Realtime publishers

The Definitive Guide™ To

Active Directory Troubleshooting, Auditing, and Best Practices

2011 Edition

Don Jones

Copyright Statement

Chapter 7: Active Directory Lightweight Directory Services

In the Windows Server 2003 timeframe, Microsoft introduced Active Directory Application Mode, charmingly referred to as ADAM. These days, ADAM has grown up and changed his name to ADLDS (or AD LDS, if you prefer): Active Directory *Lightweight* Directory Services, which is distinct from the AD directory service that we're usually referring to when we just say "Active Directory." In this short chapter, we'll explore what AD LDS is all about, when you should (and shouldn't) use it, and how to perform basic troubleshooting and auditing with it.

What Is AD LDS?

Generally speaking, AD LDS is the same as regular AD in every way, except AD LDS doesn't perform authentication for your entire network. AD LDS is positioned as a "mode" of AD that provides directory service specifically for applications. Microsoft created AD LDS in part to address the reticence people have around extending the schema of their regular directory. Schema extensions are, after all, permanent, and nobody likes to make that kind of permanent extension to the main directory. What if you stop using the application after a few years? Its extensions hang around *forever.* So AD LDS gives applications a separate directory in which to store their "stuff."

AD LDS uses the exact same programming APIs as AD DS (Active Directory Domain Services, or the "normal" AD), so programmers don't have to take any special steps. AD LDS can operate entirely independently or it can operate with replication. Because it isn't part of your main domain, AD LDS also gives you a way of more easily and safely delegating control over applications' directory use. Someone can be in charge of an AD LDS install and have zero control over the main directory.

AD LDS does not, however, have any of the infrastructure components of AD DS. It isn't a directory service for the Windows operating system (OS), so clients can't authenticate to it. AD LDS *can* use your normal domain for authentication, which I'll discuss in a second. Thus, AD LDS can be a part of your domain in much the same way that any application could be. AD LDS doesn't have Flexible Single Master Operations (FSMO) roles or many of the other infrastructure elements we associate with the full AD DS. In addition, Microsoft Exchange can't utilize AD LDS because AD LDS doesn't support the Message Application Programming Interface (MAPI) or support authentication.

AD LDS *can* be run on a wider array of operating systems—the original ADAM, for example, ran fine on Windows XP. You can even run multiple instances of AD LDS on a single machine. An AD LDS instance isn't called a "domain controller" because the instance doesn't provide true domain controller functionality; instead, it is referred to as a "data store" or simply "AD LDS instance."

Partitions

AD LDS consists of a configuration and schema partition, much like AD DS. It also includes one or more application partitions, which is where applications store their data. Data, as in AD DS, is stored as objects, and the schema defines which object classes are available and what attributes those classes can use. Just as in AD DS, the configuration partition contains the internal configuration settings that make the system work.

When you install AD LDS, you have the option to create a unique instance or a replica of an existing instance, as Figure 7.1 shows. Replicas are how you provide scalability for AD LDS in instances where a single server can't keep up with the applications' demands. You can replicate the configuration and schema partitions of AD LDS, and select specific application partitions to replicate.

