Realtime publishers

The Definitive Guide™ To

Active Directory Troubleshooting, Auditing, and Best Practices

2011 Edition

Don Jones

Copyright Statement

Chapter 6: Active Directory Best Practices

This chapter is a kind of "miscellaneous best practices" list. The trick with AD and best practices is that there's never any one right answer for every organization. You have to temper everything with what's right for *your* organization. So really, this chapter is intended to simply give you things to think about within your environment, and ideas that stem from what's worked well for other folks in situations that might be similar to your own.

Should You Rethink Your Forest and Domain Design?

First of all, step back and take a look at your domain and forest design. How perfect is it? AD design unfortunately has two conflicting goals: One is to support your Group Policy deployment, and the other is to support delegation of permissions. For the first goal, you might organize AD to really facilitate using a minimal number of effective Group Policy Objects (GPOs), especially if you need differing GPO settings for various company departments and divisions. The second goal focuses on who will manage AD objects: If you plan to delegate permissions to reset passwords, for example, then organizing your directory to group those delegated user objects will make the actual delegation easier to set up and maintain.

Keep in mind that Group Policy is the one thing you pretty much can't separate from the directory. From a security and delegation perspective, third-party tools *can* abstract your directory design. For example, many third-party identity and access management (IAM) tools enable you to delegate permission over objects that are distributed throughout the directory. You essentially use the tool to manage the delegation, and it deals with whatever ugly, under-the-hood permissions it needs to. In some cases, these tools don't actually modify the underlying directory permissions at all. Instead, they provide "in-tool" delegation, meaning they act as a kind of proxy manager, providing different user interfaces for delegated users to accomplish tasks like resetting passwords or modifying user accounts. That kind of abstraction can let your underlying directory structure conform to other needs—like those of you Group Policy deployment.

Restructuring a domain or forest can be just as complex, risky, and frustrating as migrating to AD was in the first place. The main reason to consider this kind of project is if your directory has grown, and been extended, organically over time. Corporate merges and acquisitions are a common root cause of that kind of growth. You may also find that whoever originally designed the directory didn't have a good understanding of how to do so, or that the company's needs and operations have changed since the original design was put in place. In any event, rethinking the design can have a significant positive impact on operations, maintenance, disaster recovery, and even on performance and usability—so it's worth at least *considering* the project. Determine whether the business benefits would outweigh the potential risks, and consider ways to mitigate those risks. For example, many third parties produce migration/restructuring tools that can largely automate much of the process, provide zero-impact testing capabilities, and even roll back migration changes if they prove to be problematic. Those tools obviously have a cost, so you'll have to weigh that cost against the business benefits and see if it looks like a win.

AD Disaster Recovery

Disaster recovery and business continuity is always a concern, so let's look at general best practices for making sure that your directory can be recovered in the event of a failure. We're not going to look at the more commonly-needed single-object recovery just yet—there's a section in this chapter for that coming up.

Single Domain Controller

Probably the most common failure scenario in AD is the failure of a single domain controller, often due to a hardware failure. What do you do when this happens? Well, if you've built your domain controllers properly, you won't need to do much. My assumption is that your domain controllers are doing very little apart from being domain controllers. They may be running DNS, and if they are it should be an AD-integrated DNS zone. If you don't use Microsoft's DNS, don't put your DNS servers on your domain controllers. That way, if a domain controller fails, you just rebuild it.

Keep in mind that, in AD, no domain controller is unique. They're all the same. If one fails, it's no big deal—the others just keep moving right along. Build a replacement machine (something that's trivial if you're using virtual machines), promote it to be a domain controller, and sit back and let replication take over. In other words, you don't bother backing up every single domain controller because they each act as backups for each other.

The only time this might not be a straightforward approach is when the failed domain controller is on the other side of a slow WAN link from any other domain controllers. Waiting for a large domain to replicate across the WAN can be time consuming. If you don't mind waiting, it's still the best way to go. About the only other option is to keep a backup of those remote domain controllers—making sure it's never more than a few days old. That way you can restore from that backup, and let a much lesser amount of replication bring the domain controller back up to date. Tape backups are fine for this approach, and they're easy for people with minimal IT skills to operate, so in cases where you don't have a lot of local expertise helping you out, it's not a bad approach.

