

The Proactive Migration to Windows 7

A ScriptLogic Product Positioning Paper

The Proactive Migration to Windows 7

With each new exciting release of an operating system from Microsoft, there comes the sudden realization by every IT pro that you are going to need to perform the migration to this new platform. This is usually followed by several other realizations about how “this program” probably won’t be compatible and “that user” is going to need their specific settings to migrate, etc. So while the end result (a new secure and more productive OS) looks fantastic, getting there is obviously going to be a challenge. Since you are reading this whitepaper, I can only assume you a) are planning a pending migration to Windows 7 and b) you’ve come to the previously mentioned realizations.

In this whitepaper, I’d like to discuss a very different migration strategy to moving your users and their desktops (a term I’m using generically throughout this paper to mean any PC regardless of its physical configuration – laptop or otherwise) to Windows 7; one that takes a very proactive approach that will result in a faster migration, a more standardized environment, and more productive users. Before I jump into the migration, you first need to buy into three basic statements about your desktops:

- 1) **Having one-off desktop configurations is unacceptable.** This amounts to exceedingly higher support costs (that equates to your time) over the life of the desktop.
- 2) **Your desktop configurations should be both standardized and centralized.** This includes the OS, apps, user configurations, profile settings, drive mappings and anything else that makes up your user’s environment. *Standardized* does not mean that every desktop is the same; every user has their own set of needs. *Standardized* does mean that every desktop configuration (and there will usually be more than one) is intentionally put into production by IT; IT is aware of the configuration and has put a technology in place to implement the configuration. *Centralized* means that the configuration of each and every desktop is created, stored and able to be modified to update the organization’s definition of a consistent, secure and functional working environment.
- 3) **Cramming a standard configuration into an OS image is not desktop management.** This amounts to maintaining a standard for about one week (until changes by users or IT alter the configuration) out of 3 years of a desktop’s lifetime.

If we are in agreement on these three concepts, you’re ready to begin.

What's In a Move?

Migrating to Windows 7 is not as simple as just putting in the DVD and running setup; there are hardware and application compatibility issues, opportunities to establish (or reestablish) a standard desktop environment, and the need to keep users productive. In order to address the issues that go along with a migration to Windows 7, I'll focus on the three key data sets involved:

- Operating System
- Applications
- User State

Yes, of course, there are many, many caveats to each of these data sets, but in simplifying it down to the basics, these make up the major considerations within your pending migration. I would guess that most of you have a loose plan in mind already - something like the following:

- 1) **Build the Windows 7 desktop image** – complete with Office and other company-wide apps.
- 2) **Deploy the OS** – using Windows Deployment Services (the updated version of RIS) or via WindowsPE with ImageX.
- 3) **Deploy applications** – either manually, logon scripts, or Group Policies .
- 4) **Migrate the user settings** – using Microsoft's User State Migration Tool (USMT) and/or update with logon scripts or Group Policies.

Unless you are planning to perform an in-place upgrade of each desktop, you are in the vast majority that desire to take the opportunity and create some form of desktop standard, even if it means simply wiping the OS, and putting a fresh copy of Windows 7 on the desktop. Standardizing the desktop lowers support costs and the overall total cost of ownership (TCO) of the desktop. So let's dig a bit deeper into each of the four steps I listed and look at a few issues with this methodology, not the toolset being used, but the methodology and how they will affect your desired outcome of a standardized desktop.

Imaging/Deploying Windows 7

Building a standard image of Windows 7 and deploying it is one of the single best steps you can take to a standardized deployment. I've seen organizations that have literally documented the settings on their image down to "this checkbox on this specific dialog box is unchecked" level of detail. **The mistake that many make is placing applications onto the image.** There are a number of reasons you should not put applications on a standard image:

- 1) **Application Conflict** – with applications pre-existing at the time of OS deployment, you risk conflicts of DLLs and registry entries later when additional applications need to be deployed.
- 2) **Application Configuration** – not everyone needs the exact same applications nor the same configuration. This means you are going to need to perform additional post-image steps to "correct" the configuration issues created by pre-existing applications.

- 3) **The “Standardized Desktop” is a moving target** – What you need on the desktop today may not (and probably will not) be the same a year from now. If you place applications on your image, you are time-stamping the standard back to “day 1” where any new images will need to start.
- 4) **OS Refreshes** – should you desire to refresh the OS configuration for a new set of desktops, say, 9 months from now, the applications you pre-installed to the image may need updating, resulting in the need to build a brand new image, instead of simply making a tweak to the image and deploying out applications separately.

The Proactive Migration – Imaging: The best thing you can do at this phase of the deployment is to build a Windows 7 image that has no applications and utilize a separate solution (even if it is Group Policies) to deploy your applications. The result? A standardized OS (and OS only!) that is ready to accept whatever the standardized set of applications and configurations are that meet the business goals today – NOT when the image was built. For example, assume it is two years into your last desktop refresh and you need to re-image a desktop, but it needs to have Office 2007 on it. If two years ago you built a clean image of XP with no applications, the installation of Office 2007 can be automated and configured according to spec. In contrast, the reactive method of building an XP image with Office 2003 on it two years ago would require you today to potentially perform a manual install of Office 2007 to ensure no issues arise during the upgrade.

Deploying Applications

As with all new Microsoft OSes, you cannot simply take your old applications and assume they will work properly in Windows 7, you will need to do a bit of leg work here. To get proper attention to the work that needs accomplishing, I’ll break “deploying applications” down into four parts: *testing, packaging, deployment* and *making your applications work*.

