The LDAP Guys™

# Assessment of Microsoft's Active Directory Application Mode (ADAM) as a Potential Enterprise Directory Technology versus OpenLDAP and Other LDAP Offerings

*Symas Corporation*

*Version: 1.0*
*Published: October 2007*

## Abstract

A potential customer of Symas™ Corporation recently asked us to comment on the suitability of Microsoft Active Directory (AD) and its Application Mode (ADAM) for deployment in enterprise production LDAP environments. After a period of document research and product testing, assisted by colleagues at HP and elsewhere, we have reached initial conclusions and made preliminary observations. This white paper documents this step in a longer planned research effort. This release of this document captures these early findings. Subsequent releases of this document will add depth, particularly including data from benchmarks not yet run.

Microsoft Active Directory is an evolutionary enhancement to Microsoft's Windows NT network operating system (NOS) directory. AD added many features, the most important of which was LDAP support (in Windows Server 2000) and ADAM (in Windows 2003). AD appears constrained by technology elements which limit its ability to flexibly and robustly provide LDAP services. In Windows 2003 Microsoft introduced ADAM, apparently based on AD technology intended to address some of those limitations. Overall, both of these directory technologies appear limited in capability and are of questionable value for production Enterprise directory implementations.

# Contents

## Introduction

Microsoft's Active Directory was introduced in Windows Server 2000. Active Directory Application Mode was introduced in Microsoft Windows Server 2003. These directory services technologies claim LDAP interfaces and standards compliance. As we'll see, the claims are overstated. Microsoft has not achieved dominance in the general purpose directory space with AD technology (both AD and ADAM). The incumbent directory services packages (Red Hat DS, Sun DS, Oracle's Internet Directory, IBM's Tivoli DS, CA's eTrust Directory, Novell's eDirectory, etc.) are well supported by their sales, marketing, and service organizations. Most of these LDAP products demonstrate significant technical superiority. Most enterprises have not, to date, jogged off and adopted Microsoft technologies for their enterprise LDAP applications.

As the years passed, Microsoft has nurtured their Windows Server user base and it has become quite respectable with many organizations happily deploying the platform in the Enterprise. There is still an enormous UNIX/Linux/Mainframe presence and the tension between those worlds continues on both technical and political grounds. Microsoft launched (or stepped up) an energetic sales campaign to those who administer and develop for their platform to convince them to extend the use of Microsoft's AD and ADAM technologies and to replace UNIX/Linux LDAP solutions or to design and deploy ADAM applications rather than integrate those applications and data with existing enterprise directories.

This is predictable and sensible (from their viewpoint) given their pride in their accomplishments and their tendency to downplay many of the complex realities of corporate IT. The people developing for, administering, and otherwise nurturing those Windows Server systems see, in this suggestion, a potential expansion of their influence and role in the Enterprise IT structure. This may represent healthy integration of technologies in the enterprise but the political aspect tends to stimulate aggressive adoption of technologies for other than technical reasons. Generally, that is a bad idea and would be especially so, in our opinion, in this instance.

In this paper, we analyze Microsoft's claims and technology based on information from recent copies of Microsoft documents and from testing the product firsthand with enterprise directory data. We discuss some of the implications drawn from those claims. We also touch on some indicative performance data available from Microsoft and from Symas's benchmarks of OpenLDAP. The conclusions we draw are that Microsoft's AD and ADAM are not suited for use as the technology base of enterprise directory strategies. In fact, from a careful reading of the documents, it should be evident that Microsoft are well aware of the limits of their technologies and, though they will likely never publicly endorse our conclusions, they will likely not be surprised by them.

# Background on LDAP and OpenLDAP

## A Brief Overview of LDAP

The Light-weight Directory Access Protocol (LDAP) is the Internet directory services standard. Originally, it was a bridge technology between X.500, the established but not yet dominant international directory services standard, and Internet (TCP/IP) applications. X.500 was first published in 1988. It is presently managed by the ITU-T but was developed when the organization was called CCITT in a collaboration with ISO. ISO were simultaneously developing and refining the OSI networking stack, driven by the traditional computing players. The CCITT represented the interests of both the computing community and the telecommunications industry. The resulting standard reflected the prevailing industry view of the future of networking and the evolution of directory services.

The networking industry expected aggressive adoption of X.500 by enterprises. There was a general perception that vast libraries of directory data would be available on OSI-based networks. The reality was that growth of X.500 was slower than predicted and the growth of the Internet was explosive. On the nascent Internet, there was panic that, as badly as they wanted access to all that great enterprise data, there was no way for them to easily access it.

At least two teams, one from the University of Michigan and one from PSI, developed software that accessed X.500 directories. The approaches were a bit different but conceptually similar. They collaborated on a draft RFC titled Light-weight Directory Access Protocol (LDAP). The protocol provided a set of APIs that enabled Internet applications and machines to access X.500. As time progressed, the LDAP community realized that X.500 was stalling and they enhanced the RFC(s) to define a complete Internet-based directory services stack, obsoleting the name but not replacing it given its popularity. It would be more appropriate to call LDAP (currently at Version 3 as recently updated in RFC 4510) the Internet Directory Services standard.

## OpenLDAP

The University of Michigan developed an implementation of LDAP through which the community tested various suggestions and proposals. The University of Michigan code was considered the reference implementation during the early evolution. Netscape hired away the core team from the University and they (quite legally) took a copy to build the original Netscape Directory Server. They teamed shortly thereafter with Sun and created iPlanet but the partnership foundered and both firms produced LDAP products which competed in the marketplace as proprietary (closed source) software products.

In 1998, the OpenLDAP project was formed. They also started from the (now dated) University of Michigan software and cleaned up the many known problems in the code, expanded its platform portability, and began significant reengineering. Informal estimates indicate that 80% or more of the original University of Michigan code has been rewritten. Significant restructuring of the code has resulted in remarkably flexible internal structures allowing construction of database “back-ends” to access data stored in many different low level storage technologies (Berkeley DB, SQL, LDAP, shell, meta, etc.). Another restructuring introduced overlays which provide access to the logic of directory operations and allow for the introduction of new capabilities without modifying any of the core OpenLDAP code. All of these extensions can be dynamically loaded as needed.

OpenLDAP has implemented fully online configuration through the LDAP protocol and eliminated the primary causes of server administrative interruptions. Replication technology (support for scaling out and improving availability) has been completely replaced. Numerous functional enhancements and extensions have been introduced in the core software and through optional dynamically loaded overlays.

High Availability is an important quality for the enterprise users. OpenLDAP has implemented a *mirrormode* high availability feature that provides Master-server-ready replicas ready to take over on Master server failure. This feature has been extended in OpenLDAP 2.4 to offer a full Multi-Master Replication capability. This has been a feature that was an advantage in AD and other LDAP software offerings. This release, available for testing and evaluation, closes that functional gap.

One of the primary focuses has been performance. Remarkable performance improvements have been introduced, leading to a server with dramatically faster response times and lower resource usage. This emphasis has been counter balanced with careful attention to logging, auditing, security, and reliability. OpenLDAP today is the most efficient, reliable, and flexible directory services software package available.