You'll often see smaller remote offices using an "all in one" server—a single machine acting as domain controller, DNS, DHCP, file server, print server, fax server, and who knows what else. Try to avoid that: In this day and age, that physical machine should be a virtualization host, with some of those roles split up between different machines. Either way, tape-based backup can start to become complex and large, and I recommend moving to a real-time, disk-based backup. That'll get the server back online quicker in the event of a failure, and it'll do a better job of capturing all the data that the server houses.

Entire Domain

It's pretty rare to lose an entire domain. As it's almost impossible to lose every single domain controller at the same time, "losing" the domain usually means some vast and tragic administrator error. The only resolution is, of course, to have a good—and recent—backup.

Again, this is where I firmly reject tape-based backup and recommend real-time disk-based backups instead (read my book, *The Definitive Guide to Windows Application and Server Backup 2.0*, from Realtime Publishers, for an exhaustive treatment of the subject). A real-time disk-based backup can get a domain controller up and running in minutes or hours, not days, and you'll lose no more than a few minutes' worth of activity from the domain.

Disk-based backups can also (usually, depending on the vendor) be replicated off-site, making them suitable for true disaster recovery where you've lost an entire data center, or lost the use of it, due to some disaster such as flood, fire, meteor strikes, and the like.

Entire Forest

It is *vanishingly* rare to lose an entire AD forest. I was once told that there are something like less than a dozen documented, real-world (that is, non-lab-based) occurrences. Still, the threat of whole-forest-loss is enough that Microsoft officially supports forest recovery, and a handful of third-party vendors make whole-forest recovery products.

If you feel that losing your entire AD forest is a threat you must be prepared to face, take my advice and buy a forest recovery product *now* (they're no good once the forest has actually failed; they have to grab the necessary backups first). Recovering a forest is no trivial task, and having a tool on-hand will get you back up and running more quickly than the alternative, which is usually contacting Microsoft product support for assistance.

AD Restores and Recycle Bins

Let's turn briefly to the subject of single-object recovery within AD. Prior to Windows Server 2008 R2, Microsoft didn't have a good, supported solution for AD single-object recovery. Their approach was to take a domain controller offline, put it in Directory Services Recovery Mode, perform an authoritative restore of whatever directory object(s) you lost, then bring the domain controller back online and let it replicate its changes.

Let's be clear on what I mean by *single-object recovery,* too: Bringing an entire deleted object back, including all of its attributes. You *cannot* do this by simply un-tombstoning a deleted object because when AD deletes and tombstones an object, it removes the object's attributes.

In Windows Server 2008 R2, Microsoft introduced a feature called the "Active Directory Recycle Bin," a name of which I am not a fan. This feature is only available when the entire forest is running at the Win2008R2 functional level (meaning every domain must also be running at this level), and the feature must be specifically turned on—a one-time action that can't be undone. Figure 6.1 shows the PowerShell command needed to enable the feature.