Testing your Applications

With a Windows 7 migration, the testing of applications becomes critical, as this is the step where you will need to ensure each the compatibility of each of your apps with Windows 7. While you can pretty much skip testing Office 2007 with Windows 7, as that has already been done for you by Microsoft, other applications that you may be asking Office to coexist with may not “play well” with Windows 7. There are a few things you can do to test your applications with Windows 7:

- 1) **Look for a list of compatible apps** – Try Microsoft’s *Windows 7 Compatibility Center* (still in “Coming Soon” mode at the time of writing), found at:
<http://www.microsoft.com/windows/compatibility/windows-7/default.aspx>.
- 2) **Try it out** – an obvious choice, but a viable one. Install and see how your application runs on Windows 7. Be sure to test out the app using any one of the impressive **10** compatibility modes Windows 7 supports as well. Microsoft has taken steps to make apps as compatible as possible.

- 3) **Use some of the tools in Microsoft’s Application Compatibility Toolkit (ACT)** – this toolkit specifically addresses the compatibility of the installation, configuration and security needs of an application to determine whether it will or will not work. Information on ACT can be found at <http://technet.microsoft.com/en-us/windows/aa905066.aspx>.
- 4) **When all else fails, use XP Mode** – If you’re not familiar with this new feature, XP Mode uses an embedded Windows XP Virtual PC session running on Windows 7 to actually run the app, but presents the app only (not the entire virtual desktop) to the end user. The end experience is they interact with just the (functioning) app like any other and are unaware of the virtual technology under the hood to make it happen, as shown in Figure 1. More on XP Mode with Virtual PC can be found at <http://www.microsoft.com/virtual-pc>.

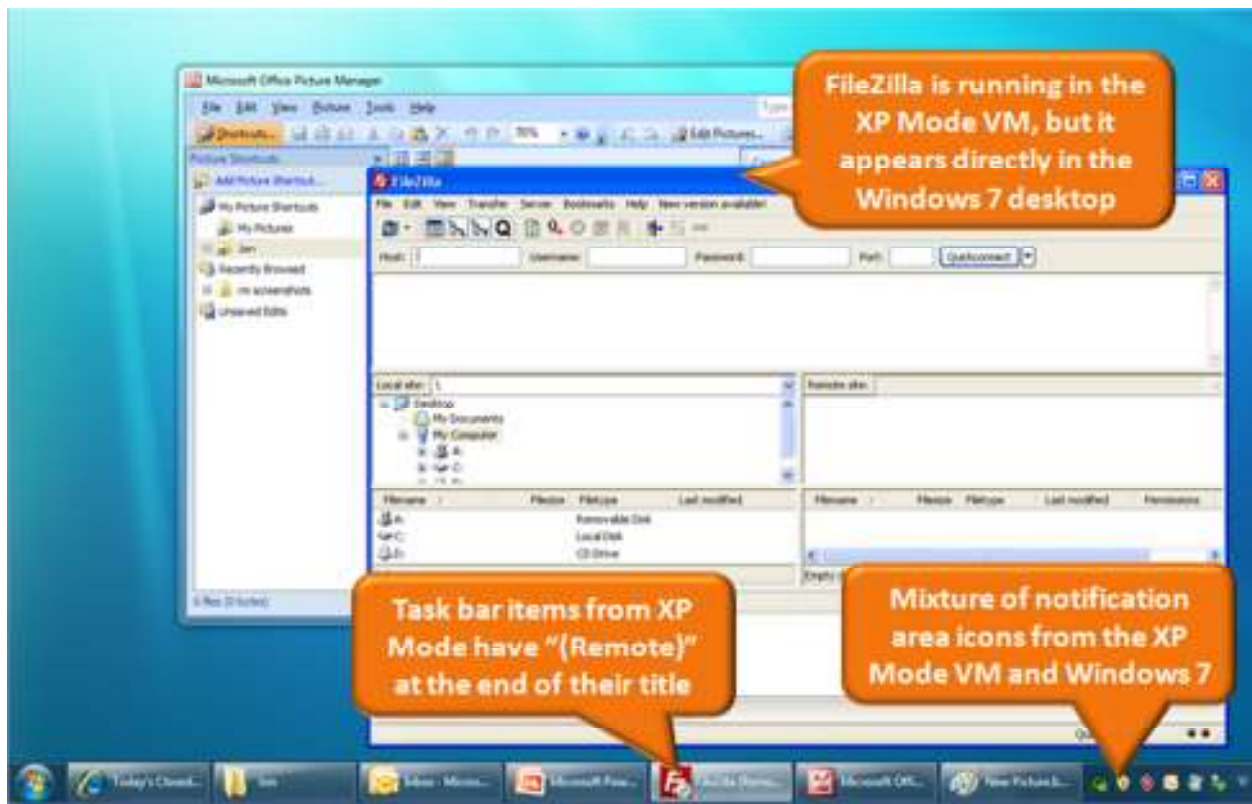


Figure 1: XP Mode presents apps running in a Virtual PC as part of the Windows 7 desktop.

You need to equally place importance on ensuring your applications play well with each other. The application issues I mentioned earlier, such as DLLs or registry entries conflicts, require that you **test each application you deploy with every other application it may potentially coexist with**. While that sounds impossible, it isn’t. ScriptLogic’s MSI Studio Pro maintains a repository (called a “Software Warehouse”) of all packaged MSIs and completely automates the process of validating a newly introduced application’s compatibility with one or more (or all) of the other applications in the repository. Figure 2 shows MSI Studio’s Software Warehouse that organizes applications into grouping defined by you and empowers you to test a given application against one or more of the other applications in the warehouse.