The OpenLDAP development community has evolved over the years. There are currently three core team developers and approximately seven active code contributors. There are additional contributors adding testing, process, and documentation capabilities. Symas has paid the salary of Howard Chu, now Chief Architect of OpenLDAP and other less active contributors to work on OpenLDAP since 1999. Howard and other Symas participants are responsible for a preponderance of the innovation and development of OpenLDAP in that period.

## Microsoft on AD and ADAM

Microsoft published a document entitled *Introduction to Active Directory Application Mode* in 2003. It appears to be current. The 2003 edition is still available on Microsoft's Web site for download at:

> http://www.microsoft.com/windowsserver2003/techinfo/overview/adam.mspx.

This section of the document extracts quotes from the document, in the order they appear in the original document. We attempt to provide a different view of the statements, some clarifying and others interpretive.

For the remainder of this section, block indented italic text are quotations from the Microsoft document. So are titles when in quotes. The rest of the text is our commentary.

### "Lack of Directory interoperability"

> *Many directory services simply do not operate with each other. A historical example is the original X.500 directory that did not support LDAP. Even today, some products that implement a directory do not support LDAP or other widely used protocols.*

These assertions are controversial to the point of being inflammatory. As to the first, there is little or no excuse for directory interoperability problems in 2007. The remaining commercial X.500 implementations all provide adequate LDAP implementations. Many other specialized or proprietary directory implementations (including Active Directory (AD)) also have LDAP interfaces available. The introduction of ADAM, a more flexible LDAP capability using AD technology, is a strong indication that Microsoft itself felt its product line was deficient in that regard. Other software providers have responded in like manner. This, and particularly Microsoft's own actions, tend to debunk that statement.

Regarding the second, OpenLDAP offers LDAP access to data stored in RDBMSs so any data available through ODBC can be accessed via the LDAP protocol. LDAP-dependent applications have access to many different LDAP capable products and to a vast amount of LDAP-accessible data through OpenLDAP's integration capabilities. And applications without LDAP support are, generally, evolving to add support as the Internet and software communities have recognized the standard and changed their roadmaps to take advantage of it.

Microsoft artfully neglects to discuss another aspect of interoperability. These Microsoft claims, in fact any and all Microsoft claims, must be evaluated in the light of the fact that Microsoft server software only runs on Microsoft Windows platforms. That includes AD and ADAM. Among vast numbers of Windows desktops and related servers it is often hard to remember that most large enterprises use other platforms for server applications. AD and ADAM technologies are not available for directories that need to be deployed on non-Windows systems in heterogeneous collections of servers. This blocks interoperability with directories needed or desired to run on hardware architectures Microsoft Windows does not support or on servers running a non-Microsoft OS. This lack of interoperability restrains enterprise freedom of action and, if ignored, provides a misleading impetus to platform change for unsupported reasons. OpenLDAP and other enterprise-grade LDAP-capable directory products are, or should be, generally available on all the relevant server platforms including Microsoft Windows.

## "Lack of choice"

> *Some vendors ship solutions that are certified to work with only a limited subset of the directory services that are in use today. A customer of these vendors may be forced, for support reasons, to implement a directory service that is not already used in that customer's organization.*

One might respond that this is simply the sowing of Fear, Uncertainty, and Doubt (FUD). The first and most significant reaction to the statement is that Microsoft is shipping solutions that are certified to work with only a limited subset of the directory services that are in use today (AD and ADAM). Microsoft customers are, by derivation, subjected to the limitations and costs pointed out as there are unavoidable or compelling reasons (as we'll see) to continue to run or install superior and more mature directory technologies. Microsoft forced the adoption of a directory service that aggravated this problem when they well could have adopted LDAP as the NOS directory technology for Windows and avoided this problem. The Microsoft decision to neither adopt the technology in the '90s nor to upgrade the AD technology to reflect the current state of the directory art makes AD (and ADAM) a "solution that [is] certified to work with only a limited subset of the directory services that are in use today".

The second point is that this was written in the middle of a period of active and committed development of LDAP support by proprietary and open source developers. This resulted in broad availability in non-Microsoft software of LDAP-compliant clients (and servers where needed). Customers today have many LDAP-enabled software applications and LDAP servers to choose from. Microsoft's failures to provide the standard-conformant support on both the client and server software forces their customers to implement a directory service which does not support the Internet standard for directories.

## "Lack of coordination"

> *In some cases, groups that are isolated from one another in an organization install different business solutions. This can result in the deployment of multiple directory technologies.*

In the real world, this happens. Nobody has figured out how to stop it. More interestingly, the Microsoft platforms were, and continue to be, frequently introduced into enterprises by clever end-users who, isolated by choice or circumstances, take advantage of the capabilities of locally available platforms. These introduce AD and/or ADAM into true LDAP environments in much the same way criticized here. Those cases Microsoft, presumably, applauds.

In a professional IT environment the introduction of non-standard or otherwise different directory technologies should be done with a plan for integration and/or migration to the superior (if standard) technology. Since these disruptive behaviors happen (and often enrich the overall IT fabric of the enterprise) it is essential that the strategies are periodically re-evaluated and resources are dedicated to evolution of the directory landscape to best integrate and refine the enterprise's directory assets.

### "Lack of security interoperability"

> *Business solutions seldom allow the use of identity credentials that are stored in a directory service but that are not associated with those specific solutions. This means, once again, deploying even more directory services to act as the secure credential stores for each individual business solution.*

Many applications are taking advantage of LDAP's proven and secure ability to validate security credentials of virtually any form through the Internet standard protocol. It would appear that Microsoft insists that unification can never occur. Microsoft's platforms continue to rely on proprietary directory protocols and interfaces that force users to use AD. In addition, their failure to provide flexibility to support the needs of other directory users and applications aggravates this situation. It denies the enterprise the opportunity to move these credentials to directory technologies superior in security, flexibility, and performance.

### "Increased security risk"

> *As business solutions that rely on directories proliferate, it becomes increasingly challenging to ensure that these solutions integrate effectively with business processes. As employees, partners, contractors, or customers initiate or change their relationships with an organization, it is crucial that their access to VPN, PKI, NOS, or other business solutions is initiated or changed immediately. When management overhead causes slow initiation, productivity is affected. On the other hand, when changes are not quickly reflected in the various directories, a security risk develops, which could allow an unauthorized individual to have access to the network.*

It is tempting to pass over this assertion without comment. But it is more clever than that. There is a tacit assumption that AD can and will address all this. That is unlikely, certainly now and probably into the future. Given the weaknesses admitted to further down in the paper, it is improbable that AD will be extended to provide these services over all the relevant protocols and services. It will most likely continue to be flawed and unable to address the requirements of an enterprise directory.

The real point here is actually about provisioning and updating permissions and credentials. It is a real problem. Unification within the context of OpenLDAP's capabilities, including various ways of interacting with AD, is becoming more practical. Symas's people can craft solutions to these problems today. The same appears true of those who sell and support other LDAP technology products. AD's implicit proposal of collecting them and validating them via AD is currently unrealistic and ADAM is likely only to make the situation more complex, not simpler.

