# Windows Server 2012 Essentials: Domain vs. Workgroup

*Sep 26, 2012* **Paul Thurrott** | ***Paul Thurrott's Supersite for Windows***

In a recent article, Windows Server 2012 Essentials: Connect Client PCs Without Using A Domain, I described how it's actually possible to configure PCs to connect to Windows Server 2012 Essentials without adding them to the domain. This raised a number of questions, however, around what a domain is, and whether such a thing would be useful in a home office or very small business. So, let's step back for a moment and take a look at these issues.

Reader reaction to the article was interesting, and very much according to experience. Many of those who had used Windows Home Server but no other type of Windows Server product have no idea what a domain is. Those who did have traditional Windows Server experience, however, wondered why I was so down on domains: After all, Essentials 2012 offers the simplest domain setup imaginable.

Both reactions are completely understandable. As it turns out, using a domain in a home office or very small business comes with both advantages and disadvantages. Originally, I planned to tackle a number of these issues in a single article, but it got a bit convoluted and lengthy, so I've split it into multiple parts. First up, a very basic overview of domains and how they compare to workgroups, aimed at Windows Home Server users

## What is a domain?

I assume that virtually everyone reading this article has a home network of some kind, with wired and/or wireless connectivity to two or more PCs. You may have never really considered what this sort of network "is," but in Microsoft networking parlance, that's type of network is called a *workgroup*. In a workgroup, each PC is essentially a standalone unit, with its own users, configurations, applications, and so on. To manage such a PC, you typically access it physically, sign in, and get to work.

Naturally, once you've got two or more PCs networked together, the notion of sharing resources comes up. Each PC has its own storage, with whatever files—a media collection, perhaps, documents, whatever—and some PCs may be connected to a printer. So Windows, like other PC operating systems, has evolved various ways to share these resources. Printer sharing is probably pretty well understood. And there are different ways in which you can share folders on a PC with others on the home network.

Some folder sharing techniques are sophisticated but tedious: You could manually configure which individual users can access each shared folder on each PC. Some are simple but dumb: You could simply sign in to each PC with the same exact user

name/password combos. Some are a bit more seamless, as with the homegroup sharing scheme Microsoft introduced in Windows 7; this works within the workgroup networking type to make sharing folders (and printers) easier.

Workgroups make sense in very small environments, like your home, or a small business with less than 10 PCs. But once you scale up beyond that, managing users and sharing, and other things like PC configurations and application installs, gets difficult. For this reason, Microsoft and other firms have created *directory services*, which centralize the management of users and their settings, PCs, applications, and other resources through a server or group of servers that work in tandem. Microsoft's directory service is called Active Directory.

Active Directory debuted in Windows 2000 and has been evolving and maturing ever since. An environment based on Active Directory (AD) is called a *Windows domain*, or just a *domain*, and this entity defines a collection of resources. These include users, of course, but also PCs, services, and other objects, and the relationships between them. In fact, one way to view a domain is via the security rights that each user is provided: These rights control access to various domain assets. But basically, AD is just a formal, centralized way to manage a network, or computing environment.

(**Note:** Domains should not be confused with Internet domains, like microsoft.com, though of course there is often a relationship between entities of each type.)

Let's put this in real world terms and say you have a home or small business with five PCs. Each PC will typically be used by a single and different user, but you may have instances in which users want to move around, and use different PCs.

On a workgroup, you would need to manually configure each PC with user accounts for each user. If you literally had to create five identical user accounts on each PC, this can get tedious, though it arguably only has to be done once. But that doesn't resolve issues such as installing applications—once per PC—configuring those applications on a per user basis—which would have to happen five times on each PC, manually—or making sure that each PC was up to date with security software. And what about changing passwords over time? You'd have to do that manually, on each PC.

Even this simple environment could benefit from a domain, assuming you can handle the complexity of the configuration and the cost of a server and its OS. The user accounts would be managed centrally, so any user could walk up to any PC at any time and sign-in: The sign-in would go through the AD server(s), and if the password was changed once, it would be up to date via any of the PCs, automatically. Applications can be deployed through AD (and related management tools), and kept up to date, as can Windows. And

you can specify security policies—and many other useful things—though an AD construct called *Group Policy*.

Sounds great, right? There's just one problem: AD is hard. On traditional (non-Essentials) Windows Server installs, AD is difficult to set up correctly and requires some IT pro expertise. The AD structure itself can be fairly complex, not quite Registry complex, but hierarchical and somewhat non-intuitive to newcomers. And ideally, a healthy AD environment would include at least two servers, which are called *domain controllers*, to replicate the AD database and provide some basic form of disaster recovery.

As the simplest possible Windows Server, Essentials 2012 takes the pain away from configuring AD, in ways that I still find incredible. And in day to day use, you only very infrequently need to use Server's normally complex interfaces—Server Manager and a full suite of older, still-used administrative tools—thanks to the inclusion of a very simple Dashboard that manages features that are both unique to Essentials and many that are common to other Windows Server versions. But sometimes you still need to get your hands dirty. Like Small Business Server before it, Essentials tries to protect you as much as it can. But it's still a full-fledged AD-based Windows Server install. Sometimes there's no getting around that. Power comes at a price.

WHS, by comparison, was also Windows Server, of course. But because WHS did not require AD—and could in fact not be made into a domain controller at all—it was much, much simpler. It was also less viable for those environments that did need the centralized management functionality of a real domain. I'm a bit curious that Microsoft didn't engineer Essentials to provide a domain/workgroup choice during Setup. If it had, we wouldn't be having this conversation. That said, Essentials 2012 offers the simplest domain set up yet. And as I'll discuss in future articles, the benefits of this approach may outweigh the complexities, if what you're used to is Windows or Windows Home Server.

More soon.