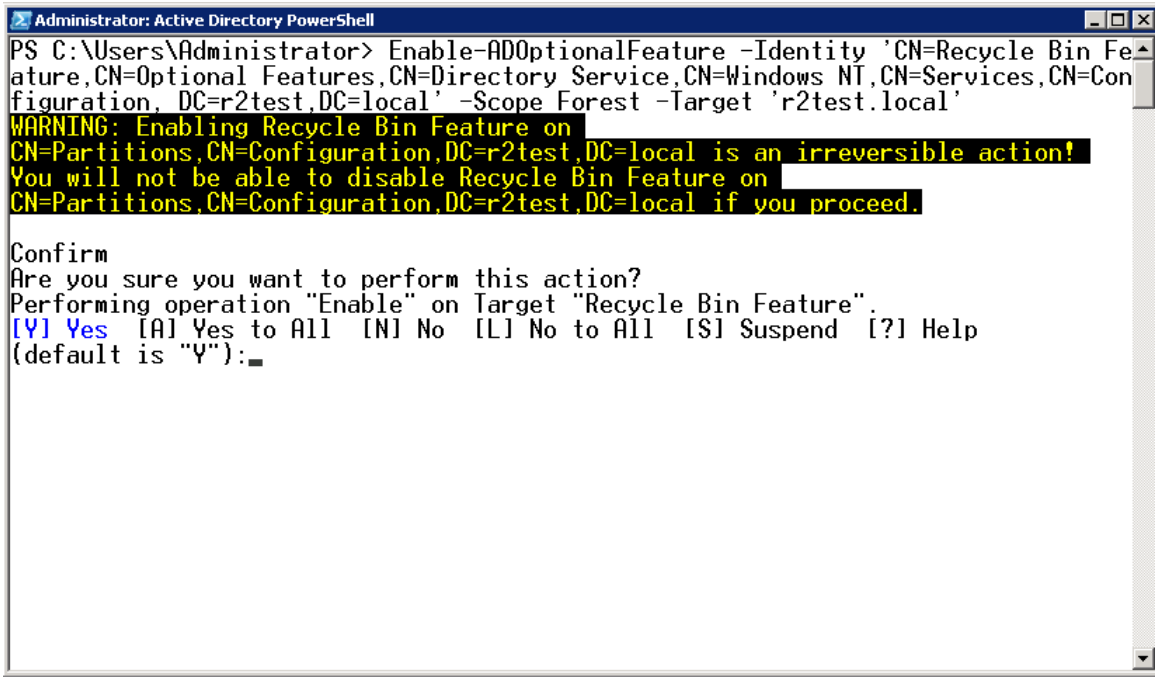

Figure 6.1: Enabling the "Recycle Bin" feature.

When on, deleted objects are copied—attributes intact—into a "Recycle Bin" container within the directory. Only you won't actually see a Recycle Bin icon, and you can't drag objects out of the "bin" back into the main directory (that lack of actual "Recycle Bin" functionality is why I wish they hadn't called it that). As Figure 6.2 shows, you can use GUI tools to view the new "Deleted Objects" container and its contents.

