Realtime publishers

The Definitive Guide™ To

Active Directory Troubleshooting, Auditing, and Best Practices

2011 Edition

Don Jones

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Chapter 1: A Non-Introduction to Active Directory

The world has been using Active Directory (AD) for more than a decade now, so there's probably little point in doing a traditional introduction for this book. However, there's still a bit of context that we should cover before we get started, and we should definitely think about AD's history as it applies to our topics of troubleshooting, auditing, and best practices.

The real point of this chapter is to identify key elements of AD that you need to completely inventory in your environment before proceeding in this book. Much of the material in the following chapters will refer to specific infrastructure elements, and will make recommendations based on specifics in common AD environments and scenarios. To make the most of those recommendations, you'll need to know the specifics of your own environment so that you know exactly which recommendations apply to you—and a complete, up-to-date inventory is the best way to gain that familiarity. To conclude this chapter, I'll briefly outline what's coming up in the chapters ahead.

A Brief AD History and Background

AD was introduced with Windows 2000 Server, and replaced the "NT Domain Services" (NTDS) that had been used since Windows NT 3.1. AD is Microsoft's first real directory; NTDS was pretty much just a flat user account database. AD was designed to be more scalable, more efficient, more standards-based, and more modern that its predecessor. However, AD was (and is) still built on the Windows operating system (OS), and as such shares some of the OS's particular patterns, technologies, eccentricities, and other characteristics.

AD also integrated a successor to Microsoft's then-nascent registry-based management tools. Known today as Group Policy, this new feature added significant roles to the directory beyond the normal one of authentication. With Group Policy, you can centrally define and assign literally thousands of configuration settings to Windows computers (and even non-Windows computers, with the right add-ins) belonging to the domain.

When AD was introduced, security auditing was something that relatively few companies worried about. Since 2000, numerous legislative and industry regulations throughout the world have made security and privacy auditing much more commonplace, although AD's native auditing capabilities have changed very little throughout that time. Because of its central role in authentication *and* configuration management, AD occupies a critical role for security operations, management, and review within organizations.

We also have to recognize that, outside from governing permissions on its own objects, AD doesn't play a central role in *authorization.* That is, permissions on things like files, folders, mailboxes, databases, and so forth aren't managed within AD. Instead, those permissions are managed at their point, meaning they're managed on your file servers, mail servers, database servers, and so forth. Those servers may assign permissions to identities that are authenticated by AD, but those servers control who actually has access to what. This division of labor between authentication and authorization makes for a highly-scalable, robust environment, but it also creates significant challenges when it comes to security management and auditing because there's no central place to control or review all of those permissions.

Over the past decade, we've learned a lot about how AD should be built and managed. Gone are the days when consultants routinely started a new forest by creating an empty root domain; also gone are the days when we believed the domain was the ultimate security boundary and that organizations would only ever have a single forest. In addition to covering troubleshooting and auditing, this book will present some of the current industry best practices around managing and architecting AD.

We've also learned that, although difficult to change, your AD design isn't necessarily permanent. Tools and techniques originally created to help migrate to AD are now used to restructure AD, in effect "migrating" to a new version of a domain as our businesses change, merge, and evolve. This book doesn't specifically focus on mergers and restructures, but keep in mind that those techniques (and tools to support them) are available if you decide that a directory restructure is the best way to proceed for your organization.

Inventorying Your AD

Before we get started, it's important that you have an up-to-date, accurate picture of what your directory looks like. This doesn't mean turning to the giant directory diagram that you probably have taped to the wall in your data center or server room, unless you've double-checked to make sure that thing is up-to-date and accurate! Throughout this book, I'll be referring to specific elements of your AD infrastructure, and in some cases, you might even want to consider implementing changes to that infrastructure. In order to best follow along, and make decisions, you'll want to have all of the following elements inventoried.

Forests and Trusts

Most organizations have realized that, given the power of the forest-level Enterprise Admins group, the AD forest is in fact the top-level security boundary. Many companies have multiple forests, simply because they have resources that can't all be under the direct control of a single group of administrators. However, to ensure the ability for users, with the appropriate permissions of course, to access resources across forests, cross-forest trusts are usually defined. Your first inventory should be to define the forests in your organization, determine who controls each forest, and document the trusts that exist between those forests.

Cross-forest trusts can be one-way, meaning that if Forest A trusts Forest B, the converse is not necessarily true unless a separate trust has been established so that Forest B explicitly trusts Forest A. Two-way trusts are also possible, meaning that Forest A and Forest B can trust each other through a single trust connection. Forest trusts are also non-transitive: If Forest A trusts Forest B, and Forest B trusts Forest C, then Forest A does not trust Forest C unless a separate, explicit trust is created directly between A and C.