Figure 2: Testing Applications for Conflicts with Desktop Authority MSI Studio

The Proactive Migration – Testing Apps: Using solutions that look at the sum of all your applications and their viability of running together on Windows 7 will make you aware of conflicts before they happen to your users. ScriptLogic’s MSI Studio and Microsoft’s ACT will ensure application compatibility.

Packaging your Applications

To ensure a consistent deployment of applications, a few requirements at the time of deployment must be considered when deciding how much emphasis must be placed on packaging (and in some cases, re-packaging) application installs:

- 1) **Application installs must be silent, requiring no user interaction** – you certainly do not want a user to be responsible for configuring their own apps, and you don’t want to sneakernet to each machine to press Next a dozen times either!
- 2) **The configuration of the install must be consistent** – this is accomplished by the installation either being self-contained within the MSI or accessed via a transform (MST) file.
- 3) **The install must be able to run with elevated privileges, if required** – Windows 7’s User Access Control (UAC) mandates that a user be only granted the most restrictive permissions that still allow them to accomplish required tasks. This can be an issue if an installation of an app requires elevated privileges. UAC addresses this on the desktop side by prompting for elevated credentials. However, the need for elevated credentials must be identified by the application itself so UAC is aware.

ScriptLogic's MSI Studio can be used as a powerful tool to not only package legacy apps that have no MSI, but to also re-package existing MSIs, and even compare two MSIs to generate an MST file. Figure 3 shows the Installation Properties IQ View where you are able to establish the elevated privilege requirement for UAC. Also note the other IQ View elements available on the left side that make the modification of MSI packages a simple task.

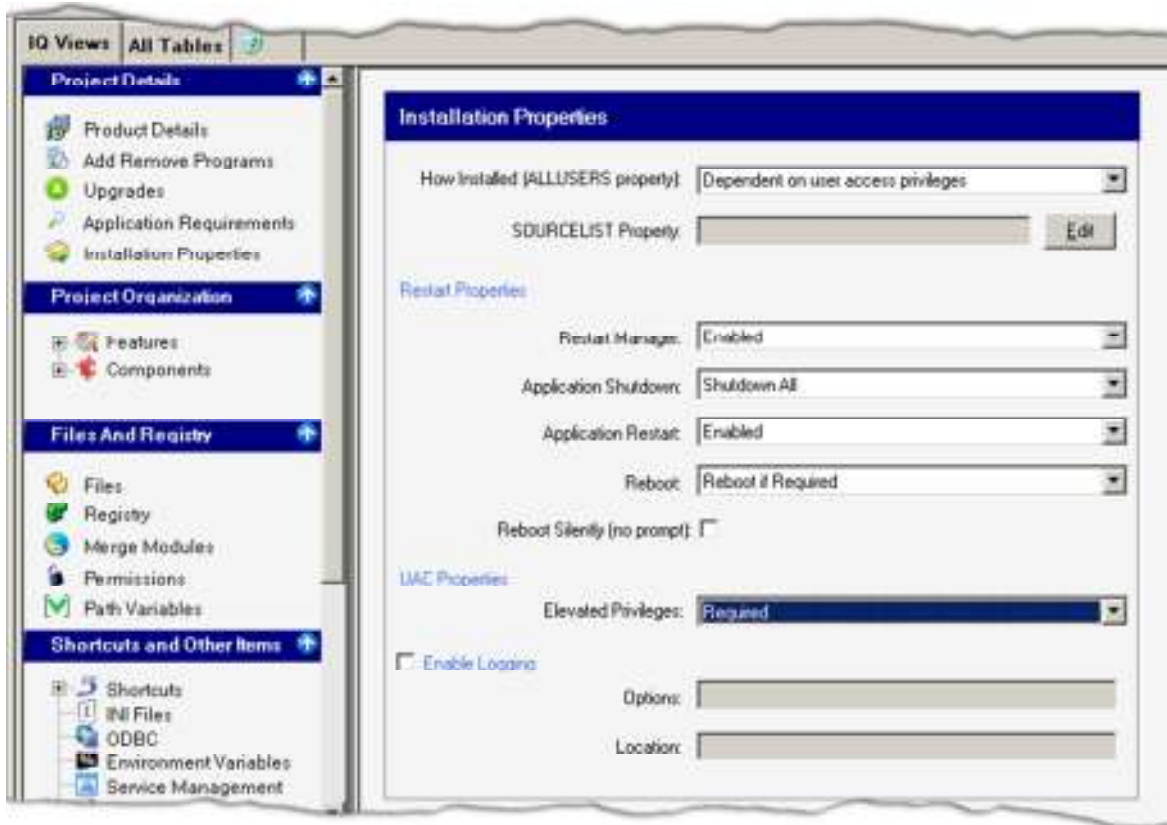


Figure 3: Modifying the UAC Elevated Privileges Requirements

The Proactive Migration – Packaging Apps: Packaging legacy apps and repackaging existing MSIs so they are customized to the specific configuration your business requires will keep you from needlessly spending time tweaking registry values, configuring options screens and visiting users for one-off changes in the future.

Deploying your Applications

I would guess today, most of you are either manually deploying apps on a one-off basis or using Group Policies or Logon Scripts to deploy them. A solid deployment strategy includes the following characteristics:

- 1) **Silent /Consistent / Privileged** – see the rationale in the previous section.
- 2) **Automated** – you should not be relying on “sneakernet” to get the deployment accomplished.
- 3) **Selective** – applications should only be deployed to those users that require them

- 4) **Distributed** – deployments should utilize a distribution network of servers to ensure the closest installation point (preferably on the same network) is used.
- 5) **Centralized** – Keeping the management of your application deployment centralized will allow you to maintain the standard desktop throughout the desktop lifecycle.