### "High cost of ownership"

- *Every business solution that is based on a different directory technology requires the following:*
- *A staff that is trained on that directory technology*
- *Different operational and administrative procedures*

- *Maintenance of additional software licenses and separate support agreements*

Enterprise IT organizations already have quite an investment in LDAP skills and use LDAP directories to support many applications. Introducing ADAM outside the normal NOS usage of AD introduces the need for new architecture and design skills. ADAM operational characteristics may be similar to AD's but the complexity of the many proposed directories and coordination of the propagation of Authoritative Data and the rest of the operational aspects will add training, procedural, and maintenance costs. Since, as will be discussed later, AD is demonstrably unsuited to the more complex requirements of enterprise directory architectures, the investments made during its introduction are additive and highly unlikely to offset investments already made and ongoing costs associated with technologies which will not be displaced. Failing to take advantage of the experience and skills of LDAP architects, developers, and administrators merely increases the proposed true cost of adoption of ADAM.

## "Increased cost of success"

> *Some directory technologies are licensed according to the number of objects that are created in the directories. This means that licensing and maintenance costs start spiraling upward as a business solution becomes more and more successful. Today, this situation affects organizations planning to deploy extranet access management solutions that are intended to service millions of customers.*

This is a pointed criticism at Sun. Sun licensed their directory software based on the number of entries (objects) stored in the directory. At present, they are the only supplier with such a pricing model. Oracle still charges by the processor used but most LDAP technology providers charge only per server. So does Microsoft, though it is usually bundled into the OS pricing. It is probably fair to think of AD as free, as a side note, because it can't be extended to take on the responsibilities of OpenLDAP, Sun's, Oracle's or any other LDAP-compliant enterprise grade directory server. It is best to consider it a proprietary integrated specialized directory and not to think of it in true LDAP terms at all. Ample evidence follows.

It is also quite amusing that Microsoft makes this particular criticism as AD user entries are used as the basis of Windows user licensing. The largest enterprises with true enterprise licenses don't care but many substantial corporations end up paying quite large license bills as they achieve success with "windows users".

Finally, on this point, the more interesting question is "What is the cost of success?". If you answer it with a more sophisticated view of total cost of ownership and factor in the cost of the systems required to support a growing directory technology, you'll find that the resource demands of a growing AD directory are quite large. AD requires several times more memory and processor power than OpenLDAP and that translates directly into Kilowatts of power for computers and HVAC. AD neither scales as smoothly, administers as seamlessly, or behaves as reliably, either, all of which translates to costs of success as well.

## "Lack of business process integration"

> *Directory information can be volatile. As users move from one group to another, change office locations or telephone numbers, and change names or job titles, their*

> *information must be updated in the directory. If this information is relied on by other business solutions that have different directories, the other directories must also be updated. Without an automated process to make these changes, data becomes stale and unsynchronized across identity stores.*

This particular claim needs to be addressed in two pieces. This first piece suggests that directory information **can** be consolidated into Active Directory and that taking such action would address the straw-man problem they raise. For many reasons, the existence of multiple directories can't be eliminated. For reasons we've alluded to and will discuss later, AD is probably a weak candidate to be the directory for unification.

However, the concern is quite serious. Quite a lot of work has gone into solving these problems. There are numerous virtual and meta directory technologies available that address some or all of these issues. Some of these address the specific challenge of AD. OpenLDAP offers capabilities as standard features that lessen the need for enterprises to acquire expensive add-ons for virtual and meta directories.

> *What organizations really need is a directory that they can deploy to support both their NOS infrastructure and their applications that can, where appropriate, take advantage of the security that is built into the NOS infrastructure. Active Directory Application Mode achieves this goal without the burden of expensive training, additional licensing, or operational costs that can be incurred by the installation of an additional directory technology to support directory-enabled applications.*

No, the requirement stated in the first sentence should **not** limit the enterprise to the level of any one directory technology. This is an implicit claim that the NOS infrastructure has superior technology which is not true. Besides, in this statement, it is implied that AD, the NOS directory, is the most secure directory on the planet and that's a claim that is at least controversial. Linking this to ADAM which relies on AD and its infrastructure is not a particularly useful assertion once you realize that OpenLDAP, properly configured and deployed, may well be significantly more secure and that the training, licensing, and operational costs are either sunk or unavoidable anyway. This is a clever bit of propaganda but none of the assumptions can withstand scrutiny.

## ADAM's Justification

> *Active Directory Application Mode is a new capability in Active Directory that addresses certain deployment scenarios that are related to directory-enabled applications.*

It might be more honest to say that ADAM is a new capability related to or derived from AD. It appears to be a new download, different code, and quite separate from AD though it might share quite a lot of code with AD.

It is **certainly** more honest to say that ADAM addresses certain "deployment" or "production limitations" than the more forgiving "scenarios". This is a clear and unambiguous statement that AD fails to provide the flexibility, extensibility, and other attributes needed to be a true directory services technology. AD may be excellent as a NOS directory, but this is an admission that it is NOT an LDAP directory. It is a NOS directory that supports LDAP access to its data.

ADAM is an attempt to extend it but then ADAM doesn't measure up as a competitor to OpenLDAP or any of the other true LDAP directory services packages.

> *ADAM runs as a non-operating-system service, and, as such, it does not require deployment on a domain controller. Running as a non-operating-system service means that multiple instances of ADAM can run concurrently on a single server, and each instance can be configured independently.*

Here we get another clear and unambiguous statement about Microsoft's system and directory design. First, LDAP directories which are fully capable of providing NOS and all other directory services are not required to **be** on any particular server to provide excellent, secure, high performance services. The requirement that the directory be deployed on a class or type of machines is another limitation on the freedom of action for the customer. Furthermore, there is no particular demand on most LDAP servers to run in any mode or under a specific user ID or restrictions. AD is inflexible in this and that means that experimental or educational instances are difficult to use.

The fact, admitted to in the second sentence, that only one instance of AD can be run on such a server is a further weakness in its architecture. Neither of these facts is by any means an advantage over any LDAP competitor. By extension, the fact is that the claimed advantages for ADAM are only advantages when compared to AD. LDAP software packages routinely offer these capabilities and we are unaware of any of them which run in some restrictive "service" mode.

> *Active Directory Application Mode represents a breakthrough in directory services technology that overcomes the previously mentioned obstacles, maintains flexibility, and helps organizations avoid increased infrastructure costs.*

ADAM may be a breakthrough for Microsoft. It does not appear to offer any significant technical innovations compared to most existing LDAP packages, especially OpenLDAP. Having reviewed all of the obstacles the claim refers to, we conclude that the noted obstacles are either artificial or erroneous.

## Architectural Issues with ADAM

Microsoft makes much of the "simplicity of application mode". The underlying philosophic assumption is that applications only need little directories and share little data among themselves. ADAM enables construction of numerous directories, each addressing the need of an application and relying on AD for user credentials, etc. We are quite confident that their Microsoft contacts consistently suggest this approach to Directory Architecture but it is flawed at the root.

Microsoft has aggressively centralized NOS user information of all kinds in its NOS directory (AD) and its system and application configuration data in its internal OS directory (the Windows Registry). The Registry's design is such that there are numerous instances of failure to share application-related user preferences and settings as a result of the rigidity of first-pass design of the hierarchy. AD is a bit more flexible but marries meta data to structure which tends to restrict its flexibility and robustness. AD is simply too inflexible to be the encompassing technology for a complete enterprise directory. AD is too tightly integrated into the OS to radically change. So the only decision open to them was to recommend directory proliferation, a deprecated philosophy even in Microsoft. They were critical of it in the justifications and then, later in the paper, turned around and recommended it.