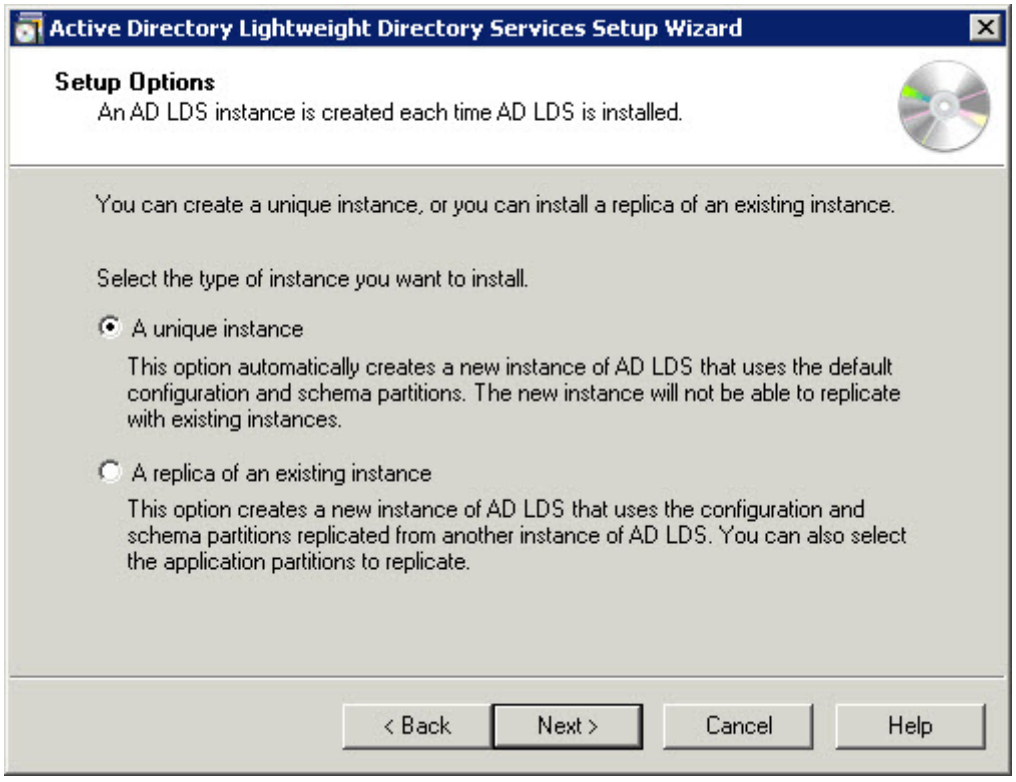

Figure 7.1: Creating a unique or replica AD LDS instance.

Synchronizing With AD DS

To synchronize AD LDS with a normal AD DS domain, you first have to export your directory's schema and load it into AD LDS. That way, AD LDS can "see" all of your normal domain's objects. AD LDS installs an AD Schema Analyzer tool, and you can use its Load Target Schema option (see Figure 7.2) to load the schema from an existing domain controller.

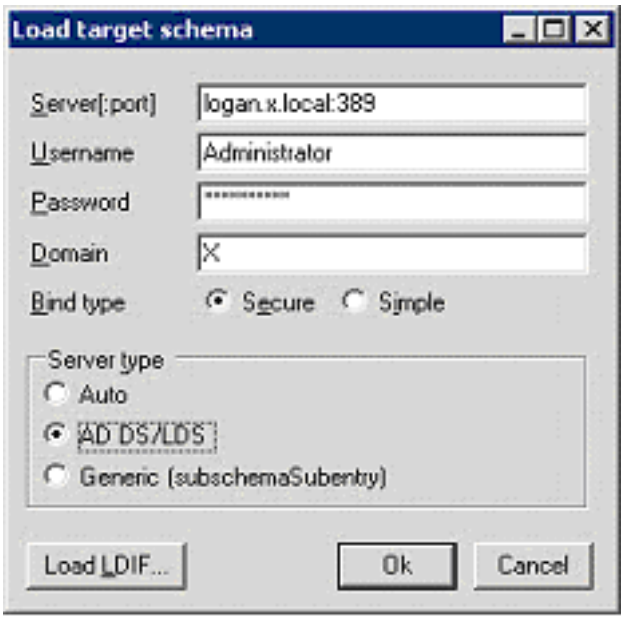

Figure 7.2: Loading the schema from a domain controller.

Resource

There are several other steps you'll need to take in order to make synchronization work; see the tutorial at http://www.thegeekispeak.com/archives/64 for a complete walkthrough.

Replication

AD LDS instances can replicate with each other. Just as in AD DS, replication in AD LDS provides both fault tolerance and load balancing for the services provided by AD LDS.

Before configuring replication, it's important to configure the AD LDS service to run under a user account. In addition, ensure that the computers hosting AD LDS are in the same (or trusted) domains. Each instance's service should be running under the same user account, not the built-in Network Service account.

AD LDS replicates data based on a *configuration set.* All AD LDS instances joined to the same configuration set will replicate a common configuration partition, a common schema partition, and whatever application partitions are configured in the configuration set. You can—very roughly—think of a configuration set as a domain from AD DS, meaning that all the AD LDS instances in the same configuration set will contain the same data. One trick is that an AD LDS instance can contain application partitions *beyond those in the configuration set.* Any application partitions in the configuration set will be shared with all instances replicating that set; any application partitions *outside* the configuration set will be unique to the instance where they live. Any AD LDS instance can participate in only one configuration set at a time, so if you have application partitions outside of a configuration set, those will *not* be replicated.

