Mike Nelson – SVD article

Using VDI as a Test Environment

"I wish I had a clean system right now to test that on". I cannot tell you how many times I've heard that statement from folks in IT. And, most of the time, these folks have some kind of Virtual Desktop solution deployed in their environment, but don't really even know the power to which they possess with it. VDI can be so much more than just desktop delivery if you take it a few steps further.

Most desktop developers will tell you that they have to have several environments to work in to create, test, and prove out their code. Before VDI really caught on some years ago, and even still today, most relied on the client Type-2 hypervisors like VMware Workstation or Microsoft's Virtual Server to help them get the environments they needed. This is also true to other areas of IT including Quality Assurance, User Acceptance Testing, and even Desktop Package deployment specialists. But now, with the power and flexibility of VDI, those days should be gone for most.

Before I get too far into it, I have to say that I'm not really advocating the use of what I call a "Poor Man's VDI" for testing purposes. To me, a Poor Man's VDI is using a hypervisor platform like VMware or XenServer to create static one-to-one virtual machines that have no "Broker" service that controls their creation, resources, state, or destruction. This is too difficult to manage outside of being used by a handful of people, and, as it scales, it becomes much more costly as an overall VDI solution. Rather, for this article, I am specifically referring to a VDI solution that incorporates a full blown Broker service, such as VMware View or Citrix's XenDesktop, which really gives you the best flexibility and scalability in a one-to-one or one-to-many desktop deployment.

Some of the really great advantages of using a VDI environment for your entire desktop (and possibly even server – more on that later) Operating System testing needs are:

- Quick to deploy multiple desktops
- Configuration or "State" of the desktop can be variable, which can be beneficial for multiple testing landscapes and allow for more flexible testing
- Snapshots are always great for Moment-In-Time testing.
- Wizards that can create from one to a thousand plus desktops for you, and even add them to an Active Directory domain, with a few mouse clicks. Couple this with PowerShell scripting (my favorite) to add testing users, and you've got a pretty great automated solution!
- Thin-provisioning makes great use of limited (and costly) disk space.
- And, did I mention that it's quick?

Now, you'll notice that I dropped a somewhat subtle (well, ok, not very subtle) hint earlier about being able to use VDI to deploy server operating systems. Well, you can, but I'll be the first to tell you that some VDI vendors probably won't officially support it. That being said, I always walk on the side of risk and have deployed VDI Broker controlled Windows Server 2003 R2 and 2008 R2 virtual machines without any issue using both XenDesktop and View. Here's a good discussion on the XenDesktop method. VMware's View product actually does support 2008 R2 as a supported configuration, so there's no risk-taking there. I have used both, but the lack of support for server OS's on Citrix's part (outside of Provisioning Server) can be a deal breaker for some, while View embraces them and it works very well. It should also be noted that I believe this was the master plan by VMware to support server OS's in View since they are doing away with their Lab Manager product line and combining all of those capabilities into View.

When I am talking with those folks who make that statement about really wishing they had a clean system to test on right now, I always come back and ask, "Do you have a virtual desktop solution you can use?". No matter if their answer is yes or no, I tell them that they either already have the tools to get them where they want to be and just don't know it, or how they can get them, and how to create their own testing environment for what they need today and in the future.