When we talk about *trust,* we're saying that the *trusting* forest will accept user accounts from the *trusted* forest. That is, if Forest A trusts Forest B, then user accounts from Forest B can be assigned permissions on resources within Forest A. Forest trusts automatically include every domain within the forest so that if Forest A contains five domains, then every one of those domains would be able to assign permissions to user accounts from Forest B. Each forest consists of a root domain and may also include one or more child domains.

Figure 1.1 shows how you might document your forests. Key elements include meta directory synchronization links, forest trusts, and a general indication of what each forest is used for (such as for users or for resources).

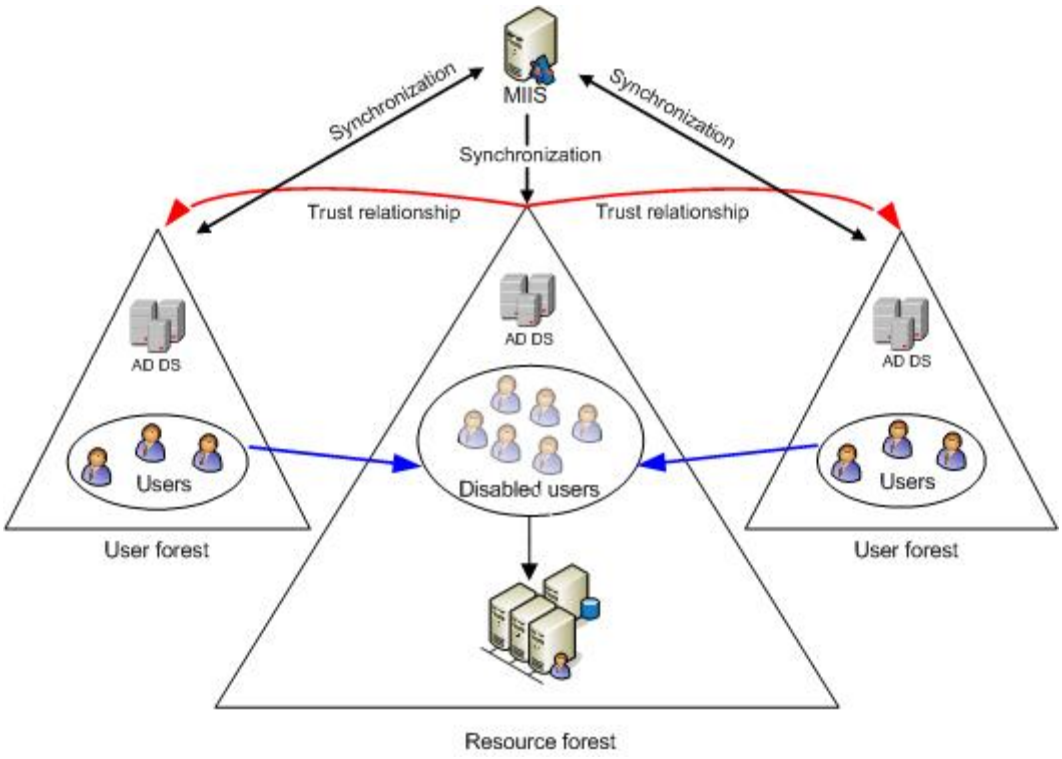

Figure 1.1: Documenting forests.

Note

For the various diagrams in this chapter, I'm going to draw from a variety of sources, including my past consulting engagements and Microsoft documentation. My purpose in doing so is to illustrate that these diagrams can take many different forms, at many different levels of complexity, and with many different levels of sophistication. Consider each of them, and produce your own diagrams using the best tools and skills you have.

Domains and Trusts

Domains act as a kind of security boundary. Although subject to the management of members of the Enterprise Admins group, and to a degree the Domain Admins of the forest root domain, domains are otherwise independently managed by their own Domain Admins group (or whatever group those permissions have been assigned or delegated to).

Account domains are those that have been configured to contain user accounts but which contain no resource servers such as file servers. *Resource domains* contain only resources such as file servers, and do not contain user accounts. Neither of these designations is strict, and neither exists within AD itself. For example, any resource domain will have at least a few administrator user accounts, user groups, and so forth. The type of domain designation is strictly a human convenience, used to organize domains in our minds. Many companies also use mixed domains, in which both user accounts and resources exist.

Domains are typically organized into a tree, beginning with the root domain and then through domains that are configured as children of the root. Domain names reflect this hierarchy: Company.com might be the name of a root domain, and West.Company.com, East.Company.com, and North.Company.com might be child domains. Within such a tree, all domains automatically establish a transitive parent-child two-way trust, effectively meaning that each domain trusts each other domain within the same tree.

Forests, as the name implies, can contain multiple domain trees. By default, the root of each tree has a two-way, transitive trust with the forest root domain (which is the root of the first tree created within that forest), effectively meaning that all domains within a forest trust each other. That's the main reason companies have multiple forests, because the full trust model within a forest gives top-level forest-wide control to the forest's Enterprise Admins group.