The general direction of the directory services community is to unify directories, extend their capabilities to flexibly support multiple applications, and motivate broader adoption of the standard as a way to simplify the IT environment and application complexity. This recommendation of proliferating application oriented directories is working opposite the impetus of most directory strategies.

The proliferation of directories almost inevitably leads to an increase in redundancy of data among the data stores (directories). It is compelling to the application designer to include all the data relevant to their application into "their" directory. They may arrange for data feeds or other synchronization approaches but these redundancies aggravate security and manageability problems and the resource drains on the overall IT infrastructure.

Microsoft admits of a more serious problem with AD:

> *For example, data for an application might contain highly volatile information, causing high replication traffic that could strain network resources if it is stored in the NOS directory.*

This clearly states a concern that AD and its replication is threatened by large numbers of updates and the load from replication. There is a myth that LDAP server software is designed for a high percentage of read (bind and search) access and is somehow not as good at updates (add, modify, delete) as traditional databases. This is often asserted by people otherwise highly respected in LDAP circles. It is simply false today.

Virtually all the enterprise grade LDAP software packages have adopted rigorous ACID (Atomicity, Concurrency, Isolation, and Durability … with cascading rollback) capable database technologies. This means they are as robust and accurate at update data integrity with full transaction support as any other database software. In any ACID-compliant database, updates are much slower than reads because the ACID support introduces serialization (locking) and other code to support the capability. Those ACID overheads are consistent and produce similar performance challenges to all ACID-compliant databases.

LDAP directory servers are now very capable of supporting highly volatile information. Even the newer replication capabilities are capable of providing more timely updates to downstream replica servers. This means that there are general-purpose LDAP servers for whom volatility is no bigger a problem than it is for any enterprise database.

Microsoft's expressed concern that Active Directory replication might cause too much network traffic tends to limit its value as an enterprise directory technology because enterprises increasingly want to incorporate more volatile attributes to various types of objects. There are powerful applications emerging that press for location-based data, for user provided and managed data, and for system or device status data to be associated with objects in the directory. This assertion that the directory be subdivided or fragmented defeats the objective of data redundancy reduction. It also increases complexity.

> *Application directories evolve over time: business requirements change, forcing changes in directory schemas or configurations. Active Directory Application Mode runs as an independent service, as opposed to an operating system service. Therefore, you can use Active Directory Application Mode to modify local or targeted ADAM instances without making changes to your organization's directory infrastructure.*

This is quite amusing, actually. Active Directory is a System Service. So is OpenLDAP when installed correctly on Microsoft Windows. The Microsoft argument in this statement is that there is some problem with changing the directory infrastructure. We can only conclude that there are obstacles to changing Active Directory's schema or configuration. We should not conclude as they intend us to that there are barriers to changing schema or configuration on enterprise-grade LDAP servers.

The argument is made that it is advantageous to isolate application attributes and create fragmented directories to avoid managing the integration of new application requirements into the enterprise directory. This runs contrary to enterprise demands for reduced cost and complexity in the directory infrastructure.

> *Active Directory Application Mode is easily installed or uninstalled on developer workstations. This allows rapid restoration to a clean state during the application prototyping and development process.*

This is only an advantage over AD, not real enterprise directories. Any of the enterprise packages allow you to install test instances with full function on the same machine as a production directory or on any other machine you'd like to use for testing. Laptops or desktops make fine test systems. In OpenLDAP, you can merely clear out the database and, if needed, the configuration data and not have to reinstall.

A little further down, Microsoft shows their anticipated architecture for a directory enabled "Web portal" in Figure 1, copied below:

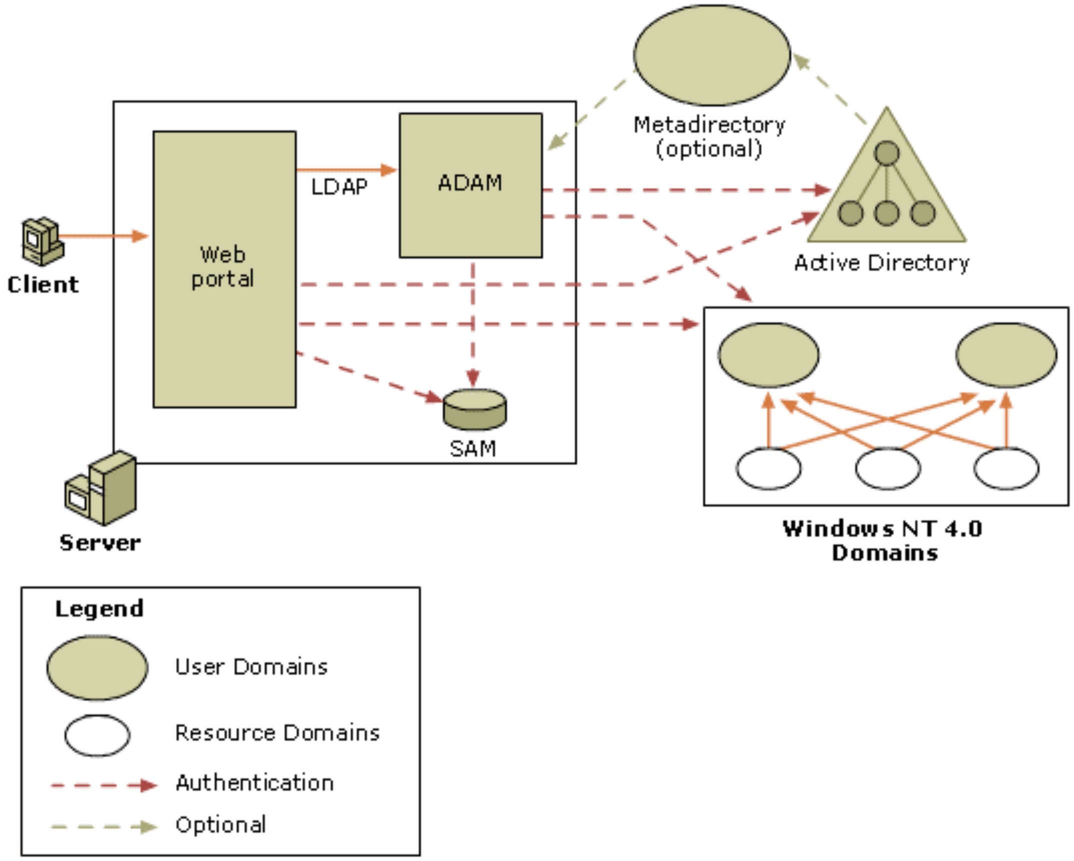

*Figure 1: Microsoft's original "Application Specific Solution"*

This example shows the complexity introduced by forcing the NOS and application directories apart. It also shows the real complexity of the Microsoft security solution with an instance of the "Security Accounts Manager" (SAM) also in play. There are several ways to integrate AD into an OpenLDAP directory and store the application data. Symas has an NT-LDAP Gateway to make NT directory data available through OpenLDAP as well. We would expect the solution would look more like Figure 2, below.

