Getting Active Directory Up And Running

Full commitment to Windows 2000 or XP means converting to Active Directory. Few organisations have gone all the way, but the numbers are growing.

By David Norfolk

Everybody now accepts Windows 2000 and XP as the future of Windows, but real acceptance of Active Directory, the Directory Service at its core, is far less common. You have two choices: to deploy Windows 2000 as a more scalable, and more manageable evolution of Windows NT and treat Active Directory simply as an evolution of the Registry and SAM; or, to exploit the Directory Service fully. Any configuration file or user-list can be considered as a directory but a full-function Directory is a specialised database with a set of functions similar to (but not necessarily the same as) those defined in the international X.500 standard (some directories, including Microsoft's Active Directory, are not X.500 compliant) and is intended to support multi-application, distributed, enterprise-scale computer systems.

Definition

So what exactly is a full-function Directory Service? It is a database which provides operational access to the metadata that describes a practical, distributed, computer system implementation. It describes the objects in the system, their relationships, security rules, locations and so on, and this information is used in real time, during operation of the system.

It is, in essence, an extended configuration file, which is managed, logically, in one place, even if it is referenced across the world, and which provides local replication of data (for performance) and a good degree of practical fault-tolerance. It is a reusable store of user, security and hardware configuration information that can be shared by all the applications running on a computer system. You don't need to worry about keeping the user access permissions on either side of a slow and unreliable WAN link synchronised, using complex, and largely duplicated, code for every application, because the Directory can manage this for you. However, a directory is a compromised database. Providing the ACID characteristics of a distributed transactional database is expensive and / or compromises performance. ACID describes the desirable characteristics of a truly enterprise-capable database:

- **Atomicity** - updates either complete or back-out automatically, leaving no possibility of partial updates;
- **Consistency** - the data "state" is always correct throughout the database, even on different servers;
- **Isolation** - an update being made is isolated from other transactions until it completes successfully;
- **Durability** - changes, once made, aren't lost.

The nature of the information in a directory is that it is "mostly" read-only, and it usually doesn't matter if your address information is out-of-date for 15 minutes on a server on the far side of the world (which compromises Consistency) and as long as Atomicity is maintained for the update of single objects, it can be compromised for sets of related objects (there is time to correct consequent errors manually).

You can manage a Directory as a "loosely consistent" database, optimised for read access, and as directory access is an obvious performance bottleneck in directory-enabled systems, doing so is essential for practical directories today. Microsoft doesn't use its relational database SQL Server for Active Directory, it uses an optimised 32-bit database engine called JET (similar to, but not compatible with, the database engine in Access), because SQL Server couldn't provide the performance required.

IBM and Oracle are currently better placed to provide an RDBMS directory implementation, but it seems that neither DB2 nor Oracle can really quite cut it yet in this application, although Oracle would doubtless disagree with this view. Oracle Internet Directory is a standards-based LDAP directory which claims to exploit the scalability, high availability and security features of the Oracle database. Commentators don't see Active Directory moving onto SQL Server (the version codenamed "Yukon"), for example, until around 2003, with the "Longhorn" release of Windows Server. By then, there could even be alternative, more appropriate, technologies for directory applications (the Associative Model of Data has potential in this area).

The loosely consistent directory, optimised for retrieval, is purely an arbitrary compromise. It limits a directory to storing information that is important, because storing and deleting obsolete information is an overhead (Active Directory currently doesn't support deletion of objects from its schema, largely because of the impact on historical data); short, so that maintaining distributed replicas is feasible; and read more often than it is updated, so that inconsistency issues are limited. Talking about Active Directory, we will accept the directory compromises usual today, but remember that they may prove less necessary as technology improves.