Most of these characteristics can be achieved by using an MSI packaging solution and either group policies or multiple installation points. But that's not the hard part. The greatest challenge of all of the above characteristics is ensuring the appropriate applications are installed for the appropriate users. In other words, if you have 1,000 users, it is the deciding how to identify the 35 of the 1,000 users that should get the Payroll application installed that is the challenge.

Group Policies are limited to 6 basic criteria you can use within the GUI: Domain, Active Directory Site, User, Group and Organizational Unit. This means you will either need to create "yet another group" as you did back in the NT days, or the more modern "yet another OU" in AD. That simply is not a good answer. Logon Scripts, depending on the scripting language you use, are far more robust in their ability to granularly select sets of users based on a much higher number of criteria. The problem with scripts is you need to write, test, debug, sandbox, test again and then finally deploy them. That's a lot of your time spent working just to get the Payroll app out to 45 users.

ScriptLogic's Desktop Authority provides unparalleled granularity in its ability to selectively deploy applications. Figure 4 shows Desktop Authority's patented Validation Logic technology which utilizes 50 different criteria, ranging from client OS, to group affiliation, to use within a virtual machine, to time ranges. Boolean operators (OR and AND statements) make these Validation Logic elements exponentially more granular providing essentially 50^n levels of granularity where n represents the number of Validation Logic elements you wish to add to narrow the focus to only the specific users needing the deployed application.

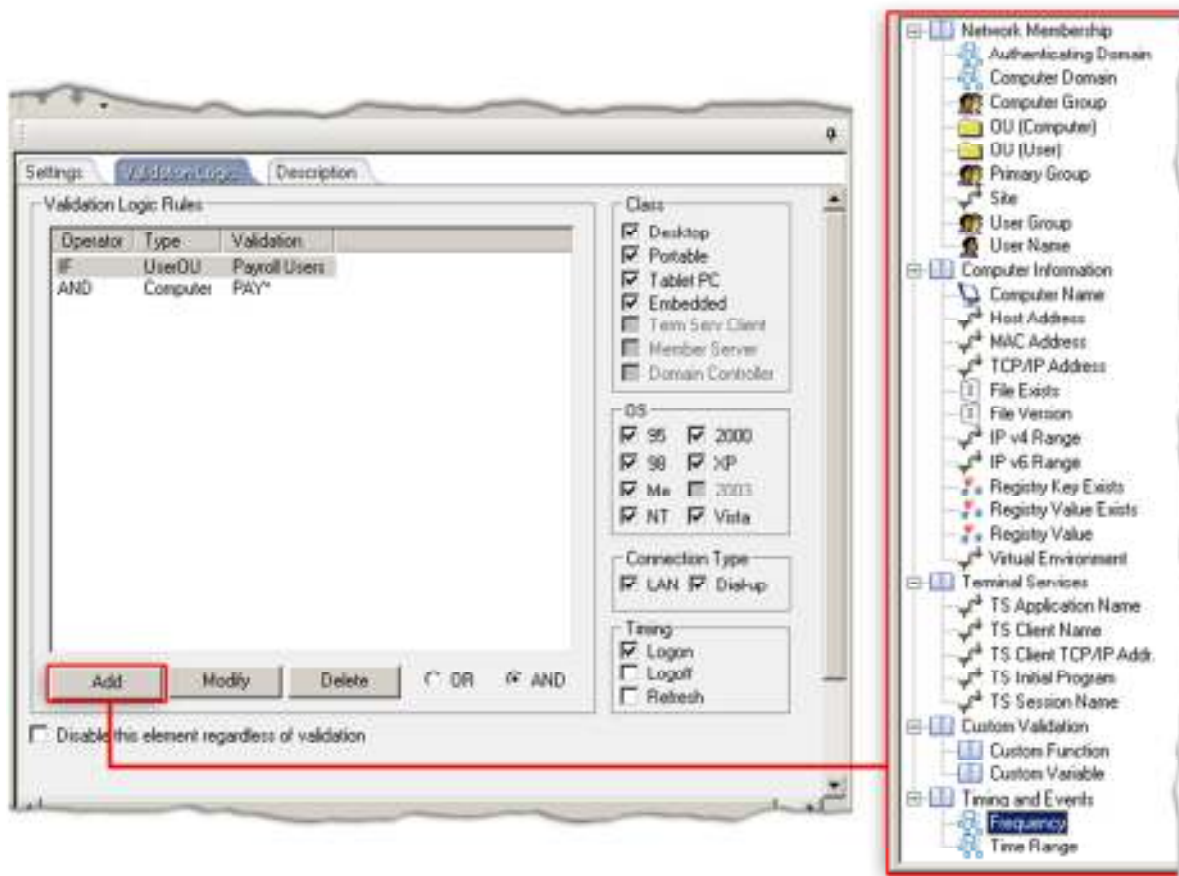


Figure 4: Granularly deploy applications with Desktop Authority's Validation Logic

Desktop Authority also incorporates a distributed MSI Repository on each server hosting its Update Service. As shown in Figure 5, each MSI you deploy to your Windows 7 desktops can be configured to either automatically select a server (where it simply finds a server running the Update Service in the same AD site as the client) or a distribution server can be specified. Bandwidth conservation can be accomplished using the Time Range Validation Logic element, which will only allow an application to be deployed within the time range specified.