AD LDS supports the same kind of site and site link objects as AD DS, which are used to create and calculate the replication topology. I've written about replication earlier in this guide, and pretty much everything you know about AD replication—and sites and site links—applies to AD LDS as well. Replication *within* a site—that is, between instances on the same local area network (LAN)—is automatic and more or less real-time. Beyond setting up configuration sets to determine *what* will replicate, you don't have to do anything. Between sites, however, you *must* define site link objects—something that you don't have to do in AD DS. Intersite replication also requires you to set up the replication schedule, frequency, and availability—something you *can* do in AD DS, but which many admins don't manually configure.

Note

You can also override the automatic intrasite replication settings to specify a schedule, frequency, and so on.

Resource

Microsoft provides a complete guide to managing AD LDS replication, and configuration sets, at http://technet.microsoft.com/en-us/library/cc816770(WS.10).aspx.

Authentication

I technically lied about AD LDS not doing authentication. What it can't do is *authenticate a Windows computer* in the way that AD DS can. AD LDS can absolutely provide custom authentication for an application, and a lot of people use it as the directory for, say, an extranet Web application. Essentially, you're just using AD LDS to store custom user objects rather than sticking that information into a traditional relational database, which is what a lot of developers do. AD LDS is optimized for read access, making it a very quick and simple operation to look up a user, validate their password, and so forth.

You'll also see folks using AD LDS when they have an application that requires simple LDAP authentication and that wants to store data in the LDAP directory but they don't want that to be their main domain. AD LDS *does* support the full LDAP protocol, including authentication, so it can work well in that instance. The application would provide a user's X.500 Distinguished Name (DN) and password. AD LDS' security policy for password complexity, account lockout, and so forth are enforced by the local computer's security policy rather than a GPO (AD LDS doesn't do GPOs). However, if the computer is a member of a domain and a GPO applies to it that sets password complexity or other account policies, then those will obviously apply to AD LDS as well. Unfortunately, LDAP does transmit passwords in clear text if you aren't using LDAP over SSL, so be aware of that limitation.

AD LDS also supports Windows principal authentication, also known as SSPI authentication. This permits someone to use their AD DS domain account to authenticate to an AD LDS instance, or to use local user and group accounts created on the machine hosting AD LDS. To use domain accounts, AD LDS must be a member of the domain. In a domain environment, authentication happens with the Kerberos protocol, providing better security, mutual authentication, and complete protection of users' passwords (although it can fall back to NTLM authentication depending on your domain policies for that).

AD LDS also supports *proxy authentication,* also known as *bind redirection,* in which users authenticate using an AD LDS account (that is, a user account stored in AD LDS) but can use their AD DS domain password. Again, the AD LDS host computer needs to be a member of the AD DS domain, and you'll usually need some kind of account synchronization tool like ForeFront Identity Manager to synchronize the objectSID from AD DS to the corresponding AD LDS user accounts. This uses LDAP, so it's important to set up LDAP over SSL to secure the domain passwords on the network.

> **Resource**
>
> There is a great article at http://technet.microsoft.com/en-us/library/cc784622.aspx that explains these authentication options in some detail, including instructions for setting up the options.

When to Use AD LDS

AD LDS is useful whenever you have an application (other than Microsoft Exchange Server, which is a notable exception) that needs to store data in AD and you don't want to extend the schema of your main directory for that purpose. AD LDS is also a good choice if you're developing an application that will eventually integrate with AD DS. With AD LDS, you can have a *locally-installed* directory on your development or testing systems, because AD LDS can run on a broader range of OSs and doesn't have the extensive prerequisites of AD DS.

Anytime you find yourself asking, "Should we extend the schema of our directory?" then you should at least put AD LDS on the table for consideration, especially if your gut reaction to that question is, "NO!!!"

When Not to Use AD LDS

AD LDS is *not* a replacement for AD DS. It can't authenticate users to a domain, and it can't authenticate domain-joined computers. Windows machines can't "join" an AD LDS instance. AD LDS is intended for use primarily by applications, often in conjunction with a normal AD DS domain.

Troubleshooting AD LDS

The biggest thing you'll wind up troubleshooting in AD LDS is replication. Fortunately, its replication works *exactly like* that in AD DS, so the troubleshooting sections in the earlier chapters of this guide still apply.

Auditing AD LDS

AD LDS does support change auditing, meaning you can have an event written to the Windows event logs whenever a change occurs. These events often include old and new values for object attribute changes, which can be useful for creating an audit trail for compliance. It's the same feature as in AD DS, in fact, and you enable it in the same way.

> **Resource**
>
> The article at http://technet.microsoft.com/en-us/library/cc731764(WS.10).aspx has instructions for creating an audit trail for compliance. Although the article focuses on AD DS, the content applies to AD LDS as well.