A good general guide to enterprise-scale directories and their issues is *e-Directories, Enterprise Software, Solutions and Services* by House, Hahn, Mauget and Daugherty, Addison-Wesley, 2000, ISBN 0-201-70039-5). This avoids that blind vendor specificity common in books about Directories: the authors work for IBM, so IBM's SecureWay directory features and many examples use Microsoft Active Directory, but this is far from excessive and Novell's eDirectory and even Banyan Vines StreetTalk are mentioned. Beware of anyone that tells you that there is only one right approach to directories; that the LDAP (Lightweight Directory Access Protocol) standard, which merely specifies an interface API, can fully replace the full-function X.500 specification; and that loose consistency is an end in itself rather than a useful compromise.

Considerations

The first thing to note before you start writing, or installing, directory-enabled applications in an operational environment is that you must have a working, stable directory installation to build on. This means not merely installing the software but also putting in place a coherent directory strategy, useful naming conventions and reliable support procedures - and trained staff. It is best to avoid a "big bang" implementation, and instead to go for an incremental approach with evaluation and "back-off" options at all stages if anything goes wrong. This approach isn't really supported by Active Directory as delivered, and you will need third-party utilities (from Quest Software, or NetIQ) to make this workable.

There isn't room here to explore the implementation issues in detail but they are explored in a paper written for Microsoft Consulting Services by Sanjeev Kamboj, Quest's Microsoft Solutions Manager for Europe, to explain them to a large Swiss Bank in London [*you'll find the document on this month's CD-ROM - Ed*].

Another useful source for real-world insights into the implementation of Active Directory can be found on the NetIQ Web site in *The Definitive Guide to Windows 2000 and Exchange 2000 Migration* - a free "registerware" eBook.

"Another useful source for real-world insights into the implementation of Active Directory can be found on the NetIQ Web site in "The Definitive Guide to Windows 2000 and Exchange 2000 Migration" - a free "registerware" eBook."

What To Do With It

When you have finally implemented your directory, there are four general uses for it and, given the trouble it has probably taken to get it right, it would be a shame to limit its use. The first, and often most important, use is as a conventional Administration Directory, which is used to control the cost of user, resource and security and management in large networked systems. This is the primary use of Active Directory, and until you've got this working properly there is no point in trying anything more complicated. However, once you have Active Directory working properly for Windows 2000 you can extend it to other applications. Sometimes, you have no choice (Microsoft's Exchange 2000 is automatically directory-based) but any enterprise-wide application is a candidate for directory enablement.

The second general use is as a Corporate Directory, which is used to present a single corporate identity to both the employees and customers of a corporation. This is usually done as an attempt to tidy up after a corporate acquisitions or business process redesign exercise, and is largely intended to address morale and marketing issues. Often this won't involve a single directory but the linking of multiple directories using metadirectory technologies - a metadirectory is a Directory that contains pointers to different Directories on different platforms and provides a single point of entry to the directory structure.

Microsoft is experimenting with metadirectory technology with Microsoft Metadirectory Services (originally a bought-in product called Zoomit) but metadirectories are also available from, for example, Critical Path and Novell (which calls it DirXML).

Thirdly, directories can be used to speed up new development by making existing information available, usually using the standard LDAP interface supported by most serious directories today. This shouldn't require modification of the existing systems as long as they use a directory but sometimes a minimal (or you won't bother) development effort will be required, using metadirectory technologies, to access information from proprietary directories or configuration files.

Finally, directories or metadirectories can be used to implement a global Internet Directory, storing details of customers, potential customers and suppliers outside your organisation but drawing on information already stored internally.

Metadirectories won't be covered in detail here, but writing your own directory applications usually involves extending the schema, as it is unlikely that the directory as delivered will contain all the object types your application needs. This must be done with care (especially if your directory doesn't support deletion of objects from the schema), as the directory is central to your IT system. If the directory breaks, no one gets any work done until you fix it and (because a directory compromises on update integrity in some ways) a directory isn't as robust as a proper database. Getting half-completed transactions out of a directory (even transaction processing sometimes fails, typically through technical support forcing the deletion of something important or forcibly removing a vital server from the system) isn't always easy, for example, and a transaction set often can't be defined across several objects, so an application failure can sometimes cause serious problems.