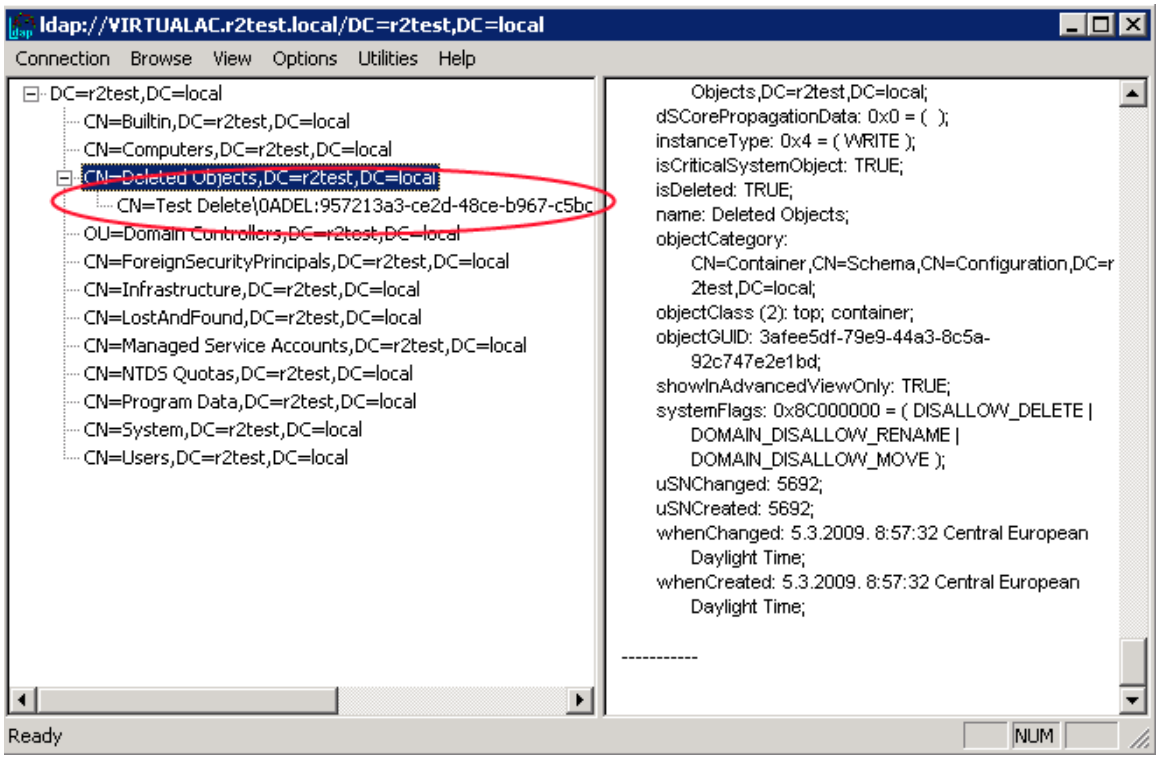

Figure 6.2: Viewing the Deleted Objects container.

To actually restore an object requires the use of rather byzantine Windows PowerShell commands; there's no actual GUI component for working with "recycled" AD objects.

The "Recycle Bin" feature is also a bit unintuitive. For example, if you need to restore an OU and its contents, it's a two-step process: Restore the OU, then the objects that used to live in it. Some organizations will have concerns about that recycled information—including employees' personally-identifiable information (PII)—persisting in the directory past the objects' deletion. Although a traditional backup would also persist that information, it doesn't do so "live" in the directory, and that makes a difference to some folks.

The "Recycle Bin" feature is also limited to *object* restoration; it can't restore a single *attribute* from an object that may have been improperly changed.

So this new "Recycle Bin" feature is, at best, a bare-bones way of getting single-object recovery for a very small organization that *will not* consider third-party tools. Me, I'm a fan of third-party tools. A single AD disaster recovery solution can give you a true, graphical recycle bin with drag-and-drop recovery and single-attribute recovery and will scale all the way up to complete domain or forest recovery if necessary. Everything but a domain/forest restore can be done without taking a domain controller offline, helping everything stay productive, and in most cases, these tools integrate into the familiar Active Directory Users and Computers console, making them even easier and more accessible.

You could argue that Microsoft should build that kind of functionality into the base product. Maybe so, maybe no: Every third-party recovery tool I've looked at works slightly differently, and those differences reflect different customer needs. Microsoft would only be able to squeeze us all into the same functionality; as the situation stands, we can select from whatever solution fits our particular needs the best. Microsoft, as I've suggested in earlier chapters, needs to deliver a good platform—I don't necessarily think they should deliver every possible permutation of a management tool that an organization might need.

This Isn't Retail

I've made this argument about third-party tools before. Too often, I see a "packaged retail" mentality around computer software. You go and buy Microsoft Office, you don't expect to have to buy add-ons to make it work. Okay, I get that—Office is an end-user product. Most end-user products come complete: Cars come complete. Even kids' games sometimes ship with batteries included.

Windows, as a server operating system (OS), isn't a packaged retail end-user product. It's more like a house: The builder is giving you a platform, and you *expect* to spend money above and beyond that structure. The structure should come with good plumbing, but you attach your own faucets. The floors should be flat and solid, but you're putting your own furniture on them.

Yes, some builders will throw in minimal versions of these add-ons—kitchen appliances, bathroom fixtures, and so forth. But these are almost always the bare-minimum versions. They're rarely the high-end, custom stuff you know you want.

Sure, you *can* buy a house that comes with all the custom high-end stuff, but that's like working with a Microsoft VAR. In addition to the home builder (Microsoft), you've also got a designer (the VAR) buying your curtains, furniture, and so forth, and giving you the resulting product for a single package price. You *can* do that with Windows: Get the base platform and all the third-party tools needed to make it awesome, all from one vendor, and all for one price. That vendor just isn't Microsoft, because they're in the business of making the basic structure, not customizing it to fit every possible business need.