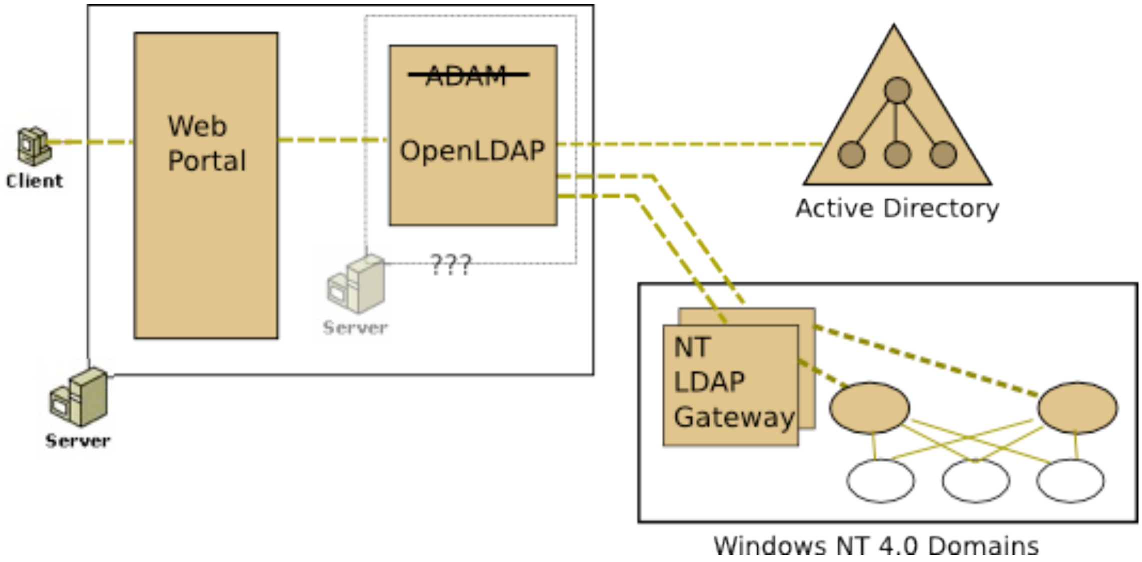

*Figure 2: An OpenLDAP-based "Application Specific Solution"*

Notice how much simpler this is. What is not visible is that there is only one directory so there is a minimum of points of provisioning and management. To the extent that the AD forest and/or NT 4.0 domain directories need managing, they can be managed the way they've always been managed. There are new options available, but since the IT staff is trained and the procedures are in place for that NOS management, there is no reason for change. However, the enterprise directory remains coherent and there is no reason for redundant directory information.

OpenLDAP offers additional opportunities. The Web portal's authentication and application data directory can either be on the same server or it can be somewhere else in the fabric of the enterprise. OpenLDAP offers translucency so that attributes can be "added" to the Active Directory objects and stored in the OpenLDAP instance. Nothing is changed in the Active Directory. All translucent attributes remain in the OpenLDAP database. Translucency is an advanced way of using the virtual directory capabilities of OpenLDAP. OpenLDAP's powerful virtual and meta directory capabilities are built-in. Like the rest of OpenLDAP, they are configured through LDAP using the *cn=config* dynamic configuration capabilities.

## Active Directory Technology Performance

Microsoft published performance data for Active Directory performance on both 32-bit and 64-bit versions of Microsoft Windows Server 2003. This paper can be downloaded from:

> *http://www.microsoft.com/downloads/details.aspx?FamilyID=52e7c3bd-570a-475c-96e0-316dc821e3e7&DisplayLang=en*.

Symas has no benchmarks on exactly the same hardware configuration or using the same workloads. We have had to derive preliminary performance estimates for comparison from published benchmark data on other configurations and using different workloads. One source of benchmark data is the Symas benchmark results page: http://www.symas.com/benchmark.shtml. Another source of data is the Symas blog in two separate entries:

> *http://www.connexitor.com/blog/archives/archive_2007-m04.php#e130 and http://www.connexitor.com/blog/archives/archive_2007-m04.php#e131*.

For this comparison, however, we'll use the benchmark HP published during 2007:

> *http://h71019.www7.hp.com/ActiveAnswers/cache/393495-0-0-0-121.html*.

Microsoft uses its ADTest tool to do the benchmarks. HP and Symas used the open source *slamd* tool developed by Sun. Most LDAP benchmarking today is using *slamd*. We have no way of evaluating the quantitative difference introduced by ADTest through use of *ADSI* or other interfaces to AD. All *slamd* testing uses the standard LDAP network protocol.

### Preliminary Statement

This section relies on benchmark data done by HP and Symas measuring performance of OpenLDAP. This is the data in which we have the most confidence and represents the processes and tools we will use to develop authoritative benchmark data to address this comparison going forward.

Symas would expect that any competitive LDAP directory server software package would achieve performance (efficiency) substantially higher than AD. The sources referenced above contain or link to benchmarks of some other LDAP packages and the data to be found there support our assertion that LDAP software is substantially more efficient and likely faster on all metrics.

### Simple Bind Benchmark Data

Microsoft Active Directory achieved 3214/second "simple bind" operations on the 100,000 entry 32-bit configuration and 3079/second on the 100,000 entry 64-bit configuration. Symas and HP have not isolated simple bind performance in their recent benchmarks. They use the Authentication Rate workload defined in the *slamd* job libraries. The Authentication Rate workload does an anonymous search followed by a bind so it performs quite a bit more work. On the HP ProLiant systems, OpenLDAP delivered 12,800 to 13,600 authentications per second (depending on model) for a 250,000 entry database. That's better than four times the rate of binds and more like eight times the rate of LDAP operations.

For the 3,000,000 user (entry) database, AD 32-bit bind performance falls off dramatically as expected and the 64-bit simple bind performance dips below 3,000/second to 2,997/second. HP only tested two configurations with databases at 3,000,000 users. They delivered 13,043 and 13,639 authentications per second. On one configuration HP tested 5,000,000 users and got 13,700 authentications per second. All are substantially over four times AD's performance, two are over four and a half times AD's performance. Again, this suggests OpenLDAP performance is probably in the range of eight to nine times faster.

## Comments on AD and ADAM performance

Symas does not have adequately rigorous benchmark data to compare to other Microsoft-published data. Without such data, it would be irresponsible to make quantitative performance claims. Pending a series of *slamd* benchmarks which will establish AD and ADAM performance for several standard *slamd* workloads (jobs) we can only say that based on various informal *slamd* benchmarks performed during development that we expect the simple bind comparisons to be representative of the comparative performance for other operations.

The memory required for AD to store the entries appears to be around three times that required for OpenLDAP. Again, this is extrapolating without direct measurements to compare. This will also have to be measured going forward. If these considerations prove out true, that would increase the cost to enterprise users of AD inefficiency. It is difficult to believe that ADAM performance will be dramatically better.