Even if you rely entirely on these default inter-domain trusts, it's still important to document them, along with the domains' names. Figure 1.2 shows how you might build a domain diagram in a program like Microsoft Office Visio. The emphasis in this diagram is on the logical domain structure.

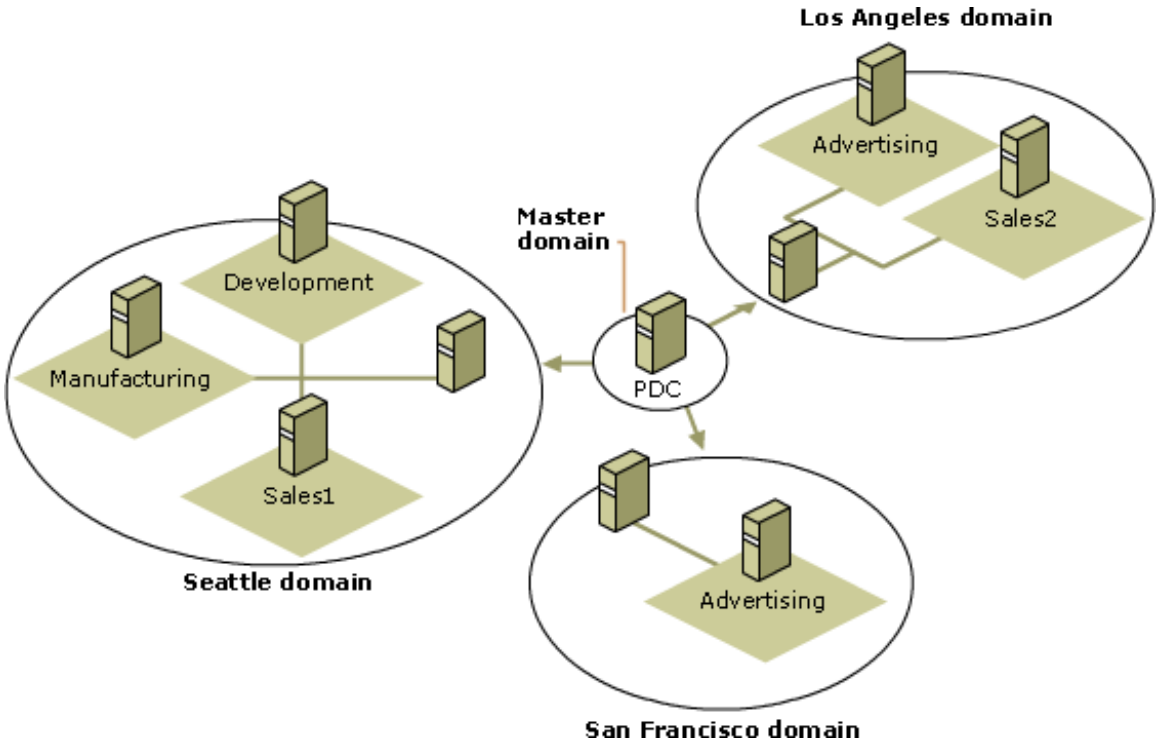

Figure 1.2: Documenting domains.

If you have any specialized domains—such as resource-only domains, user-only domains, and so forth—note those in your documentation. Also note the number of objects (especially computer and user accounts) in each domain. That is actually one of the most important metrics you can know about your domains, although many administrators can't immediately recall their numbers.

Domain Controllers

Domain controllers (DCs) are what make AD work. They're the servers that run AD's services, making the directory a reality. It's absolutely crucial, as you start reading this book, that you know how many DCs you have, where they're located, what domains they're in, and their individual IP addresses.

In many environments, DCs also provide other services, most frequently Domain Name Service (DNS). Other roles held by DCs may include WINS and DHCP services.

A DC's main role is to provide authentication services for domain users and for resources within the domain. We typically think of this authentication stuff as happening mainly when users show up for work in the morning—and in most cases, that *is* when the bulk of the authentication traffic occurs. However, as users attempt to access resources throughout the day, their computer will automatically contact a DC to obtain a Kerberos *ticket* for those resources. In other words, authentication traffic continues throughout the day—albeit at a somewhat slower, more evenly-distributed pace than the morning rush.

That morning rush can be significant: Each user's computer must contact a DC to log itself onto the domain, and then again when the user is ready to log on. Users almost always start the day with a few mapped drives, each of which may require a Kerberos ticket, and they usually fire up Outlook, requiring yet another ticket. Some of the organizations I've consulted with have each user interacting with a DC more than a dozen times each morning, and then several dozen more times throughout the day.

We tend to size our DCs for that morning rush, and that capacity generally sees us throughout the day—even if we take the odd DC offline mid-day for patching or other maintenance.