When it comes to Windows as a server OS, you *have* to include certain third-party tools as part of the cost of doing business. The cost for the Windows license is just the beginning: If you have auditing needs, or disaster recovery needs, those are going to cost extra. If you're in the type of company that doesn't like to spend money on "extras" anytime, ever, then you shouldn't expect to be able to meet all of the business' needs all of the time, either.

Security

I don't actually have a lot to say on the topic of security best practices. I think Microsoft's Best Practices Analyzer (BPA—which will be discussed in the final section of this chapter) does a good job of covering the high-level security settings in AD; anything else really comes down to your specific business and operational needs. Do you delegate permissions within the directory or rely on a more monolithic permissions structure where Domain Admins do all of the work? Neither approach is wrong; it simply depends on how your organization is structured for that kind of administration.

Replication Topology

Definitely take the time, now and then, to review your AD replication topology. Using your site architecture, draw out a picture of the replication topology, like the one in Figure 6.3.

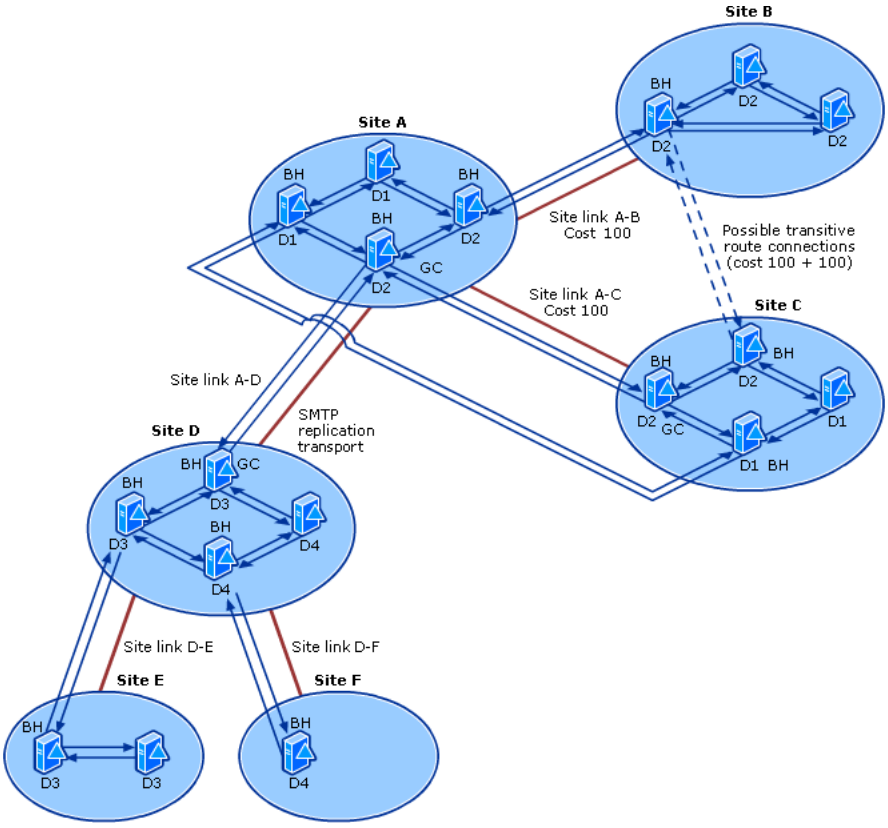

Figure 6.3: Mapping your replication topology.

What's even better are some of the third-party (including some free ones out there) tools that can analyze your directory and draw this type of picture for you—as Figure 6.4 shows. The differences between your actual topology, and the one you *think* you have, can be enlightening.