Finally, the indications here and anecdotal evidence indicates that Microsoft recommends keeping individual directories (AD Domain Controllers) relatively small. We can only speculate on the reasons but one suspects that there are both performance and reliability reasons for these recommendations. Many enterprise LDAP software packages have been demonstrated and benchmarked on directories of over one hundred million entries. OpenLDAP has been successfully benchmarked one hundred and fifty million quite large entries on a single very large system with excellent results. This difference will likely lead to larger and more complex directories on true LDAP products than on ADAM. This also reduces the likely complexity of maintenance, replication, etc.

Microsoft generally does not access AD through LDAP. That means Windows internally may see better performance than indicated by these preliminary findings.

## Prominent Limitations of ADAM compared to OpenLDAP

This section documents numerous (but by no means comprehensive) limitations of Microsoft's ADAM Directory Server as compared with OpenLDAP, a standards-based LDAP server. ADAM imposes restrictions that are often difficult, and occasionally impossible, to work around. These problems stem from ADAM's starting point of Active Directory's proprietary directory model and back-end storage database. Trying to retrofit LDAP into that proprietary model yields inconsistencies in every major area of functionality. Directory architects who assume that, unless explicitly restricted by LDAP, there will be no inherent restrictions in structuring and managing directory data, will face some rude shocks with ADAM.

Neither the LDAP standard nor the OpenLDAP product imposes any of the limitations described below.

### Schema Limitations

#### Attribute Character Length

ADAM imposes hard coded character length limitations to many standard LDAP attributes, and there are no documented means of increasing these limits. Examples include: *cn*, *o* and *ou* attributes limited to 64 characters; *displayName* limited to 256 characters; description limited to 1024 characters.

#### Attribute Value Limits

ADAM imposes a hard coded limit of 1500 values for search results for a multivalued attribute at one time. For example, given a group with 3500 members, ADAM clients would need to perform searches with the following type of Microsoft-specific attribute syntax to retrieve all of the group's members:

*member;range=0-1499*
*member;range=1500-2999*
*member;range=3000-**

If the ADAM client did not know how many members were in a group (the usual case), it would have to keep searching in increments of 1500 until it received a batch with less than 1500 results. Standards-based vendor applications and LDAP clients expect to retrieve all of an attribute's values in a single query, and would have to be rewritten to work with ADAM.

#### Relative Distinguished Names

ADAM mandates the specification of a relative distinguished name (RDN) attribute for each objectclass, and only one such RDN attribute is possible. To illustrate this, the following two entries could not simultaneously exist in an instance of ADAM:

```
dn:            cn=account1, o=company.com
objectclass:   top
objectclass:   companyAccount
cn:            account1

dn:            uid=account2, o=company.com
```

```
objectclass:    top
objectclass:    companyAccount
uid:            account2
```

If cn is chosen as the RDN attribute for the *companyAccount* objectclass, *uid* would not be allowed to form the RDN for additional entries. Also, once chosen, there is no documented method for modifying the RDN attribute after the schema has been initially loaded.

Additionally, irrespective of its definition in the schema, an RDN attribute is restricted to a single value. For example, ADAM disallows this type of entry:

```
dn:             nameAttr=support@company.com
nameAttr:       support@company.com
nameAttr:       customer_support@company.com
```

## OU Limitations

By default, ADAM only allows *ou* objects to be created under the following objects: ou, c, o, dc. The possibleSuperiors attribute of the *ou* object class in the schema has to be explicitly modified in order to modify the types of objects that can contain one or more *ou*.

## Distinguished Name Syntax Attributes

ADAM refuses to add DN-type attributes (e.g. *member*, *owner*, *manager*, etc) that refer to entries that do not exist in the local instance at the current point in time. This makes it impossible to use DN-type attributes to point to entries in a remote (virtual) directory. It also adds extra complexity to the common operational tasks of entry provisioning and reloading directory data. To illustrate, here is a sample LDIF file that cannot be loaded from scratch into an ADAM installation:

```
dn:             uid=manager@company.com, o=company.com
uid:            manager@company.com
assistant:       uid=helper@company.com, o=company.com

dn:             uid=helper@company.com, o=company.com
uid:            helper@company.com
manager:        uid=manager@company.com, o=company.com
```

ADAM will refuse to add the manager entry until the helper entry has been added, and it will refuse to add the helper entry until the manager has been added. The only workaround is to strip out all DN-style attributes from every entry, load the resulting LDIF file, and then apply a second LDIF file that adds back the DN-style attributes. In addition to being inconvenient, this reduces the efficiency of any bulk-load mechanism that might otherwise apply to loading large batches of entries in a single process.

## Objectclass and Attribute Definitions

In the ADAM schema, the inetOrgPerson objectclass is derived from Microsoft's *user* objectclass. This violates the definition of *inetOrgPerson* in RFC 2798 where it is derived from *organizationalPerson*.

Attributes such as *cn*, *mail*, *c*, *co* and others are restricted to being single-valued in Microsoft's schema. These violate LDAP standards such as RFC 4524, where the attributes are defined to be multivalued.

Microsoft has made seemingly minor changes to standard-track schema that will have substantial effects on other LDAP-enabled applications. For example, in AD and ADAM the definition of the '*uniqueMember*' attribute is specified as DN-syntax, whereas the standards document RFC 2256 designates it as 'Name and Optional UID syntax'. This seemingly minor change substantially changes the way these attributes are handled by AD and ADAM and severely constrains their values. This, in turn, breaks interoperability with other directory servers. Microsoft has not published these deviations from the standards and it can often be costly to locate and compensate for them, if such compensation is even possible.

## Data Access Limitations

### Anonymous Binding

By default, ADAM does not permit anonymous binding. This has to be explicitly enabled by poking the value 0000002001001 into the *dsHeuristics attribute* of the *CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,CN={GUID}* entry.

### Access Control

Users/groups from one ADAM naming context cannot be added to groups in another naming context. This means that an operations group created in *o=company.com* cannot be given administrative privileges to modify schema entries under *CN=Schema,CN=Configuration,CN={GUID}.*

Overall, ADAM's access control specifications are fairly limited when compared with standards-based LDAP servers. For instance, ADAM does not provide any ability to match on regular expressions. OpenLDAP, by contrast, provides a highly extensive, and extensible ACL framework -- including regular expressions, set-based controls, and even modular, dynamically loadable ACLs for customized access control code.

## Replication Limitations

ADAM replication is Multi-Master and based on scheduling parameters. Use of scheduling may reduce network traffic, but it guarantees longer periods of inter-directory consistency.

ADAM replication must be established during installation of the instance of the ADAM directory. In Microsoft terms, it must be added to the "configuration set" at that time.

Multivalued attribute conflict resolution is essentially non-existent. No resolution is attempted except for "linked value attributes". We assume that "only one of the updated values will be replicated" means only one set of values (after the update) will be replicated and that makes us believe that the other modification will be lost without notification to the application. Relevant text:

> *If two or more values in a multivalued attribute on an object are updated simultaneously on two different ADAM instances, only one of the updated values will be replicated. In other words, simultaneous updates to a multivalued attribute*

> *that occur on two different ADAM instances are considered in conflict, even if the updates apply to different values within the multivalued attribute. The only exception to this rule is for linked value attributes (such as group memberships), which do allow for simultaneous updates to different values within the linked value attribute.*

OpenLDAP presently has a similar resolution behavior on replication of multivalued attributes (consistency by entry not by value) as Multi-Master Replication does not use *delta syncrepl* where only changed values are replicated. When that's implemented, OpenLDAP will not have this behavior. Sun's JSDS does, however, replicate by value in multivalued attributes and we believe Red Hat (Fedora) DS does as well.