Figure 5: Distribution points can be automatically or manually defined

The Proactive Migration – Deploying Apps: By centralizing the management of your app deployment and harnessing the power of Validation Logic (or equivalent if not using Desktop Authority), you complete the “Imaged OS with Deployed Apps” puzzle using a method that doesn’t just accomplish the same end result as an imaged OS containing apps; this method maintains the standard throughout the lifecycle, allowing the standard to change as the business needs change. The image remains clean, the most current apps are deployed, and all deployment is centrally managed and executed.

Sidebar: Deploying with System Center Configuration Manager

If you are using SCCM to deploy your applications, ScriptLogic's MSI Studio for SCCM integrates with SCCM to automatically publish MSI packages directly into SCCM, as shown in Figure 6.

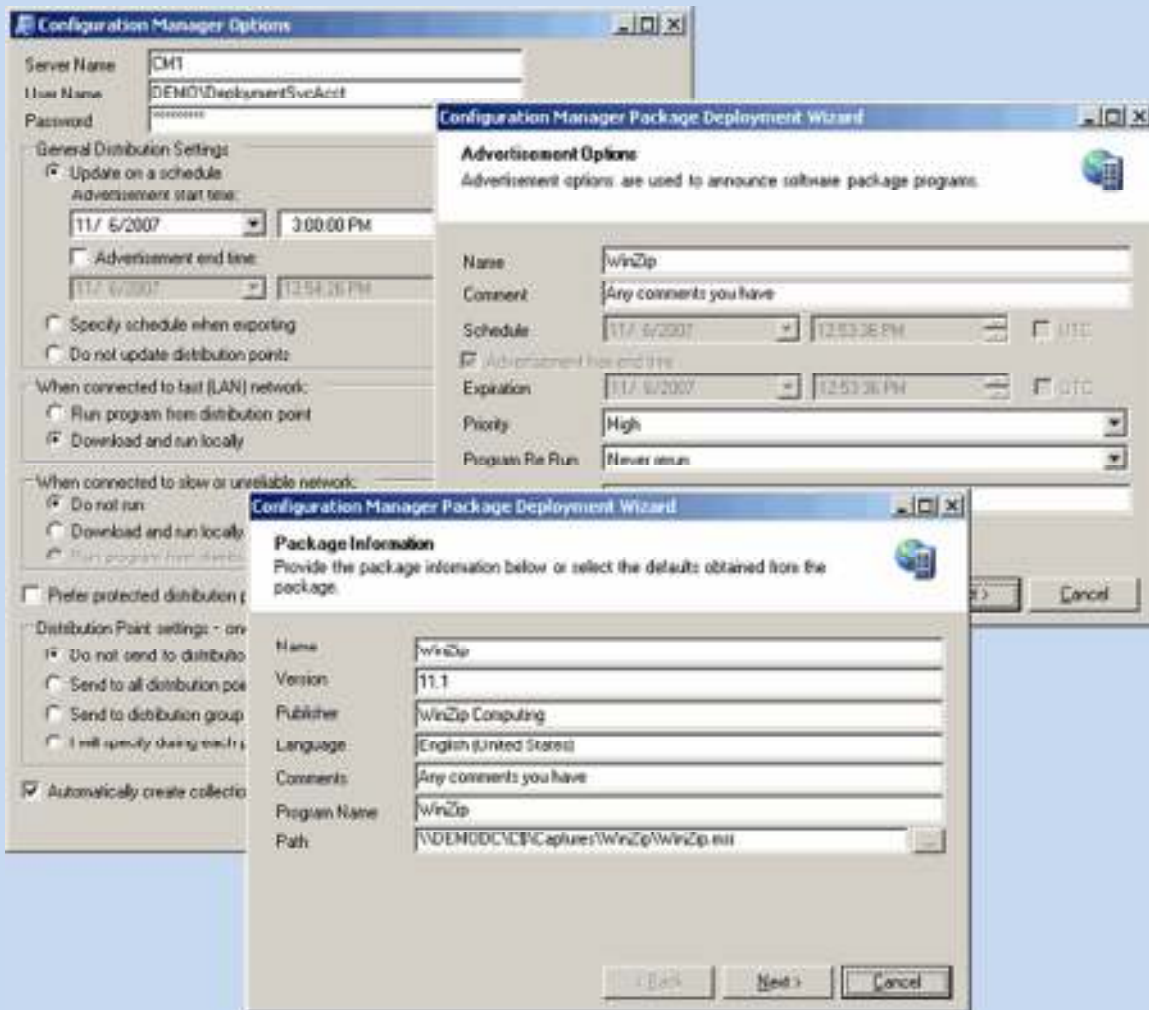


Figure 6: MSI Studio for SCCM publishes packages directly into SCCM

Making your Applications Work

One of the key aspects of application deployment that is usually not addressed is the issue of the users being able to actually use the application appropriately. There are a few aspects of this I want to discuss, but the overall theme is that you need to configure all the supporting aspects of an application to truly make it usable. Consider the following supporting aspects of an application required to make it useful:

- 1) **File System/Registry Permission Changes** – often installs work fine due to elevated privileges being used, but when a low-level user tried to run an application, it fails.
- 2) **Post-installation registry or INI file tweaks based on user location** – if you have roaming users, moving to another location may require modification of the configuration (e.g.: a server UNC path) to function.
- 3) **Supporting Drives, Printers, Desktop Shortcuts, etc** – Having the Payroll application installed without access to the Payroll database file and the Payroll printer may not cut it.