Each DC maintains a complete, read/write copy of the entire directory (the only exception being new-fangled *read-only domain controllers*—RODCs, which as the name implies, contain only a readable copy of the directory). Multi-master replication ensures that any change made on any DC will eventually propagate to every other DC in the domain. Replication is often one of the trickiest bits of AD, and is one of the things we tend to spend the most time monitoring and troubleshooting. Not all domain data is created equally: Some *high-priority* data, such as account lockouts, replicate almost immediately (or at least as quickly as possible), while less-critical information can take much longer to make its way throughout the organization.

Figure 1.3 shows what a DC inventory might look like. Note the emphasis on physical details: IP addresses, DNS configuration, domain membership, and so forth.

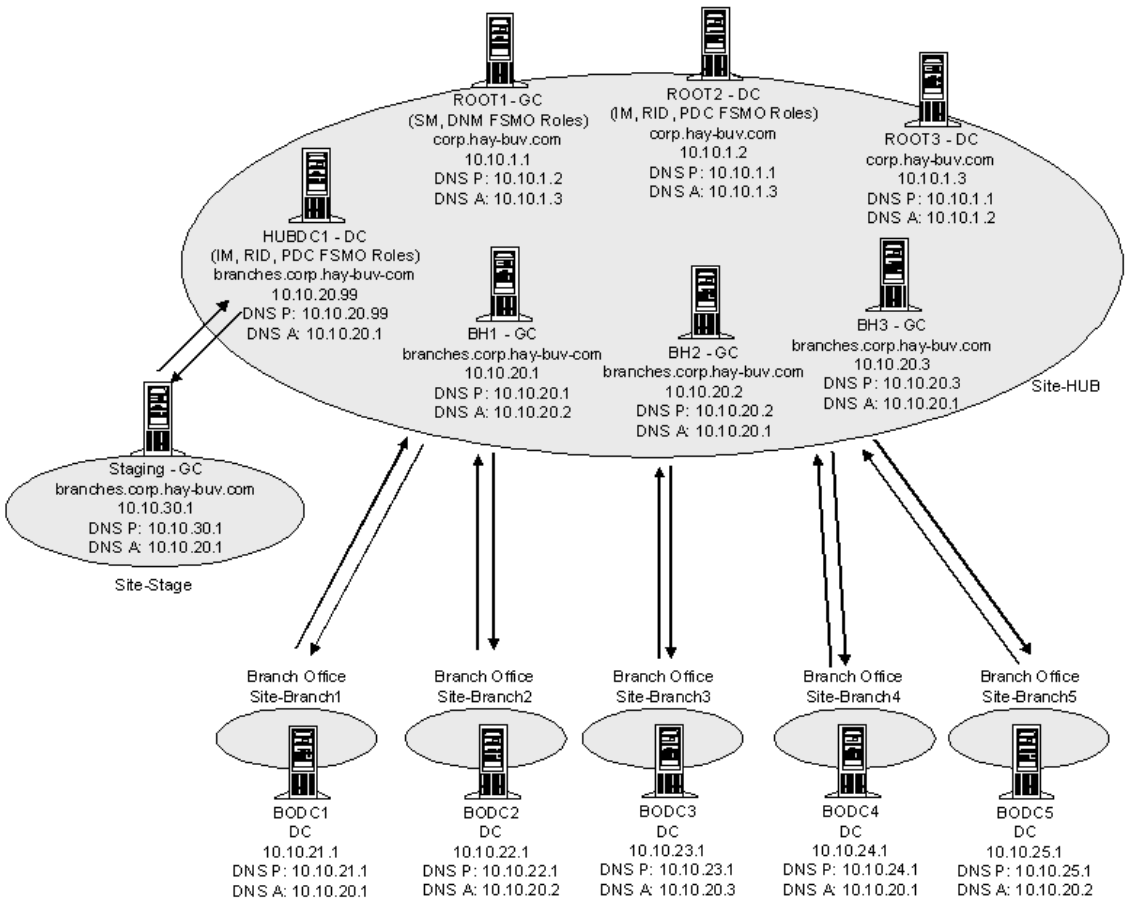

Figure 1.3: DC inventory.

It's also important to note whether any of your DCs are performing any non-AD-related tasks, such as hosting a SQL Server instance (which isn't recommended), running IIS, and so forth.

Global Catalogs

A global catalog (GC) is a specific service that can be offered by a DC in addition to its usual DC duties. The GC contains a subset of information about every object in an entire forest, and enables users in each domain to discover information from other domains in the same forest. Each domain needs at least one GC; however, given the popularity of Exchange Server and its heavy dependence on GCs (Outlook, for example, relies on GCs to do email address resolution), it's not unusual to see a majority, or even all, DCs in a domain configured as GC servers.

Make sure you know exactly where your GCs are located. Numerous network operations can be hindered by a paucity of GCs, but having too many GCs can significantly increase the replication burden on your network.

