5 Application Management Mistakes That Doom VDI Projects

Virtual Desktop Infrastructure (VDI) has never been hotter. But many VDI projects stall, or even fail, because of five common application management mistakes. Learn what they are – and how to avoid them – in this eBook.

Introduction

When it comes to desktops – physical or virtual – it's all about the applications. Every human needs different apps for their work. If we all used the same apps, we would be happy using Windows terminal servers as our desktops.

That's why cloud-hosted virtual desktops are growing so fast. You get the same data center-class security and 24x7 access to your Windows applications that you get with terminal services. But now you get the complete personalization and flexibility of having your very own desktop.

Which brings us back to the apps. How will you deliver them? How will you update them? How will you minimize licensing costs? How will you offload your overworked IT staff from having to manage every little app? This is where many Citrix XenDesktop and VMware Horizon View projects get stuck.

Organizations make 5 common mistakes when it comes to planning and implementing their application management strategy. This eBook tells you what they are and how to avoid them, and offers real-life case studies on customers who didn't let their apps get in the way of VDI success.

Mistake #1
You Couldn't Virtualize All Your Apps

As many organizations have discovered, not all apps can be virtualized with traditional application virtualization technology. Microsoft App-V, VMware ThinApp, and other first-generation app virtualization tools are excellent for packaging apps in protective isolation bubbles so they can run without conflict.

However, there are many apps that cannot be virtualized with this technical approach:

- Applications that interact with low-levels of the operating system using kernel-mode drivers such as antivirus, firewall software, and VPN clients.
- Applications with device drivers such as scanners and printers.
- Custom and homegrown apps that require complex Setup procedures, Shell integration, DCOM or COM+ services, and libraries that use global hooks.

Hamilton County, Indiana found itself in this situation. Its VDI project was stuck at 100 desktops for over a year because QuickBooks, Roxio DVD burner, Dymo label writer, Track-It! help desk, Odyssey court and justice software, and other apps required by the local government body would not virtualize, even with the help of outside consultants.

How They Fixed It

The county got past this roadblock when the IT team implemented new desktop layering technology for desktop provisioning, application virtualization, and image management. Instead of isolating apps in their own bubbles, desktop layering uses a new technical approach – file system and Windows registry virtualization, or C: drive virtualization – that enables almost any application to be virtualized separate from the Windows OS.

As **VMware's case study on Hamilton County** shows, the VDI project is now a success. 400 desktops are in production, with more being rolled out as PCs reach end-of-life. The combination of VMware Horizon View for secure, high performance virtual desktop connectivity and Unidesk for layered desktop management and application virtualization has enabled all of Hamilton County's apps to be packaged and delivered independent of its single Windows gold image. The app layers can be delivered in any combination to any number of desktops, making it easy for the IT staff to meet the county's diverse app requirements.

Mistake #2
You Underestimated the Time and Expertise Needed to Package Your Apps

Even if your apps can be virtualized using isolation technology, most organizations find that the sequencing and packaging process requires far more time and expertise than anticipated. By the time you've finished the desktop setup, pre-scans, post-scans, scripting workarounds, Windows registry changes, and deployment, you'll find a full day has passed. Or more.

The app virtualization vendors acknowledge this. Here is an excerpt from the *Microsoft Application Virtualization 5.0 Sequencing Guide*:

App Type	Description	Time Scale
Moderate	This is probably the most common application type. Moderate application types might require some changes while sequencing to function correctly, or may require no changes but have a larger install that takes more time. In rare instances, both scenarios can occur. Changes that might be encountered in these packages include making changes to the registry, altering the DeploymentConfig or UserConfig file to launch with additional parameters and scripts, or there may be additional applications needed to install together as a suite to allow cross-functionality.	Typical sequencing time: 1-4 hours
Complex	These are large applications or applications that take four or more hours to install, significant amounts of customization to function in the virtual environment, or both. Packages like these will normally be 3-4 GB in size and may require compression to get the package under the 4GB App-V limit. Other hurdles you may encounter are the application's reliance on files being in a specific place and functions hard-coded to that install. These applications may require you to manually edit batch and command files to point to resources in the virtual environment. If this is the case, it is highly recommended utilizing a program that can scan multiple files and make several changes at once. You also may be required to install a device driver separately since drivers cannot be virtualized. Applications of this complexity can be sequenced, however it is imperative that, before you begin, all the pieces must be in place. All knowledgeable resources should be engaged and available, sequencing hardware should be better than average, and finally, sequencing applications such as these should be done by an experienced sequencer.	Typical sequencing time: 4-8 hours, but could be longer depending on the size and number of files

Sunrise Health hit this challenge early in its 3,000-user VDI project. Over 100 applications are required by the healthcare organization's clinicians, nursing stations, kiosks, and administrative staff. Pilot testing convinced the lean IT staff that trying to isolate all of its apps – and repeat the process whenever apps needed updating – would not be viable.

How They Fixed It

Sunrise Health solved this issue with desktop layering. The layering process is so fast and easy that packaging contests were run internally to see which IT administrator could virtualize the most apps.

Sunrise Health has now packaged close to 100 apps as layers. The layering process typically takes less than 30 minutes – even for complex apps – and requires no special packaging or sequencing skills. Sunrise Health administrators select an Installation virtual machine, log onto the machine, install the application (doing whatever they would normally do to install the app on a regular desktop), and click Finalize. The application layer is immediately available to be assigned to any desktop on top of Sunrise Health's clean Windows OS layer.