ADAM uses the Knowledge Consistency Checker that AD uses to construct the "most efficient topology for replication traffic". This is a technology quite critical to Active Directory as AD Replication topologies can get quite complex. Microsoft, shows some pretty complex topologies in documents like the *Active Directory  Replication Topology Technical Reference*, available at:

> *http://technet2.microsoft.com/windowsserver/en/library/1f3bb1c1-ba8a-4b4e-9f23-f240566e3d661033.mspx?mfr=true*

The complexity of these replication arrangements appears, based on the language used in the various technical documents, to be a reaction to slow and/or fragile networks. There are discussions of the use of SMTP protocols instead of RPC over such unreliable or slow links, for example. We conjecture that this replication is a symptom of the performance problems discussed elsewhere in this document.

### Management Limitations

ADAM requires port 135, described as an "RPC Endpoint Mapper", in order to be managed properly. This is a proprietary Microsoft protocol which is not fully documented. It is a known point of entry for multiple Windows worms, with a vulnerability profile that remains uncertain. OpenLDAP requires no proprietary ports for its management; it can be managed entirely over the standard LDAP ports.

ADAM does not provide a way to log all LDAP operations performed by the server, or for a viewable audit log that records the actual changes made.

ADAM does not support independent operational attribute retrieval (RFC 3673).

ADAM does not support the operational attributes *creatorsName*, *createTimeStamp*, *modifiersName*, *modifyTimeStamp* (RFC 4512).

While outside the LDAP standards, ADAM does not provide for LDAP-based querying of operational statistics. OpenLDAP provides *cn=monitor* which provides that capability. Other vendors offer various mechanism for on-line querying of operational statistics.

ADAM does not support per-user or per-group settings for search time limits or size limits. It only supports a global *timelimit/sizelimit* for the entire instance. Note that it is highly recommended that administrators never alter these settings. For more details, see follow up #8 in:

*http://www.openldap.org/its/index.cgi/Historical?id=5092*

That points you to an article on a Web site that explains and amplifies the recommendation that you never change *MaxPageSize*:

*http://searchwinit.techtarget.com/tip/0,289483,sid1_gci1265206,00.html*

The premise is that AD can become unresponsive, “flooding port 389” and causing a “self inflicted DOS (denial of service). This is another specific instance of AD being fragile in the face of higher loads that standard LDAP servers handle as a matter of course.

## Performance Limitations

### Indexing

ADAM has limited support for attribute indexing. It seems to be able to support equality indexing and a form of substring indexing. But it does not have any documented support for presence indexing, full substring indexing, or approximate (phonetic) indexing. The documentation mentions only one case of substring indexing: "[...]Tuple index for the attribute to improve medial searches -- this tuple index supports searches like *sn=*smith*, where the wildcard appears at the front of the search string". Common searches like *cn=jo*smith**, as performed through white pages lookup tools and other applications, are not optimized with these indexes.

### Caching

The details of ADAM's entry/attribute caching mechanism are not documented. System managers cannot make tradeoffs between the expected working set of data, server memory and the overall desired footprint.

## References

How ADAM works: http://technet2.microsoft.com/WindowsServer/en/library/7cfc8997-bab2-4770-aff2-be424fd03cda1033.mspx?mfr=true

FAQ: http://www.microsoft.com/windowsserver2003/adam/ADAMfaq.mspx

AD Schema reference: http://technet2.microsoft.com/windowsserver/en/library/97cae647-d996-48ff-b478-c96193abeadb1033.mspx?mfr=true

SANS Institute Internet Storm Center for Port 135: http://isc.sans.org/port.html?port=135

## Important capabilities missing from ADAM with reference to OpenLDAP

In general, in the earlier sections of this document, we have tried to compare AD and ADAM to the normal, standards-based, capabilities of the bulk of the LDAP directory services development community. There are experimental or special purpose LDAP Open Source packages which are not intended for or appropriate for enterprise use. Two excellent examples of these are *tinyldap* (a truly tiny read-only implementation) and *apacheDS* (currently an evolving test-bed for new concepts and functions for future considerations in the evolutions of the LDAP standards). While we believe OpenLDAP is the leading technology among the LDAP DS products, we stress the simple fact that **any** of the products from Open Source LDAP projects (OpenLDAP, Fedora DS, OpenDS) or from commercial vendors (CA, IBM, Isode, Novell, Oracle, Red Hat, Sun, or HP's Symas OpenLDAP, etc.) are, by virtue of their more sophisticated and flexible capabilities, superior to AD and ADAM for enterprise directories.

The subsections that follow, however, focus on OpenLDAP specific capabilities which may or may not be available in other LDAP offerings. Enterprises evaluating directory technologies should ask the providers of candidate technologies to provide similar supporting information for their own technologies.

### Database connectivity (back-ends)

LDAP provides access to data "stored" in the X.500 hierarchic data model and through the standardized LDAP operations and over LDAP's Internet protocol (TCP/IP) friendly protocol. There is much data stored in other LDAP directories or databases that an enterprise might want to stitch together under that model, through those operations, and over that protocol. One name for this is Virtual Directory. OpenLDAP created clean and efficient internal interfaces to allow data to be attached as part of the Directory Information Tree (or as separate databases) through flexible and easily developed "back-ends".

A back-end for SQL allows enterprises to make the data actively and efficiently managed by mission critical applications using RDBMSs appear to be entries in an LDAP directory. This opens up their use by LDAP-based applications. It also lets enterprises surface selected RDBMS data as objects in their unified enterprise directory.

OpenLDAP ships with numerous database back-ends. The *monitor* and *config* back-ends provide databases for operational (monitor) statistics and directory configuration, respectively. The *LDIF* back-end uses the LDAP Data Interchange Format (LDIF, the IETF standard for ASCII text representation of directory entries, see RFC 2849) to store directory entries. The *LDIF* back-end is used by the *config* back-end. Two back-ends are built on the Berkeley Database (BDB): the *bdb* back-end and the *hdb* back-end. These two are quite similar in many ways but *hdb* implements a true hierarchic database supporting subtree rename (the *modDN* operation that is required by LDAPv3 and no other LDAP software we know of supports). The *bdb* and *hdb* are the principal production database back-ends.

Other back-ends include the *ldap* backend providing access to other LDAP directories, the *shell* back-end that lets developers provide access to flat files or other proprietary data stores, the Perl back-end, and the *meta* back-end that provides functional meta-directory capabilities.

## Directory operational extensions (overlays)

Many times, enterprises want extensions or enhancements which are considered mandatory for their environment but which are not appropriate to provide in the core directory services package. Some of these extend data collection: audit or change logging. Others provide internal functions not consistent with the standard: attribute sorting, for example. Still others provide features consistent with but outside the specifications of the standard: translucency and referential integrity are examples. Unless the directory software provides appropriate interfaces into the operations of the directory itself, one has little choice but to modify the core of the directory's source code.