Mainframe standards of configuration management will be needed if you intend to update the schema. You will need to be confident that your programmers know what they're doing and obey all the rules - and don't even think of trying schema updates before your basic Active Directory migration has finished successfully, and is working as you intended. An alternative approach to schema update is to leave it to the vendors (who presumably know what they're doing) to package them with their software (as already mentioned, Exchange 2000 is the prime example of a directory enabled application for Active Directory and extensively updates the Active Directory schema). Even so, install the product in a test environment and evaluate a period of parallel running before unleashing a vendor's schema update on your operational Directory.

Given reasonably disciplined staff and programmers even extending the Schema yourself shouldn't be a problem with a properly designed Directory Service such as Active Directory. However, you must follow the rules - a schema extension that redesigns (overwrites) one of the standard schema objects could be disastrous, for example, but Microsoft uses unique Object Identifiers, registered with a standards body, which should ensure that new directory objects couldn't be confused with existing ones. Even so, schema updates must be managed and a DBA (DataBase Administration) group to manage and co-ordinate directory updates would be a good idea, especially with a directory like Active Directory, which doesn't currently allow schema changes to be removed.

> *"Quest Software's FastLane ActiveRoles, awarded the Windows 2000 Magazine "Best of Show" award in the Windows 2000 category at Tech Ed 2001, is a good example of a directory-enabled application."*

Example

Quest Software's FastLane ActiveRoles, awarded the Windows 2000 Magazine "Best of Show" award in the Windows 2000 category at Tech Ed 2001, is a good example of a directory-enabled application. It extends the Windows 2000 security model by grouping sets of permissions as "roles". Determining the exact Access Control Entries (ACEs) needed by a typical developer, say, may take some thought but once they are associated with the "Developer Role" setting up subsequent developers, and auditing the appropriateness of developer ACEs, becomes easy.

FastLane ActiveRoles extends the functionality of Active Directory in other ways. It provides rules that let you control the format of Directory information and apply corporate standards to it. This is almost essential in an Enterprise directory if it isn't to become cluttered with lost or unused garbage over time. Rules can also be used to enforce simple workflows: to ensure that new users are automatically added to the appropriate security group as they are created, for example. This enforces a degree of logical reverential integrity, and without some such scheme you won't be able to rely on the directory - and it won't get used to its full potential.

It is appropriate for ActiveRoles to extend the Active Directory schema because:

- Assignment of users to roles can change, but is relatively stable - the information is read more than it is updated, and the Directory ensures that updates need only be made once, regardless of where and how often the Role is used.
- Small amounts of "directory-like" information are involved (this was a design goal for the product, so as to avoid excessive replication traffic).
- The end result is useful - the hassle of updating the schema is justified.

Extending the schema in this way means that ActiveRoles integrates well with native Active Directory facilities such as "Native Delegation". It makes the native facilities more usable and adds extra functionality - rules and "business view" groupings - that aren't available in Active Directory as delivered.

The alternative to modifying the Active Directory schema when designing directory-enabled solutions is to use a proxy approach - to build an external database structure, referenced from the directory, to hold the extra information. This is, arguably, less elegant but it must be stressed that the choice between these approaches is somewhat arbitrary and that each has advantages and disadvantages. A proxy solution is going to duplicate some functionality (replication and update integrity) already in Active Directory but there is less risk of damaging the operation of the critical enterprise directory if something goes wrong. You can store large amounts of information in a proxy without impacting directory performance, for example, and can back off a proxy-based application easily.

Quest Software, which extends the schema to implement Fastlane ActiveRoles, chooses to use a proxy-based approach for its migration tools, largely because migration information ceases to be of interest when migration finishes but associated schema extensions cannot be removed from the schema currently. In fact, even ActiveRoles has a mode in which information can be stored locally in the registry, so you can evaluate it without extending the Active Directory schema.

In House?

Should you use your operational Directory for your own in-house applications? If you have the discipline needed to manage updates to a shared mission-critical resource, then certainly you should. If you have trouble co-ordinating work across several teams and have the occasional disaster caused by inadequate testing, then probably not. Your Directory is a valuable investment and you should maximise the return on this investment but an organisation that has experience of a large operational mainframe facility will probably cope better with managing directory updates than one with experience limited to departmental NT systems.