> **Note**
>
> In Figure 1.3, "GC" is used to indicate DCs that are also hosting the GC server role.

FSMOs

Certain operations within a domain, and within a forest, need a single DC to be in charge. It is absolutely essential for most troubleshooting processes that you know where these Flexible Single Master of Operation (FSMO) role-holders sit within your infrastructure:

- The RID Master is in charge of handing out Relative IDs (RIDs) within a single domain (and so you'll have one RID Master per domain). RIDs are used to uniquely identify new AD objects, and they are assigned in batches to DCs. If a DC runs out of RIDs and can't get more, that DC can't create new objects. It's common to put the RID Master role on a DC that's used by administrators to create new accounts so that *that* DC will always be able to request RIDs.
- The Infrastructure Master maintains security identifiers for objects referenced in other domains—typically, that means updating user and group links. You have one of these per domain.
- The PDC Emulator provides backward-compatibility with the old NTDS, and is the only place where NTDS-style changes can be made (any DC provides read access for NTDS clients). Given that NTDS clients are becoming extinct in most organizations, the PDC Emulator (you'll have one in each of your domains, by the way) doesn't get used a lot for that purpose. Fortunately, it has a few other things to keep it busy. For example, password changes processed by other DCs tend to replicate to the PDC Emulator first, and the PDC Emulator serves as the authoritative time source for time synchronization within a domain.
- Each forest will contain a single Schema Master, which is responsible for handling schema modifications for the forest.
- Each forest also has a Domain Naming Master, which keeps track of the domains in the forest, and which is required when adding or removing domains to or from the forest. The Domain Naming Master also plays a role in maintaining group membership across the forest.

Marking these role owners on your main diagram (such as Figure 1.3) is a great way to document the FSMO locations. Some organizations also like to indicate a "backup" DC for each FSMO role so that in the event a FSMO role must be moved, it's clear where it should be moved to.

Containers

The logical structure of AD is divided into a set of hierarchical containers. AD supports two main types: *containers* and *organizational units* (OUs). A couple of built-in containers (such as the Users container) exist by default within a domain, and you can create all the OUs that you want to help organize your domain's objects and resources. Again, an inventory here is critical, as several operations—most especially Group Policy application—work primarily based on things like OU membership.

Figure 1.4 shows one way in which you might document your OU and container hierarchy. Depending on the size and depth of your hierarchy, you could also just grab a screenshot from a program like Active Directory Users and Computers.