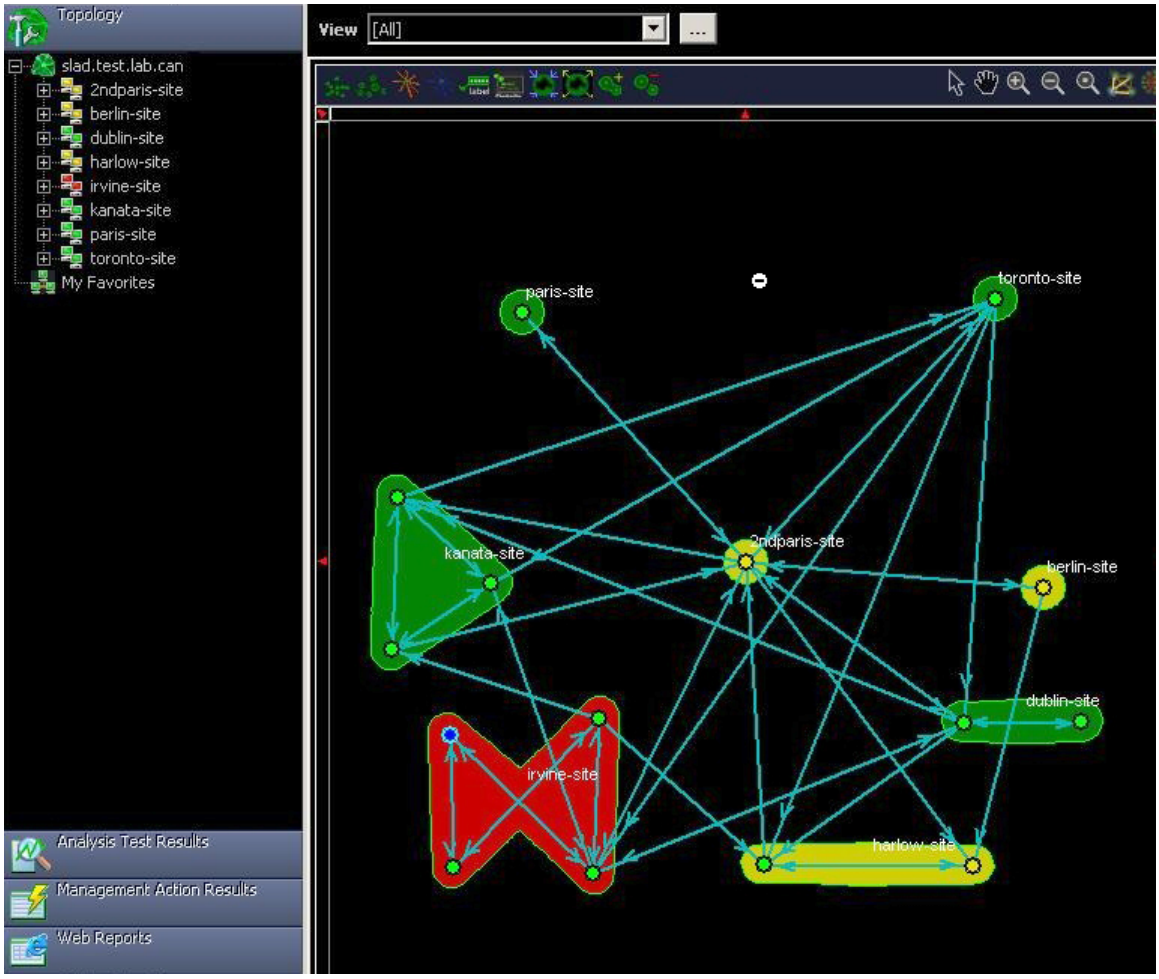

Figure 6.4: Tool-generated actual replication topology.

The goal should be to simply ensure that no domain controller is too many steps away from every other domain controller so that replication can quickly get changes out to every domain controller in a minimum number of "hops." At the same time, you want to ensure that the physical WAN links can handle the replication traffic you're putting on them. That's especially true when you have a lot of manually-configured site link bridges, which deliberately "double up" the traffic on your WAN links in an effort to reduce replication hops between distant sites.