The advantages of directory-enabling in-house applications are efficiency and improved time-to-market. User configuration management, security access controls and so on are difficult to code if they have to operate across a distributed network. If you put them into a directory, replication (duplication of data locally for improved performance) and "loose consistency" is managed for you - together with a degree of fault-tolerance - and you know that these facilities work. With less code to write there is less testing to do and you can deliver more functionality faster, and users will need less training, as existing operational routines can be re-used.

> *"The advantages of directory-enabling in-house applications are efficiency and improved time-to-market."*

There is no question that, as Microsoft pointed out when Novell pioneered directory-enabled networking, the directory involves some overheads. It puts a database management system into a network operating system, which then needs extra disk space and CPU power to cope. A directory introduces potential performance problems, particularly associated with replication. Although a Directory is logically centralised - it can be managed in one place - its data is duplicated locally in the interests of performance and copying a large change to all the replicas can bring a network to its knees. A directory is also a single point of failure - bring down the Directory and no one will be able to log into any system, anywhere.

Vision

But the biggest potential issue with a directory service, however, is training. Windows NT admin staff typically don't understand enterprise computing in general or directories in particular and it isn't safe to let them loose on a directory system until they are trained and have some experience: hiring a few Novell CNEs, who have been using directories for years, may be a good move.

In the face of the issues with directory deployment, you must have a clear vision of what directory enabling your network offers you in return. Perhaps the best place to find out what Microsoft's Active Directory could aspire to is the Novell environment. Novell's eDirectory (previously NDS, Novell Directory Services) is a mature product and, as well as managing NetWare, it is used to manage access permission for network attached storage (the Netdevice range), to manage the desktop environment (the Zenworks product), to manage security (BorderManager) and so on. Someday Active Directory will offer similar facilities and you will probably want to use them - so plan an incremental directory strategy now to ensure that you don't make decisions that take you to a dead end.

However, although having a good directory strategy based on a clear vision is important, don't expect users to buy into it directly. Users aren't interested in directories although they are interested in some of the things you can do with them - single system sign on, a desktop that moves with them as they log into different computers across the enterprise, properly managed security that works consistently wherever you are. You will need business sponsorship for building a directory-enabled organisation but don't sell directories to business users, sell the things you can do with directory-enabled applications.

Critical Success Factors

If you want to do more than just implement Windows 2000 - that is, if you want to take full advantage of the opportunities offered by Active Directory - then there are a few simple points to consider. These may not guarantee the success of a major directory initiative, which rather depends on the ability of your organisation to function as a cohesive enterprise in pursuit of defined goals, but ignoring them will probably ensure failure.

1. Plan your directory implementation carefully before you start - this is your one chance to get naming conventions right and to bring your network into line with your organisational structures. Don't waste it.
2. Choose an incremental upgrade to a Windows 2000 domain set-up in parallel, rather than a "big bang" upgrade in place. This avoids impacting your current production service and provides a backout path at all times, although it does mean investing in a toolset such as the FastLane DM/Manager Migration tool. However, increased safety is worth the extra cost.
3. Aim to implement a clean and tidy Directory without incipient problems. Tidy up your NT domain (non-standard names, resources that are no longer needed, users that have left the company) before you start. The SAM on your Primary Domain Controller, which becomes the Active Directory, was limited in size and function (which is why you're moving off it). Make sure that your new hardware has enough disk space and CPU power to cope with a (potentially) much larger directory database management system - be generous.
4. Do not start on directory enabling applications, and the enterprise itself, before you have finished the migration to Windows 2000 and confirmed that it is working properly - this includes confirming that your Help Desk and support procedures can cope. Only then, implement Exchange 2000 (the killer directory-enabled application) or whatever - and don't even think of modifying the Directory schema yourself until it has been shown to work properly for a while.
5. Invest in formal classroom-based training in Directory administration and maintenance, at least for key staff.
6. Treat the directory as a corporate resource. Consider setting up an administration group for the Directory, including specialists in it internal operations, and make this responsible for co-ordinating changes to the directory.