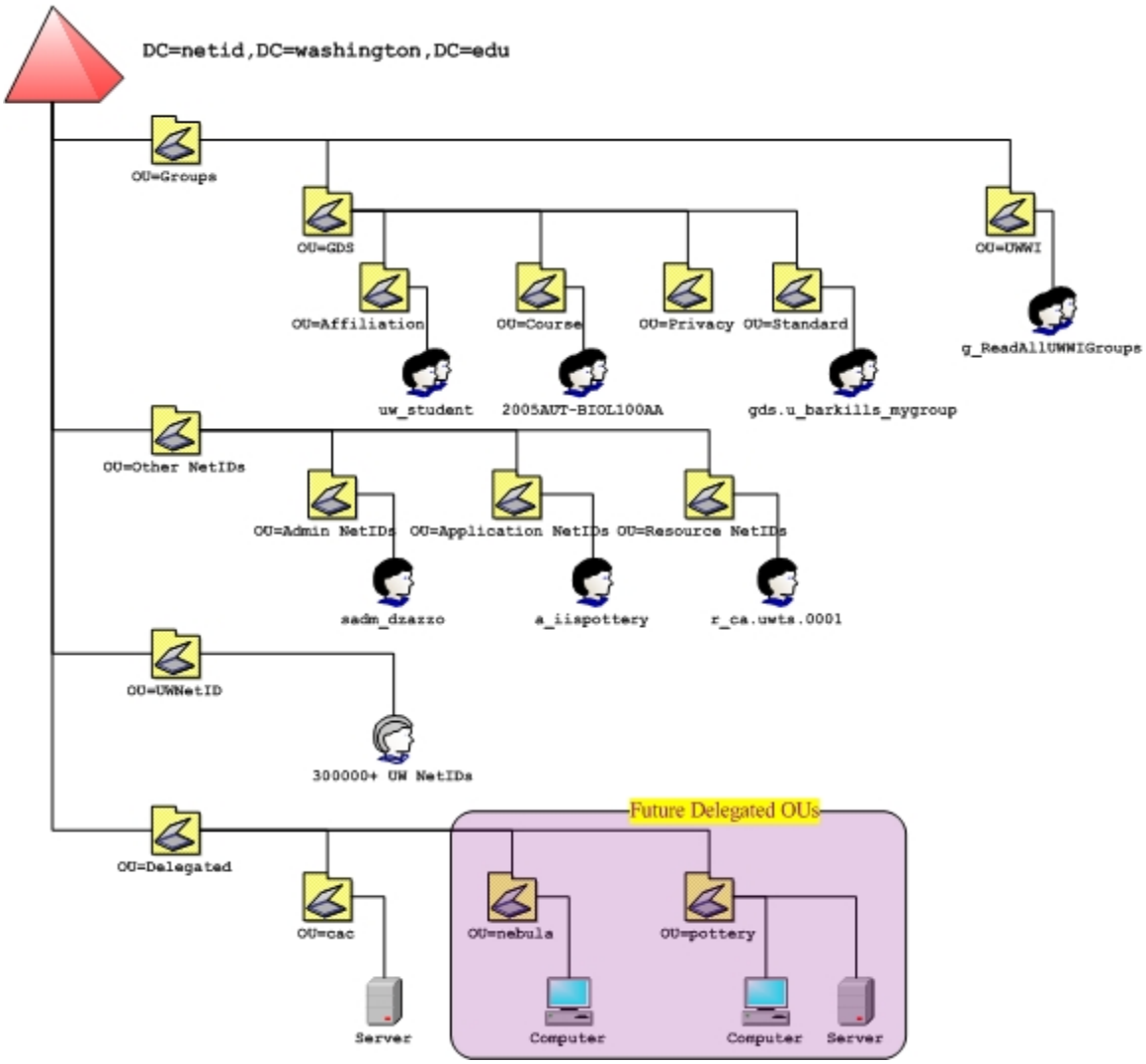

Figure 1.4: Documenting OUs and containers.

Try to make some notation of how many objects are in each container, and if possible make a note of which containers have which Group Policy Objects (GPOs) linked to them. That information will be useful as we dive into troubleshooting and best practices discussions.

Subnets, Sites, and Links

In AD terms, a *subnet* is an entry in the directory that defines a single network subnet, such as 192.168.1.0/8. A *site* is a collection of subnets that all share local area network (LAN)-style connectivity, typically 100Mbps or faster. In other words, a site consists of all the subnets in a given geographic location.

Links, or *site links,* define the physical or logical connectivity between sites. These tell AD's replication algorithms which DCs are able to physically communicate across wide area network (WAN) links so that replicated data can make its way throughout the organization. Documenting your subnets, sites, and links is quite probably the most important inventory you can have for a geographically-dispersed domain.

Typically, you'll have site links that represent the physical WAN connectivity between sites. A *cost* can be applied to each link, indicating its relative expense. For example, if two sites are connected by a high-speed WAN link and a lower-speed backup link, the backup link might be given a higher cost to discourage its use by AD under normal conditions. As Figure 1.5 shows, you can also create site links that represent a virtual connection. The A-C link connects two sites that do not have direct WAN connectivity. This isn't necessarily a best practice, as it tells AD to expect WAN connectivity where none in fact exists.

