Over the past decade, Novell Directory Services (NDS) has evolved from a common database of network resources to NDS eDirectory, which extends directory services across the enterprise and to the Internet. Today, Novell’s vision of connecting the world with NDS has become a reality. NDS eDirectory allows companies to connect multiple systems—both inside and outside company firewalls.

To enable companies to integrate multiple, separate directories with NDS eDirectory 8.5, Novell provides two integration technologies—federation, which is a feature of NDS eDirectory 8.5, and DirXML, which is a separate product that works with NDS eDirectory 8.5. Although both eDirectory federation and DirXML enable companies to integrate separate directories with NDS eDirectory, each technology provides distinct advantages and uses. This article examines these advantages and uses.

(You can also merge two eDirectory trees to make a single tree with only one schema. For more information, see “What About the DSMERGE Utility?” on p. 29.)

EDIRECTORY FEDERATION

Today, more than ever, companies are leveraging directories to manage digital identities for intranets, extranets, web sites, e-mail systems, and e-commerce. Many of these directories contain identities for the same individuals. Because these identities are stored in disparate directories, however, business partners must duplicate management tasks to provide common customers and employees with access to web sites and extranets.

For example, suppose ABC Corp. and XYZ Corp. are partners. If employees of ABC Corp. need to access information on XYZ Corp.’s web site, XYZ Corp. must create new identities in its web directory for each ABC Corp. employee. Each time a new employee joins ABC Corp. or each time an employee quits or is fired, ABC Corp. must contact XYZ Corp. XYZ Corp. must then manually make the necessary changes to its web directory.

Maintaining duplicate identities for the same individuals in disparate directories leads to excessive management costs and the possibility of inconsistent data. Users are also affected: Because users have multiple identities, they must keep track of multiple login names and passwords and authenticate to each web site separately. Having multiple identities makes accessing information more difficult.

eDirectory federation addresses these problems by allowing business partners, common service providers, and e-tailers to link their directories and share common identity information. With eDirectory federation, each company maintains complete administrative control over its own eDirectory tree. To provide partners and common customers with access to web sites and other resources, each company simply grants eDirectory rights to users, groups, or containers located in other federated trees.

By federating their eDirectory trees, partner companies can seamlessly access the information they need on each other’s networks without having to duplicate users in each tree. Federated eDirectory trees also simplify the user experience: Users can access the web sites of various business partners with a single identity.

For example, suppose a real estate company and a mortgage company federate their eDirectory trees to allow common customers to access each company’s web site. If a home buyer logs on to the real estate company’s web site, he or she can access the mortgage company’s web site without having to create a separate web identity.

How eDirectory Federation Works

eDirectory 8.5 uses Domain Naming System (DNS) to connect and resolve objects between two or more DNS-rooted eDirectory trees. When you install a new eDirectory 8.5 tree, you have the option to create a traditional (non-DNS-rooted) eDirectory tree or a DNS-rooted tree. A DNS-rooted tree uses a DNS domain to locate the [Root] of other trees. eDirectory can then locate objects in other eDirectory trees on the Internet.

To federate DNS-rooted trees, each company simply creates two DNS “A” records for each domain. These records define how the other DNS-rooted trees on the Internet can locate a server that hosts the partition root for their eDirectory tree.

In a DNS-rooted tree, eDirectory uses standard eDirectory protocols to resolve some parts of the tree and relies on DNS to resolve other parts of the tree. For example, suppose that...
Novell.com and NCMag.com have federated their eDirectory trees. If user Debi.Editors.NCMag.com attempts to access a private technical publications area of Novell.com, a web server at Novell.com checks to see if Debi.Editors.NCMag.com is a member of the eDirectory group Writers.Techpubs.Novell.com. This process is outlined below. (See Figure 1 on p. 28.)

2. DNS resolves the name and hands back the IP address of a server that hosts NCMag.com.
4. eDirectory then resolves the rest of the name using eDirectory resolution protocols.

The entire eDirectory federation process is fairly simple. Adding a federated partner is simply a matter of adding the partners' DNS name to the eDirectory partner list and granting the partners' users, groups, or container objects the appropriate eDirectory rights.

Removing a partner is also easy: You simply remove the partner from the eDirectory partner list or revoke the necessary eDirectory rights.

Benefits of eDirectory Federation

Federated eDirectory trees provide a number of benefits to companies and users. The key benefit is that common service providers do not need to maintain duplicate identities in multiple directories for the same users. Because each user has only one identity across multiple federated trees, companies can easily grant or revoke access rights to that user.

For example, a company can grant a user access to resources by making that user a member of an eDirectory group. To revoke that user's access rights, the company simply removes the user from the group. If a company deletes a user, this action revokes the user's access to all other federated trees.

With eDirectory federation, users have to remember only one username and one password for multiple federated trees. A user's identity travels with the user to any federated web site, and that one identity provides users with access to a variety of web services.

DIRXML

eDirectory federation provides a bridge between two separate directories, allowing you to grant common users rights to access both directories. DIRXML, on the other hand, synchronizes data between security domains (that is, directories or databases), allowing the data to be managed according to the security principles of the hosting domain.

The average company today maintains between 20 and 180 different directories, e-mail databases, employee address books, network databases, Human Resources databases, and special-purpose application...
databases. As you know only too well, keeping common information in these directories up-to-date and synchronized is both costly and resource intensive. When information in one database changes, companies use a variety of methods, including manual updates and synchronization, to update the other databases with the new information. Despite their best efforts, however, maintaining consistent information across these separate data stores is nearly impossible. Consequently, a company's databases often contain outdated and incorrect information.