OpenLDAP introduced an Overlay capability into OpenLDAP in OpenLDAP 2.2. Overlays can be dynamically loaded and are entirely optional. Enterprises with no requirement for any of the overlay functions pay no administrative or performance overhead. Enterprises choosing to use an overlay merely provide additional administrative information, add the overlays to the configuration, and get immediate benefits.

Possibly more importantly, Enterprises (and ISVs) can independently develop unique and proprietary overlays to do what they want. They need no permission or assistance from OpenLDAP. There are no licensing considerations. Apple has decided to convert some of their proprietary modifications and extensions to Overlays for the version of Open Directory (OpenLDAP) shipped with Tiger. The Overlay versions install cleanly and eliminate the need to maintain patch files for a modified copy of OpenLDAP. It is a significant saving in complexity and, presumably, cost. This is an excellent example of the other benefits of the Overlay APIs.

## Replication

### Protocol Efficiency

Replication is used in directory architectures for two separate purposes: performance (scaling out) and for increased availability. The weaknesses of ADAM's replication is discussed above and compared in the general case of LDAP products. OpenLDAP introduced an innovative new replication technology in OpenLDAP 2.2 (synchronization replication or *syncrepl*) and extended that with "*delta syncrepl*" in OpenLDAP 2.3.

The original University of Michigan approach to replication was referred to as *slurpd*, a push-based approach. There were many problems associated with *slurpd*. The OpenLDAP project decided to move the responsibility for replication from the Master Directory Server to the Replica Directory Server. Now the replicas request synchronization either continuously (*refreshAndPersist*) or periodically (*refreshOnly*). This approach has proven to be extremely robust, efficient, and fast.

However, *syncrepl* still replicates by sending entire updated entries (objects) when changes were made. The *delta syncrepl* option reduces network traffic by sending only changed attributes to the replica when synchronization is requested. This increases the resource usage at the Master server but reduces the load on the network and improves overall performance.

Since the *syncrepl* capabilities have proven to be superior to *slurpd*, the project has decided to drop support for *slurpd* in OpenLDAP 2.4 (now in Beta).

### Partial and Fractional Replication

AD and ADAM replicate entire entries (objects). This means all the attributes of every entry appear in every replica. Often, there are attributes in entries which are inappropriate to replicate. First, they may simply not be needed by client programs accessing the replica servers. To replicate such data needlessly would be inefficient and cause the storage required by the replica to be much larger. Second, selected attributes might represent security problems demanding selective replication only to servers where there is a legitimate need for the data. These are extremely important considerations that neither of the Microsoft technologies take into account. ADAM's proponents would proliferate directories and create special subset directories but they would, in turn lead to the very directory bloat that centralization and unification are trying to eliminate. There is a better answer.

In OpenLDAP, replicas can be configured to support subsets of the available data in the Master server(s). The subset may be a subset by attributes as described above (known as fractional replication). It can also be a subset of the entries by some selection approach (known as partial replication). Both of these can dramatically reduce the resources required for a replica and/or offer dramatic performance improvements for replica-using applications on servers with limited capabilities.

## High Availability (Master Server soft Fail-over)

One of the biggest systems architecture concerns about directory servers is availability. A well integrated enterprise directory is mission critical to many applications and processes and outages are intolerable. For this reason, the Master Server stands out as a high-profile potential point of failure. OpenLDAP developed *mirrormode*, a soft fail-over capability to provide for very rapid resumption of service on failure of a Master directory server.

With *mirrormode*, a backup copy of the Master server's database is maintained (using standard *syncrepl* replication) on a machine appropriately configured and managed to be ready to take over the Master Server's workload. A software TCP/IP application load balancer/scheduler uses an application monitor to frequently verify that the Master Server is functioning properly. If the monitor indicates that the Master is down, the load balancer switches the traffic directed through it to the replica which then takes over as Master server. This can all happen in fractions of a second and with no human intervention. Client replica servers are generally configured to fail across to the designated backup Master and all continues without any particular fuss.

The backup Master server can be used to service incoming bind (login) and search requests while it is merely the backup. Steps may need to be taken to make sure load balancing takes this into account as the load of the now-defunct Master server may add too much traffic for preferred performance.

The *mirrormode* feature was developed by Symas during the development and availability of OpenLDAP 2.3. It was available in the OpenLDAP project's source code repository but not included in any OpenLDAP releases. The *mirrormode* capability will be released in OpenLDAP 2.4.

### Multi-Master Directories and Replication

In spite of a strong belief within the OpenLDAP Project and Symas Corporation that the Multi-Master technology introduced by the iPlanet partners (Netscape and Sun) was problematic, Symas developed a Multi-Master capability for OpenLDAP based on the *mirrormode* technology. Multi-Master directories have replication problems when different updates are made to the same data (attribute) in an entry at approximately the same time. There are detection and resolution algorithms that are well documented and understood but that do not eliminate the possibility that the collision can not be resolved and that the directory is, by some measure, incorrect. The Symas/OpenLDAP implementation significantly reduces the probability of such unresolved conflicts but does not eliminate all of the potential problems.

Multi-Master updating and replicating will be introduced in OpenLDAP 2.4 and is available in the current beta release (2.4.5) for evaluation and testing.

## Other Benefits of OpenLDAP: Access to Code and Support

One set of factors related to Open Source directory software packages separates them from the proprietary directory software packages: transparency and support.

Open Source is, by definition, software whose source code is readily available to anyone -- developers, users, researchers, whomever. That means that the software is entirely open to inspection. Nothing is hidden. Every feature and interface can be inspected. No secret back-doors or interfaces can be hidden from view.

Open Source software source code is actively inspected and reviewed by members of the project and the community at large. Many software experts are capable of analyzing a project's source code and verifying that there are no hidden, unwanted, traps. This transparency is important to enterprises wanting the certainty that they understand all the risks of adopting a package.

This is generally not possible with proprietary software. There are some vendors and restrictive access licenses that provide a look into proprietary software packages but this is not at all normal. This opacity leaves the enterprise with a decision of whether to trust the ISV's claims, a difficult situation at best. Since nobody but the ISV's people know what's in the code, even the ISV can be surprised by capabilities intended to be removed but left by accident or by individuals introducing capabilities for convenience or other reasons. In packages of the magnitude of today's platforms, this is a serious concern.

The second advantage of Open Source is freedom of access to support. Because the source code is readily available, it can be adopted by anyone with adequate skills and they can become proficient in development and support. Most projects have contributors with those skills from various companies and other affiliations and those contributors represent potential suppliers of support, for example.

OpenLDAP is developed by people in several languages on several continents. Most of them are quite competent to provide various levels of support. With OpenLDAP, Symas, the primary commercial support provider, has teamed with Hewlett-Packard to provide a very capable world-wide support organization. Symas has trained HP Technical Support teams that provide global support. HP offers normal business hours and 24x7 support in most countries and languages at competitive prices and Symas provides engineering (Level 3) support to HP in support of their offerings. It is a unique situation. Symas has no comparable agreement with any other global provider and that means HP has the only global support offering built around support from OpenLDAP's most prolific contributors.

This relationship was established within the HP Open Source and Linux Operation, a business unit dedicated to Open Source. It is a major corporate thrust and represents a new symbiosis between the traditional computing vendors and a key Open Source middleware project.