To truly deploy an application, you also need to deploy all of the supporting aspects of the user's desktop that enable the application to function. You can start with Microsoft's Standard User Analyzer (part of the Application Compatibility Toolkit) to test out the permissions issues that may be keeping an application from running. As you can see in Figure 7, you can select an application to test and see which files, registry entries, INI files, and more need security changes in order to have the application function.

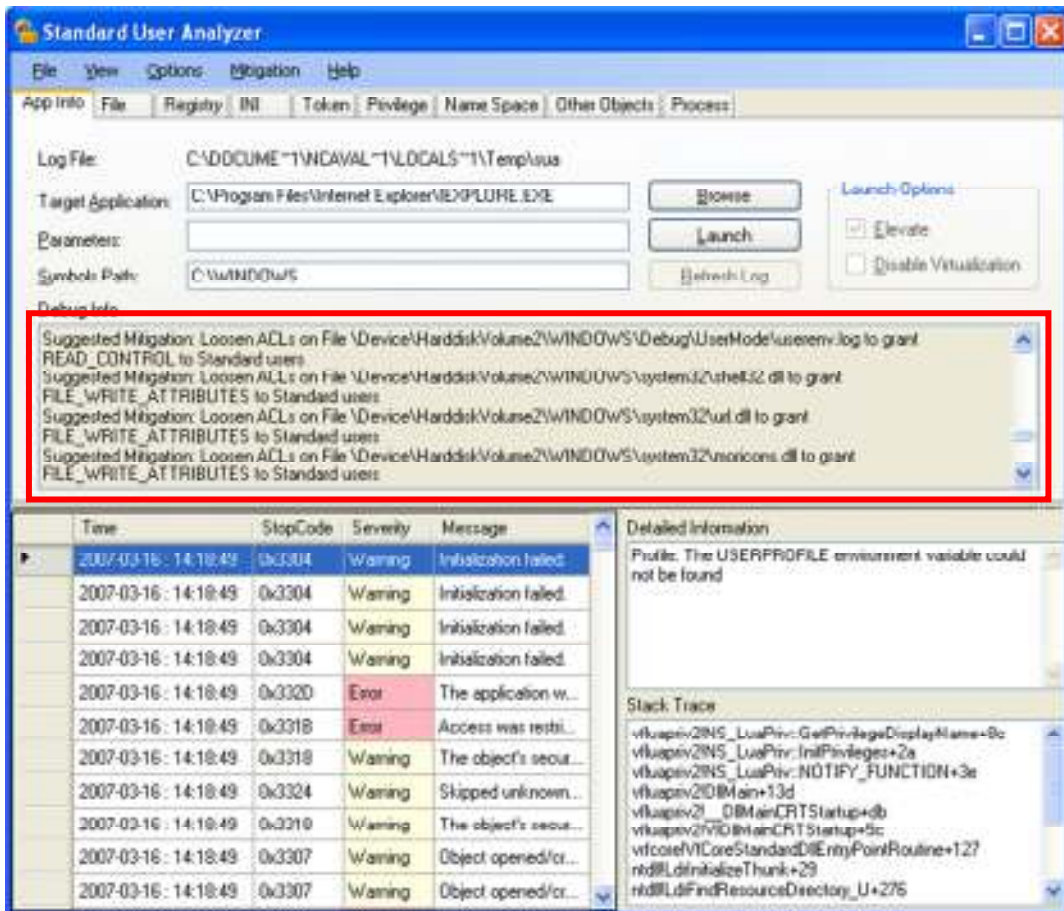


Figure 7: Testing Application Usage with the Standard User Analyzer

Even with this information, you are faced with how to implement the suggested changes centrally – the SUA will only suggest changes for the desktop machine it is running on! The same issue applies to the other types of configuration changes required I listed above (INI files, drive mappings, shortcuts, etc). Desktop Authority’s comprehensive coverage of every aspect of the desktop means you can centrally configure and deploy out all of the supporting settings to the desktops requiring these elements. Figure 8 demonstrates how you are able, in addition to deploying the application, to centrally establish each of the supporting elements with Desktop Authority that will complete the deployment and make the user productive.



Figure 8: Ensuring the application is usable with Desktop Authority

The Proactive Migration – Making Your Applications Work: Deploying an application by itself is only part of the story. If you deploy the application and each of the supporting elements you will have a **complete deployment solution**, requiring zero post-deployment tasks to make the application work. More on this in the next section...

Migrating Your User Settings

It is simply not enough to image the OS and deploy applications to the desktop. Beyond application-specific configuration needs, think about what makes up a user's desktop: security policies, firewall settings, drive mappings, shortcuts, registry tweaks, Outlook profiles, Office settings, power management and more; it is the sum total of each of these and other settings that make up a desktop. Without these settings included as part of the migration, user productivity will decrease and helpdesk calls will increase rapidly. To keep user productivity high without sacrificing IT productivity, your user settings migration will need to have the following characteristics:

- 1) **Migrates all settings critical to business function** – two parts to this one. First, the stuff that needs migrating to keep the user working needs to get moved. Second, that which is not critical should be left behind – for example, migrating the user's wallpaper of their dog is not critical to the business, and probably should not require IT's time (and therefore the company's money) to move it.
- 2) **Migrates centrally** – you most definitely should not be migrating users one-off at their desktop.
- 3) **Managed the configuration well after the migration** – this is key to keeping each desktop within the standard. If all you do is move settings from an old desktop to a Windows 7 desktop, you've lost control of the configuration once the migration is complete and you're back to one-off configurations, sneakernet and lowered IT productivity.