It's *really* important not to rely solely on a hand-drawn diagram of your replication topology because AD won't always make the exact same calculations as you about which domain controllers should be bridgeheads, and it's easy to overlook things like site link costs that might be making AD calculate unexpected and unwanted topologies. Get your hands on some kind of tool that can draw a topology *based on what AD is actually doing,* and compare that with your hand-drawn "expectation diagram."

FSMO Placement

Recommendations on FSMO placement have changed over the years; http://support.microsoft.com/kb/223346 offers the latest guidance. In general, it's considered safe to stack all of the FSMO roles onto a single domain controller, provided it is located at a hub site (that is, has good physical WAN or LAN connectivity to most other sites). The only exception is for environments that don't have a Global Catalog (GC) hosted on *every* domain controller; in those cases, move the infrastructure master to a domain controller that doesn't host the GC.

Some FSMO roles are forest-wide: The schema master and domain naming master should co-locate with the PDC emulator of the forest root domain. Again, that domain controller should be well-connected to the other domain controllers in the forest, ideally located at a hub site that has good WAN connectivity to most other sites.

Virtualization

Can you virtualize your AD infrastructure? Of course you *can.* Should you? In a word, yes. You should. The long-term benefits of virtualization have been proved by scientists: easier workload management, easier disaster recovery, easier scalability, lower power requirements, lower cooling requirements, less data center space—and the list goes on and on.

Frankly, there's no reason *not* to. AD works and plays quite well in a virtual environment. In fact, with modern memory overcommit, you can really leverage AD's unique usage patterns. AD gets busy and needs a lot of memory in the mornings when everyone is logging on. So co-locate your AD virtual machines with virtual machines that run other tasks, such as line-of-business applications. As logon traffic settles, people grab the bagel, and get to work, AD virtual machines will need less physical memory, and that can then be devoted to the line-of-business virtual machines. Just scatter your AD virtual machines across several virtualization hosts and you're golden.

And consider installing AD on Server Core, not the full install of Windows. Server Core has a *vastly* smaller footprint, meaning more of the virtual machine's resources can go to AD. Server Core requires less maintenance (it has a lot fewer patches over time than the full install), so you'll spend less time maintaining your virtual machines. Server Core's disk footprint is smaller, making it easier to move from host to host. And Server Core can still run all of your management tools, agents, anti-malware, and other stuff (popular myths to the contrary). If you're accustomed to running DNS, DHCP, WINS, and other infrastructure functions on your domain controllers—well, Server Core runs those too. And those roles are completely manageable via the same GUI consoles you use today: Active Directory Users and Computers, DNS Management, and so on. You'll find yourself logging onto the console very rarely, if at all (even Server Manager supports remote connectivity in Win2008R2).

Ongoing Maintenance

Aside from object-level maintenance—you know, cleaning up disabled users, stale computer accounts, and so forth—what kind of ongoing maintenance should you be performing in AD? Backups are obviously important. As I've mentioned already, my preference is for continual backups made by a disk-to-disk recovery system rather than tape, but if tape's what you've got, then at least use that.

> **Disk-Disk-Tape**
>
> By the way, just because I advocate disk-to-disk backups doesn't mean I don't see the value of tape, especially for getting a copy of your backups safely off-site. Most disk-to-disk systems provide support for making a second tape-based backup for just that purpose. And because you're essentially "backing up the backup," you can enjoy longer backup windows without affecting the production environment.

Check the logs and make sure that both AD and the File Replication Service (FRS) aren't generating anything alarming. With a continual monitoring solution (like System Center Operations Manager or something similar), you can simply let the solution keep track and alert you if there's a problem.

Also keep an eye on disk space on whatever volume contains the AD databases. Again, a monitoring solution can be used to alert you when disk space gets low, so this doesn't have to be a manual task. You should also have a plan in place to regularly defragment that logical disk—third-party defrag utilities can do so continuously or on a routine maintenance schedule, or you can use the native defrag tool on a regular basis. Once a quarter works for many of my consulting clients.