> *"Plan your directory implementation carefully before you start - this is your one chance to get naming conventions right and to bring your network into line with your organisational structures."*

References

Associative Model of Data
www.associativemodelofdata.com

CriticalPath's InJoin Directory Server
www.cp.net/products/injoin_dirserver_overview.html

Jon Honeyball's article on future AD technologies
www.theregister.co.uk/content/4/20879.html.

IBM SecureWay Directory schema
www-1.ibm.com/servers/eserver/iseries/ldap/schema/

IPlanet Directory Server
www.iplanet.com

Microsoft Metadirectory Services
www.microsoft.com/windows2000/server/evaluation/news/bulletins/mmsroadmap.asp

Microsoft's view of Metadirectories
www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/identity.asp

NetIQ
www.netiq.com

Netscape Directory Server
home.netscape.com/directory/v4.0

Novell eDirectory
www.novell.com/products/nds

Oracle Internet Directory
technet.oracle.com/products/oid/content.html

Quest's Fastlane software suite
www.quest.com/solutions/ms_admin_and_deployment.asp

University of Michigan open-source Directory Services Administration Guide
www.umich.edu/~dirsvcs/ldap/doc/guides/slapd/toc.html

There is just one final point to emphasise - enterprises using Windows NT above the departmental level will almost certainly move to Windows 2000 and will deploy Active Directory (especially if they use Exchange - Exchange 2000 is a huge improvement on Exchange 5.5). However, they may not necessarily deploy Active Directory as their Enterprise directory. Don't assume that because you have Active Directory that it will be the only directory you ever need - although copying information manually between dozens of unrelated directories, not uncommon today, isn't a good idea either. The introduction of the LDAP standard means you can freely access information in different directories from all your applications (although, as LDAP defines an interface rather than functionality, you may find that an LDAP-compliant directory doesn't support all the functionality of something like Novell's eDirectory). Alternatively "Metadirectory" technologies let you combine different directories into a common logical structure.

As well as Active Directory, important directories are produced by IBM (SecureWay), CriticalPath (InJoin), Novell (eDirectory), iPlanet (Directory Server - a joint Sun-Netscape initiative), and others, each with particular strong points, and most of these (unlike Active Directory, which is limited to Windows 2000) run on several platforms. There is even an open-source LDAP-compliant directory, with X.500 gateway and replication service, from the University of Michigan called slapd (Stand-alone LDAP Daemon - the details of LDAP are defined in RFC 1777). Of course, Active Directory may become the accepted *de-facto* standard Enterprise directory (it may not too, it isn't there yet) but at this stage, don't limit your horizons to one directory vendor on one platform.

> *"Although having a good directory strategy based on a clear vision is important, don't expect users to buy into it directly."*

PCNA

New Reviews from Tech Support Alert

Anti-Trojan Software Reviews

A detailed review of six of the best anti trojan software programs. Two products were impressive with a clear gap between these and other contenders in their ability to detect and remove dangerous modern trojans.

Inkjet Printer Cartridge Suppliers

Everyone gets inundated by hundreds of ads for inkjet printer cartridges, all claiming to be the cheapest or best. But which vendor do you believe? Our editors decided to put them to the test by anonymously buying printer cartridges and testing them in our office inkjet printers. Many suppliers disappointed but we came up with several web sites that offer good quality cheap inkjet cartridges with impressive customer service.

Windows Backup Software

In this review we looked at 18 different backup software products for home or SOHO use. In the end we could only recommend six though only two were good enough to get our “Editor’s Choice” award

The 46 Best Freeware Programs

There are many free utilities that perform as well or better than expensive commercial products. Our Editor Ian Richards picks out his selection of the very best freeware programs and he comes up with some real gems.

Tech Support Alert
http://www.techsupportalert.com