Of course, this problem is compounded if a company is sharing information with partners. For example, if a customer updates his or her information on one company's web site, how does that company notify a partner that the customer's information has changed? If the process is not automated, it could take weeks or even months to update the partner's databases or directories.

**How DirXML Works**

DirXML combines eDirectory with eXtensible Markup Language (XML). (XML is a standard for data interpretation and transformation.) By combining these two technologies, DirXML allows eDirectory to replicate data to and from other directory or database name spaces, such as iPlanet's Directory Server or Microsoft's Active Directory.

DirXML allows you to replicate two or more dissimilar directories or databases as one system with one set of management tools. These directories and databases are not impacted by this replication. In fact, the directories and databases may not even know they are sharing data with a larger system.

DirXML connects all of the application-specific and enterprise directories on a network to eDirectory. When a change in common information occurs, that information is sent to eDirectory. Then, based on rules and other criteria that you specify, eDirectory propagates the change to any other application, directory, or database that requires the information. (For more information about DirXML, see “Too Many Directories? Synch ‘Em With DirXML” and “Check Out That DirXML Engine,” Novell Connection, May 2000. You can download these articles from www.ncmag.com/past.)

For example, suppose that the Human Resources department changes the personal information of an employee in the company's PeopleSoft database. DirXML and eDirectory propagate this change in the following way. (See Figure 2.)

1. When the change is detected, a built-in event system notifies the DirXML engine that a change has occurred.
2. Based on a set of rules stored in eDirectory, the DirXML engine determines which databases should receive the change.
3. After determining which databases should receive the change, the DirXML engine uses eXtensible Stylesheet Language (XSL) to transform the information to match the requirements for the receiving systems.
4. Using protocols understood by both sides, DirXML then sends this transformed data.

Novell has developed a number of drivers that enable the exchange of information between eDirectory and other popular databases. DirXML includes seven drivers:

- NDS eDirectory
- Active Directory
- Lotus Notes
- Exchange
- Lightweight Directory Access Protocol (LDAP)-compliant directories, which can be used with iPlanet, SecureWay, Innosoft, and Critical Path
- NT Domain
- Delimited text, which can be used with flat file databases
Other drivers are available, including a driver for PeopleSoft and a driver for Java Database Connectivity (JDBC). (For more information about DirXML drivers, visit www.novell.com/products/nds/dirxml/drivers.) In addition, Novell has a software developer's kit (SDK) specifically for developing new drivers. (For more information, visit www.novell.com/partners/dirxml.)

Benefits of DirXML

By using XML to replicate and synchronize data, DirXML provides a system that allows dissimilar systems to share data and events. In this way, you can seamlessly integrate dissimilar databases and directories and significantly reduce the costs associated with managing these databases.

DirXML also simplifies sharing data between applications. DirXML brings the power of eDirectory to applications without requiring a major coding effort. DirXML applies the policies and transformations that allow data to flow easily between applications.

In addition, DirXML reduces the political issues of data ownership. Because DirXML does not enforce a hierarchy or a common naming scheme, departments within companies can maintain databases with different schemas and even different data representations. These departments can also maintain ownership of their data and define which data are replicated to eDirectory and who can access that data. As a result, DirXML is an ideal solution for business-to-business environments in which business partners, customers, and suppliers want to integrate dissimilar databases.

CONCLUSION

eDirectory federation and DirXML have entirely distinct and different purposes. eDirectory federation allows companies to link their eDirectory trees, enabling business partners, common service providers, and e-tailers to seamlessly access the information they need on each other's networks without having to duplicate users accounts.

No data is synchronized between federated trees. Objects in one federated tree can have relationships with objects in other federated trees. For example, users in one federated tree can be members of groups in other trees.

With eDirectory federation, each company has complete administrative control and autonomy over its own tree. The network administrator of each tree must grant access privileges to users. At any time, a company can discontinue a federated relationship with another company.

DirXML, on the other hand, synchronizes data between different directory or database name spaces. For example, DirXML synchronizes data with eDirectory, a database, an LDAP-accessible directory, or an e-mail address book, such as Microsoft Exchange. Data is replicated to and from eDirectory to the external name space. With DirXML, schemas and data representations are entirely separate.

Sandy Stevens-Marymee is a technical writer for Technology Innovations Group, Inc. She lives in San Diego, California.

What About the DSMERGE Utility?

Novell provides another way to integrate two NDS eDirectory trees—the DSMERGE utility. This utility allows you to merge two eDirectory trees with separate schemas as one tree with only one schema. When you combine these two eDirectory trees, the DSMERGE utility removes redundant objects such as the tree [Root] Certificate Authority.

At first glance, you may not understand the significant differences between the DSMERGE utility and the federation feature of eDirectory 8.5:

• You use the DSMERGE utility to combine two eDirectory trees into a single tree. With eDirectory federation, however, each federated tree remains intact and independent of the other eDirectory trees.

• When you use the DSMERGE utility to combine two eDirectory trees, the schemas of the two trees must match. Only one schema exists after a DSMERGE operation. With eDirectory federation, on the other hand, each eDirectory tree has its own schema, and the NDS schemas do not need to match.

For more information, visit www.ncmag.com/advertise.html.