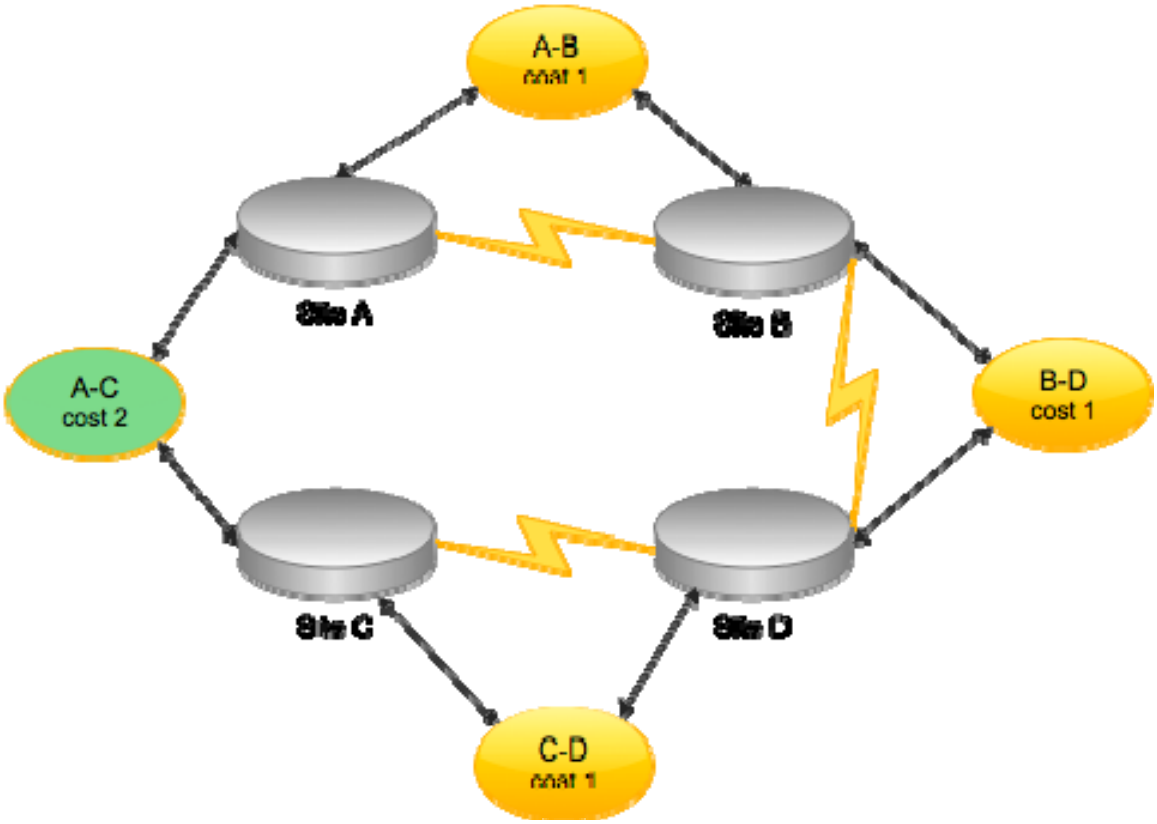

Figure 1.5: Configuring site links.

Eliminating the A-C site link will not hinder AD operations: The directory will correctly determine the best path for replication. For example, changes made in Site C would replicate to D, then to B, and eventually to A. If Site C were the source of many changes (perhaps a concentration of administrators work there), you could speed up replication from there to Site A by creating a *site link bridge*, effectively informing AD of the complete path from C to A by leveraging the existing A-B, B-D, and C-D site links. Such a bridge accurately reflects the physical WAN topology but provides a higher-priority route from C to A. Figure 1.6 shows how you might document that.

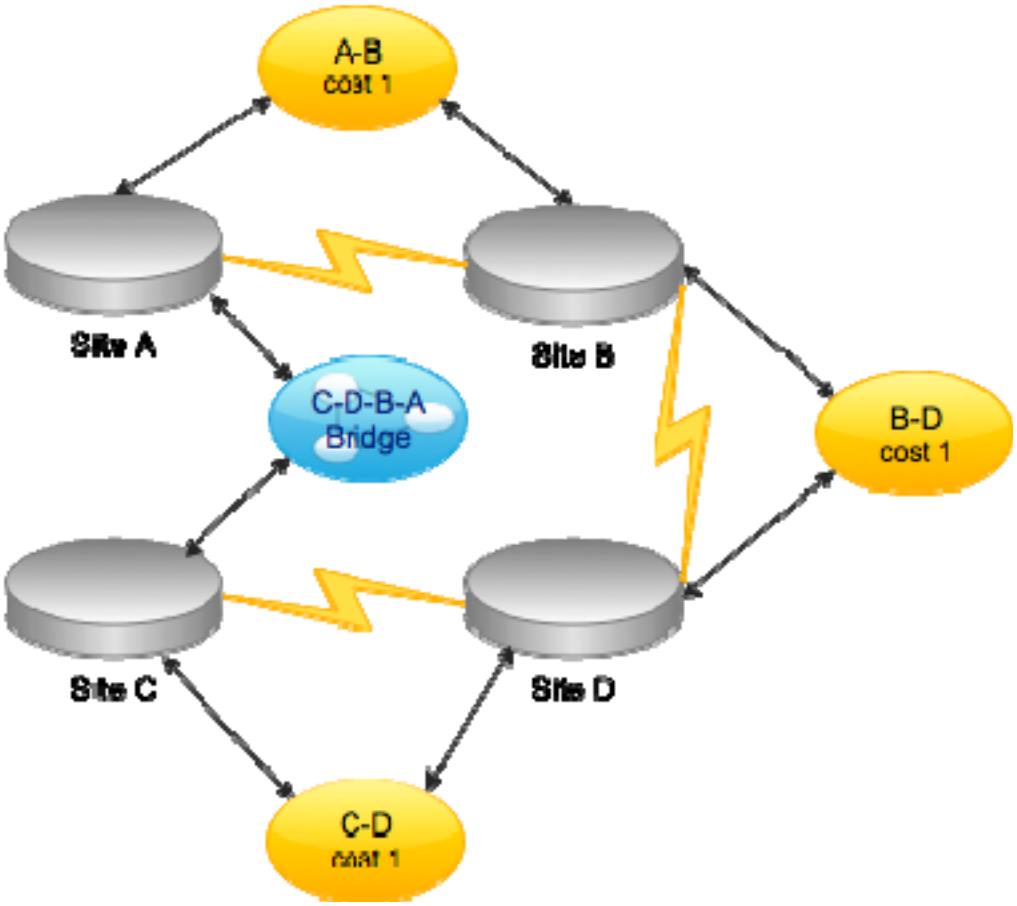

Figure 1.6: Configuring a site link bridge.

