offices and affiliates have been printing and exchanging telephone directories. As is the nature of printed information, these directories were not only time-consuming and costly to prepare and print, but the information in them was quickly outdated.

To eliminate these paper-based white pages, Reid will use DirXML to enable each office to publish local information to True North's global NDS tree. Each office will publish employees' names, telephone numbers, and titles, as well as information about its advertising and marketing services. Through DirXML-powered synchronization with NDS eDirectory, this local information will become globally available. Using web-based applications that Reid and his team developed, users at True North offices will be able to access current information about employees and services in their own or other True North offices at anytime and from anywhere. ●

Is the struggle really worth it? Reid certainly thinks so. Last February, Reid and the Worldwide Messaging and Directory Team that he manages tested the beta DirXML by synchronizing a Netscape Lightweight Directory Access Protocol (LDAP) directory with their NDS tree. Because the need for directory synchronization was and remains pressing, Reid plans to finish testing and go live with the Netscape-to-NDS synchronization—even before Novell ships DirXML. Directory synchronization, according to Reid, is simply a requirement.

Why is synchronization a requirement? The answer isn't hard to guess when you imagine the potential benefits of a figurative directory wheel. The most immediate benefit, of course, has to do with the time and money your company can save by using DirXML to connect disparate directories. Think about it. Suppose you and your coworkers spend only 10 minutes adding, deleting, or changing a user account in

each of the 100 directories on your company's network. Further suppose that your company has 1,000 employees.

If your organization experiences a 20 percent turnover annually (including hires, fires, and transfers), you will need

*The most immediate benefit [of DirXML], of course, has to do with the time and money your company can save by using DirXML to connect disparate directories.*

to add, delete, or change new user accounts in those 100 directories at least 200 times each year. At 10 minutes per change, collectively, you and your coworkers will spend 200,000 minutes or

just over 3,333 hours per year adding, deleting, or changing user accounts. In contrast, with DirXML, you can make each of those 200 changes only once to update all 100 directories. With DirXML, you will spend just over 33 hours per year updating your company's directories.

Now consider how saving time translates into saving money. Naturally, we have no way of knowing exactly what you and your coworkers are paid since wages vary from company to company, department to department, and title to title. However, for the sake of this example, suppose you and your coworkers are paid an average of U.S. \$30 per hour.

At this rate, your company pays you 50 cents per minute. Ten minutes per directory change at 50 cents per minute means that your organization spends nearly U.S. \$100,000 every year paying you and your coworkers to enter and re-enter the same information in 100 directories 200 times each year.

With DirXML, on the other hand, you enter only one change one time to

## Filtered Replicas

When you deploy DirXML, you have one or more DirXML drivers running on a particular server. For example, on a server running Lotus Notes, you may run the DirXML Lotus Notes driver and configure its publisher and subscriber channels to synchronize user-profile data between the Lotus Notes directory and NDS eDirectory.

To update NDS eDirectory, this Lotus Notes server needs replicas of NDS partitions that include user-profile information. If this user-profile data is stored in multiple partitions, such as partitions A, B, C, and D, you would need a replica of all of these partitions on the server. What are your options?

Currently, you can place full replicas of partitions A, B, C, and D on the server or write a distributed application that could walk the NDS tree to find the user-profile information. However, as Nick Nikols, DirXML development manager and architect, points out, placing full replicas of four partitions on the server running Lotus Notes is not “the optimal solution for replication.” And a distributed application that walks the NDS tree to find information is prohibitively slow, not to mention the fact that writing a distributed application isn’t easy. “So you’re left in a quandary. What,” asks Nikols, “are you going to do?”

When Novell releases the next version of NDS eDirectory this summer, you will be able to use a new replica type called the filtered replica. With filtered replicas, you can configure a filter that indicates what objects and attributes you want from a particular partition.

For example, in the case of the server running Lotus Notes

and requiring user-profile information from partitions A, B, C, and D, you can set up a filter that indicates that you want replicas of these partitions on this server to include only User objects and a specific subset of their attributes (such as first and last names and telephone numbers). You can then apply that filter to all four partitions.

The Lotus Notes server will then receive filtered replicas of partitions A, B, C, and D. These filtered replicas will preserve the directory structure but, because of the filter, the container objects within that structure will hold only User objects, their first and last names and telephone numbers.

Placing a filtered replica on a server will not prevent you from also storing full replicas on that server. For example, a server may hold a full read-write replica of partition A and read-write or read-only filtered replicas of partitions B, C, and D.

The benefits of filtered replicas are fairly obvious. For example, servers that use filtered replicas won’t require as much storage space. In addition, by using filtered replicas, you will reduce replication traffic and thus conserve bandwidth—assuming that all of the other servers are running the upcoming version of NDS eDirectory and, therefore, are also aware of the new filtered replica type.

If your company’s network includes a mix of NDS versions, servers running older versions of NDS will replicate all information from partitions A, B, C, and D to the server that requests filtered replicas of these partitions. However, the server requesting filtered replicas will filter out replication packets locally. In this case, you won’t conserve bandwidth, but you will still conserve storage space on the server for which you’ve set up filtered replicas. ●

update all 100 directories. As a result, your company spends just under U.S. \$1,000 per year to ensure that all of its directories remain up-to-date and, therefore, useful.

### CLEANING HOUSE FOR FUTURE E-GUESTS

Although the immediate benefits of using DirXML—saving the time, money, and hassle associated with entering and re-entering the same data multiple times—are evident, other less apparent benefits are just as real. For example, by using DirXML to create a metadirectory, you can ensure that NDS-enabled applications that require a global view of the NDS tree are accessing the most current information. Furthermore, by using a new replica type called the filtered replica, you can replicate specific attributes of object classes throughout the entire NDS tree without having to place full replicas of every partition on one server. (The filtered replica is a feature of NDS eDirectory 8.5. For more information about this feature, see “Filtered Replicas.”)

In addition, by ensuring that the information in your company’s directories is current, DirXML opens the doors to e-business opportunities. After all, as Nikols points out, to be worthwhile, e-business applications need access to current, consistent information.

For example, you can make information in any application accessible over the Internet, including applications that are not typically Internet-ready, such as ERP, custom, and e-mail applications. By configuring a DirXML publisher channel between these applications and NDS, you can add different object classes and attributes to NDS. By doing so, you enable DirXML to synchronize these object classes and attributes between their host applications and NDS. Once the information from nonInternet-ready applications is stored in NDS as well, remote users can use an LDAP-enabled application to access this information in NDS.

You can also use DirXML to synchronize directories at different sites over the Internet, including directories maintained by your company’s business partners, suppliers, or even customers. For example,

suppose your company has business partners who manage health care, dental, or 401K plans. Assuming your company makes the appropriate agreements with these partners and assuming these partners provide the interface to their application directory, you could create a DirXML driver to push any changes that are made to your company’s NDS User object data to your partners’ directory.

Using DirXML to synchronize your NDS directory and your partners’ application directories would benefit both your company and its partners. For example, information in both directories would always be current without requiring a great deal of effort from your company or its partners. Furthermore, your partners would not have to relinquish control over their directory to synchronize portions of it with yours. In other words, you can use DirXML to get your “house in order” today, says Nikols, to prepare for tomorrow’s e-business guests.

*Linda Kennard works for Niche Associates, an agency that specializes in technical writing and editing. Niche Associates is located in Sandy, Utah. ●*