Periodically review the log to look for replication problems—just being proactive, here. A monitoring solution can do this routinely and alert you to any problems, but it's always good to just run some of the replication monitoring tools (discussed in previous chapters) to make sure everything is working smoothly.

Finally, take time each month or so to run the BPA model for AD (on Win2008R2 and later). You can do this in PowerShell or via Server Manager (Figure 6.5 shows where to find it in Server Manager). The BPA is a collection of Microsoft guidelines for properly configuring AD and other server roles; running the model on a regular basis helps ensure that you keep AD properly configured over the long term for better security, performance, reliability, and so forth.

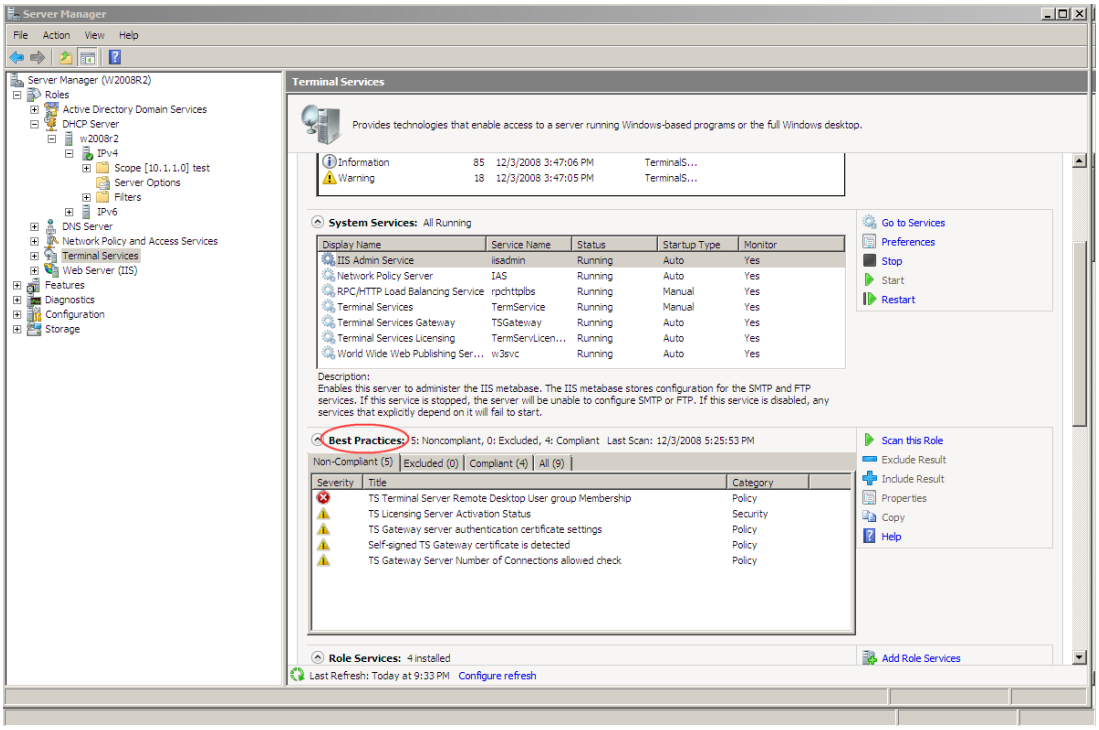

Figure 6.5: The BPA in Server Manager.

Most maintenance in AD is that business-level, object-focused kind of maintenance: stale computer accounts and so forth. AD is largely self-maintaining otherwise, meaning you just need to glance at it occasionally to make sure everything's working smoothly.

Coming Up Next

In the next chapter, I want to take a sort of intermission and discuss Active Directory Lightweight Directory Services, or AD LDS. Formerly known as "Active Directory Application Mode," or "ADAM," this trimmed-down version of AD has very specific uses within an organization and can help solve very specific problems. We'll talk about what it is, when to use it, when not to use it, and cover some of its unique troubleshooting and auditing concerns.

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.