As you document your sites, think again about numbers: How many computers are in each site? How many users? Make a notation of these numbers, along with a notation of how many DCs exist at each site.

Sites should, as much as possible, reflect the *physical* reality of your network; they don't correspond to the *logical* structure of the domain in any way. One site may contain DCs from several domains or forests, and any given domain may easily span multiple sites. However, *site links* are kind of a part of the domain's logical structure because those links are defined within the directory itself. If you have multiple domains, it's worth building a diagram (like Figure 1.5 or 1.6) for each domain—even if they look substantially the same. In fact, any group of domains that spans the same physical sites *should* have identical-looking site diagrams because the physical reality of your network isn't changing. Going through the exercise of creating the diagrams will help *ensure* that each domain has its links and bridges configured properly.

DNS

The last critical piece of your inventory consists of your DNS servers. You should clearly document where each server physically sits and think about which clients it serves. Most companies have at least two DNS servers, although having more (and distributing them throughout your network) can provide better DNS performance to distant clients. AD absolutely cannot function without DNS, so it's important that both servers and clients have ready access to a high-performance DNS server. *Most AD problems are rooted in DNS issues,* meaning much of our troubleshooting discussion will be about DNS, and that discussion will be more meaningful if you can quickly locate your DNS servers on your network.

Also try to make some notation of which users, and how many users, utilize each DNS server either as a primary, secondary, or other server. That will help give you an at-a-glance view of each DNS server's workload, and give you an idea of which users are relying on a particular server.

Putting Your Inventory into Visual Form

A tool like Microsoft Office Visio is often utilized to create AD infrastructure diagrams, often showing both the logical structure (domains, forests, and trusts) and the physical topology (subnets, sites, links, and so forth). There are also third-party tools that can automatically discover your infrastructure elements and create the appropriate charts and diagrams for you. The benefit of such tools is that they're always right because they're reflecting reality—not someone's memory of reality. They can usually catch changes and create updated diagrams much faster and more accurately than you can.

I love to use those kinds of tools in combination with my own hand-drawn diagrams. If the tool-generated picture of my topology doesn't match my own picture, I know I've got a problem, and that can trigger an investigation and a change, if needed.

What's Ahead

Let's wrap up this brief introduction with a look at what's coming up in the next seven chapters.

AD Troubleshooting

Chapters 2 and 3 will concern themselves primarily with troubleshooting. In Chapter 2, we'll focus on the ways and means of monitoring AD, including native event logs, system tools, command-line tools, network monitors, and more. I'll also present desirable capabilities available in third-party tools (both free and commercial), with a goal of helping you to build a sort of "shopping list" of features that may support troubleshooting, security, auditing, and other needs.

Chapter 3 will focus on troubleshooting, including techniques for narrowing the problem domain, addressing network issues, resolving name resolution problems, dealing with AD service issues, and more. We'll also look at replication, AD database failures, Group Policy issues, and even some of the things that can go wrong with Kerberos. I'll present this information in the form of a troubleshooting flowchart that was developed by a leading AD Most Valuable Professional (MVP) award recipient, and walk you through the tools and tasks necessary to troubleshoot each kind of problem.

I'll wrap up this book with more troubleshooting, devoting Chapter 8 to additional troubleshooting tips and tricks.

AD Security

In Chapter 4, we'll dive into and discuss the base architecture for AD security. We'll look more at the issue of distributed permissions management, and discuss some of the problems that it presents—and some of the advantages it offers. We'll look at some do-it-yourself tools for centralizing permissions changes and reporting, and explore whether you should rethink your AD security design. We'll also look at third-party capabilities that can make security management easier, and dive into the little-understood topic of DNS security.

AD Auditing

Chapter 5 will cover auditing, discussing AD's native auditing architecture and looking at how well that architecture helps to meet modern auditing requirements. I'll also present capabilities that are offered by third-party tools and how well those can meet today's business requirements and goals.

AD Best Practices

Chapter 6 will be a roundup of best practices for AD, including a quick look at whether you should reconsider your current AD domain and forest design (and, if you do, how you can migrate to that new design with minimum risk and effort). We'll also look at best practices for disaster recovery, restoration, security, replication, FSMO placement, DNS design, and more. I'll present new ideas for virtualizing your AD infrastructure, and look at best practices for ongoing maintenance.

AD LDS

Chapter 7 gives me an opportunity to cover additional information: AD's smaller cousin, Active Directory Lightweight Directory Services (AD LDS). We'll look at what it is, when to use it, when *not* to use it, and how to troubleshoot and audit this valuable service.

Let's Get Started!

With your AD inventory updated and in-hand, we're ready to begin. The next chapter will introduce you to the majority of the tools that you'll need to pry valuable information out of AD so that you can start assembling your security and troubleshooting utility belt.

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.