As with password policy and account lockout, the audit policy can be applied to an AD LDS server either through its local security policy or for domain-joined computers through an appropriately-linked GPO. Auditing works just like it does in AD DS:

1. You'll typically enable auditing through a GPO, although for non-domain hosts you can do so in the local security policy.
2. Set the Security Access Control List (SACL) on the objects you want to audit.
3. The account running the AD LDS service needs to have the "Generate Security Audit" user privilege on the servers where AD LDS runs. NetworkService and LocalSystem have this set by default, but if you're replicating a configuration set and using a domain user account, then you'll have to grant this privilege to that account.

In addition to auditing attribute changes (which is fairly new even for AD DS), you can in AD LDS audit access to the directory service and audit logon events just as you can in AD DS. However, two specific settings that don't apply to AD LDS are:

- Audit Account Management—Because AD LDS objects are viewed by Windows as objects in a directory, Widows doesn't see them as "accounts" per se, even if the object's class name is "user" (and by default, AD LDS doesn't contain a "user" class).
- Audit Object Access, Audit Policy Change, Process Tracking, and System Events—The settings also don't make sense in AD LDS because they apply to things like files and policies that don't exist in AD LDS.

AD LDS doesn't come with a full suite of tools like AD DS does, although some of the normal AD DS tools will work against AD LDS. To set up a SACL, you'll use LDP.exe and its SACL editor. You can also use the Dsacls.exe command-line utility. Simply bind the tool to your AD LDS instance (make sure you're using an admin account to do so), enumerate your partitions, and right-click whatever object you want to apply a SACL to. As Figure 7.3 shows, you'll get a familiar-looking dialog box in which to define the audit policy for that object.

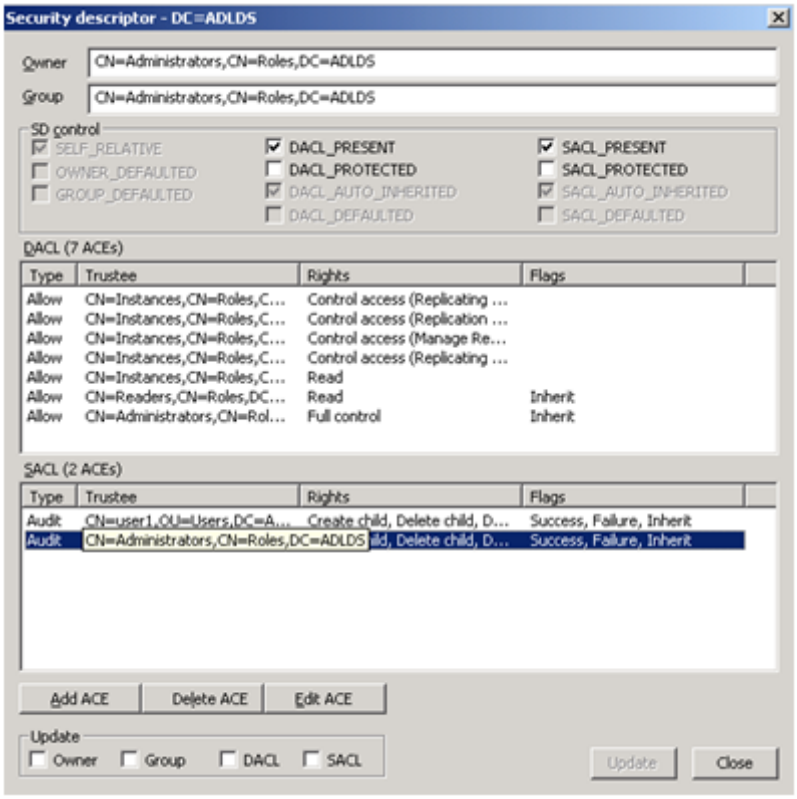

Figure 7.3: Setting a SACL in AD LDS.

Resource

You can find more information about setting up auditing, such as enabling auditing of replication events in AD LDS, at http://blogs.technet.com/b/askds/archive/2009/04/02/one-stop-audit-shop-for-adam-and-adlds.aspx. Be advised that a lot of this is pretty low-level, manual stuff because AD LDS doesn't come with the same high-level tools that you're used to with AD DS.

Coming Up Next

We're down to the final chapter in this guide, where I'll present assorted tips and tricks for AD. We'll cover things like FSMO roles, syncing, Kerberos, replication, DNS and trusts, permissions, communications, Group Policy, and much more.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.