Read the **Sunrise Health VDI case study** to learn more about the healthcare organization's growing Citrix XenDesktop and Unidesk implementation.

Mistake #3
You Didn't Account for Apps Needing to Communicate

As mentioned earlier, first-generation app virtualization tools use isolation technology to package apps in protective "bubbles," effectively hiding them from Windows and other apps. This is perfect for running multiple versions of the same software (e.g. Java or Microsoft Access) on the same desktop. However, for the majority of apps that do not need to be isolated, and that need to share data, link to each other, and cross-communicate, this presents a major problem.

If you want to stay with the isolation approach, there are two workarounds:

1. You can sequence all of the apps that need to communicate in the same package. However, this often results in very large packages that need to be re-sequenced every time one of the apps in the package needs updating.
2. You can "poke holes" in the isolation bubbles so that the apps can "see" each other. However, this adds more packaging time to an already lengthy and time-consuming packaging process. It also ups the expertise requirements.

This is the situation that law firm Bernstein Shur found itself in. Like most law firms, Bernstein Shur uses a wide range of Microsoft Word and Microsoft Outlook plug-ins. The plug-ins change frequently, and must be constantly updated.

The firm tried to isolate the plug-ins so they could be updated independent of Word and Outlook, but the plug-ins could not communicate with the base apps.

They experimented with packaging all of the plug-ins together, but that made updating them extremely inefficient. Word, Outlook, and the plug-ins had to be re-packaged every time one of the plug-ins changed.

How They Fixed It

As this **Unidesk and VMware Horizon View case study on Bernstein Shur** shows, layering was the solution.

Layered apps are not isolated. They appear to Windows and to other apps as if they are natively installed. They show up in Windows Add/Remove Programs, and their files appear in the expected directories.

Bernstein Shur has successfully virtualized all of its plug-ins as separate layers to make patching and updating fast and easy. Yet the plug-ins all work with their base applications as expected.

Mistake #4
You Tried to Deliver Apps In Your Windows Image

Some organizations get so frustrated by the first three mistakes that they say “Enough already! I’m just going to stick my apps in my base Windows image and deliver them that way!”

Don’t make this mistake.

Building every possible app into a single Windows image will force you to license every app for every user. Plus, you’ll have to update the master image every time you need to update one app. If you’re one of the .001% of organizations who can afford to do either of these things, move on to Mistake #5.

You might say “I’ll just create a few different images each containing different sets of apps.”

Don’t do this either.

Most organizations that go down this path inevitably find that two images leads to four, four becomes eight, and soon, instead of patching Windows once, they’re doing it 20 times every Patch Tuesday.

This is the last thing State of Ohio Department of Developmental Disabilities wanted. One of the goals of its 1,400 desktop VDI project was to simplify Windows management, and move some of its desktop support staff to more interesting, strategic projects that would improve the quality of services offered to the state’s developmentally disabled residents.

How They Fixed It

Desktop layering enabled them to achieve their goal. Read the **VMware Horizon View and Unidesk case study on State of Ohio Department of Developmental Disabilities** to see how the agency layered more than 60 apps to keep its Windows OS layer clean. Because all 1,400 desktops can be kept up to date by patching Windows or any of its 60 app layers once, the agency was able to redeploy 7 IT support managers to other forward-facing projects.

Mistake #5
You Forgot About One-Off, Ad Hoc Apps

Even if you use layering technology to make virtualizing your standard and departmental apps fast and easy, there is still a boatload of one-off, ad hoc apps that are used by a few users. If you try to centrally deliver all of these, you'll drive yourself crazy. So what are your options?

If you deploy non-persistent desktops, you don't really have one. Any apps that IT or end users install on non-persistent desktops will be lost after a patch, reboot, or logoff.

This is what Colby-Sawyer College discovered when it looked at deploying non-persistent desktops for its administrative staff. Desktops that wipe themselves clean might be ideal for labs where you want each student to start fresh. However, for the college's staff users who expect their unique plug-ins and apps to always be present, non-persistent virtual desktops would not be any better than their old Windows terminal server sessions.

How They Fixed It

The college's solution was to provision 275 persistent desktops for its staff using desktop layering technology. Power users with administrative rights are allowed to install one-off apps on their own to ease the burden on the IT staff. For less savvy users, the IT staff installs logs onto their VM as administrator and installs the one-off apps for them. In both cases, the user-installed and IT-installed apps survive logoffs, reboots, and Windows patches.

Layering has eliminated the old issues of persistent desktops using too much storage and requiring a full Windows image for every desktop. All 275 desktops are provisioned from a single Windows gold OS layer that only needs to be patched once. 65 apps have been virtualized as independent app layers that are also stored and updated once.

The staff's customizations – including their user-installed apps – are captured in each desktop's Personalization layer. To fix most desktop problems, the college's help desk simply rolls the Personalization layer back to a previous snapshot and reboots the desktop.

Read the **blog on Colby-Sawyer's VMware Horizon View and Unidesk implementation** to learn more.

Summary

You should now be equipped to avoid the 5 common application management mistakes that can doom VDI projects. With an application management strategy that incorporates next-generation, industry-proven layering technology, you can ensure that apps don't come between you and your VDI success.

Unidesk Corporation, 313 Boston Post Road West, Marlborough, MA 01752 USA Tel 508-573-7800 Fax 508-573-7801