Microsoft provides the User State Migration Tool (USMT) which is a fancier version of the File and Settings Transfer Wizard within Windows XP. USMT allows you to centrally migrate user settings, files, printers, etc over to a new Windows 7 machine. This tool does have a few shortcomings in that it does not migrate every aspect of the user's desktop, but it is certainly a viable tool for accomplishing a high percentage.

But there is one issue with using a solution like USMT – it will require a lot of custom development (in the form of editing XML files to define what should and shouldn't be migrated) to preserve the standard environment. You have a beautiful standardized Windows 7 OS with standardized MSI packages that are deployed in a defined manner to only those users that require them. Then if you add in the USMT data without properly defining what needs to be migrated (that is, you simply bring everything over), you are going to dump all the non-standard user settings from the old machine to the new, making the new desktop a standard plus a bunch of non-standard settings. **Using any kind of “user setting migration” solution without defining the specific data to be migrated is self-defeating if you desire to maintain a standard desktop environment.** So how do you configure your user's settings on Windows 7 while still preserving the standard???

As long as you are willing to put in the time (I'm not trying to paint a negative picture of USMT here, just a realistic one), USMT provides tremendous flexibility via modification to its XML-based configuration files. For example, if there is a specific registry key that needs to be migrated, the following code can be added to a custom .xml file:

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
  <component type="Application" context="System">
    <displayName>Component to migrate only registry value string</displayName>
    <role role="Settings">
      <rules>
        <include>
          <objectSet>
            <pattern type="Registry">HKLM\Software\Microsoft\Windows\Example</pattern>
          </objectSet>
        </include>
      </rules>
    </role>
  </component>
</migration>
```

As you can see from the example of including just one registry key, USMT is flexible, but will require a significant investment of time if you desire to migrate only those settings that adhere to your standardized environment.

Desktop Authority takes a completely different approach to this problem resulting in essentially migrating the user's settings before you ever install a single copy of Windows 7. Let me explain by first discussing what user settings Desktop Authority is capable of centrally managing and then I'll apply it to the last step in the Windows 7 migration.

If you take a look at Figure 9, you will see over 30 different aspects of the user's desktop settings that are all configurable from an easy-to-use graphical interface. Each of these configuration objects represents some part of the user experience. In sum, they make up everything that is the "user's desktop". Each configuration object supports multiple entries to facilitate different needs, with each entry having its own Validation Logic to designate the subset of your organization's population that should get the configuration setting.

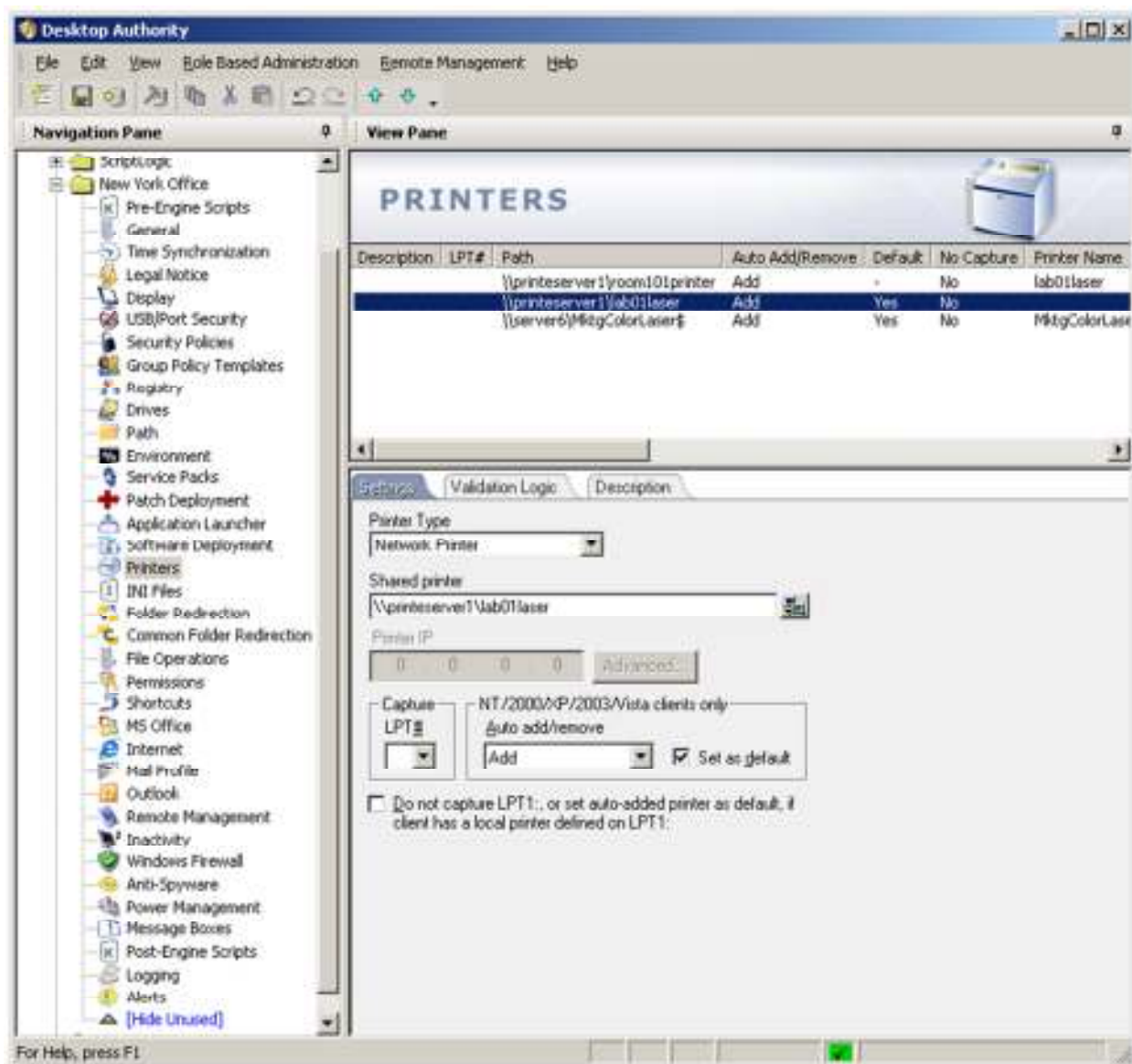


Figure 9: Comprehensive User Settings Management with Desktop Authority

In a “migrate the user’s settings solution” scenario, you are simply moving what they had to their new desktop. **This is a short-term solution.** That is, it is easy to do now, but will cause more work for you in the long run. The problem with migrating the user settings is that all of the configuration data will reside on the new desktop - and that desktop only. Should there be an issue with that new desktop (it crashes, the disk dies, etc), or the user needs to log into the network from another floor, building, etc., none of their settings will be accessible from another desktop leaving you with a tremendous amount of work to get the user productive with all their apps, settings, printers, drive mappings, etc and all that work for a temporary amount of time until they return to their original desktop.

In contrast the “migration” of user settings actually happens before the OS migration with Desktop Authority. You begin by centrally configuring your user’s desktops from within Desktop

Authority. Each aspect of their desktop no longer is singularly set on a specific desktop; instead it is centrally controlled so that there is nothing specific about the user on any given desktop. The desktop essentially becomes an appliance, allowing your users to freely move from desktop to desktop without needing to call you to help set them up each time they move to another machine. **This solution provides you the same short-term goals, but with long-term lasting results.** When the user “moves” from their XP desktop to the newly installed Windows 7 desktop, they will simply log on and their apps, settings, drive mappings, printers, Outlook settings, IE settings, Group Policies, security settings and more will all be established for them.

The Proactive Migration – Migrating User Settings: With a “migration of settings” type of migration you are putting yourself into a very reactive situation where, in the future, you will be required to perform more work since the user configuration will only reside on a single desktop. With Desktop Authority, you are establishing a standard user experience that requires no additional work on your part in the future, even if the user needs to move to another desktop!

Table 1 lists the basic data sets that USMT migrates to Windows 7. You’ll notice that Desktop Authority, while supporting each of the data sets, does not actually migrate them. Instead it manages them by creating a working environment that facilitates long-term management of the user’s desktop settings.

Data Sets to be Migrated	Supported by USMT?	Supported by Desktop Authority?
My Documents	Yes	Yes via Folder Redirect to Server
Desktop Files	Yes	Yes via Folder Redirect to Server
Favorites	Yes	Yes via Folder Redirect to Server
Start Menu	Yes	Re-established via new standard
“All Users” profile folders	Yes	Yes via Common Folder Redirect to Server or reestablished via new standard
Files (outside of My Documents)	Yes	Yes via Copy Operations object
Access Control Lists	Yes	Re-established via Permissions

Table 1: Data Set Migration Support

The first four data sets are addressed via Folder Redirection. This is a more proactive measure for true desktop machines. Laptops will obviously need to have the data migrated, which still gives USMT viability within the migration. But for desktop machines, redirecting the most important data sets to a server has a few advantages:

- 1) **Backups** – server-based data can be more easily backed up with fewer desktop agents, issues of desktops being turned off, etc.
- 2) **Roaming Users** – moving the data to a server means that a user can sit anywhere within the organization and access their documents, settings, favorites, and more

Other settings are handled by establishing a new standard. The idea is not to create more work for you, but instead to set the standard once within Desktop Authority, and then no longer need to

address the issue. So for example, setting up the Start Menu shortcuts for the new Windows 7 desktop/Start Menu/etc, you'd simply and quickly use Desktop Authority's Shortcuts object and create new shortcuts in each of the locations required, as shown in Figure 10.



Figure 10: Even shortcuts make up part of the new desktop standard

When Should You Get Started?

Recent Gartner research notes state that organizations should wait for Windows 7 SP1 before migrating (which they estimate at **12 to 18 months after the release of Windows 7**) Migrations will need to start well before the actual deployment, noting that standardized OS creation, app testing and performing a pilot test will take up to 9 months. This thinking lines up exactly with the Proactive Migration using Desktop Authority. The message is simple and clear:

- 1) **Implement Desktop Authority now** – using it as your proactive desktop management platform for desktop configuration, inventory, security, and support.
- 2) **Begin to establish the standard for your desktops now** – begin with those settings that you can manage today: drive mappings, printers, etc and then begin to bring each application, configuration, setting and registry tweak into the fold, preparing for Windows 7.
- 3) **Migrate the OS** – when you are ready, you can focus on the installation of Windows 7 only.
- 4) **Have your users log onto their new desktop** – the first time they log on, every aspect of their desktop – applications, settings, desktop environment, printers, and more – that you

have centrally configured with Desktop Authority will be pushed to the new desktop (and every other desktop that user logs onto).

Desktop Authority: The Proactive Migration Solution

In this whitepaper, I've discussed the basic steps needed to migrate to Windows 7, pointing out the reactive mistakes many make that only cause more work in the future. Using the Desktop Authority family of solutions facilitates a seamless migration to Windows 7 while establishing a desktop standard that will serve the interests of your users, the business you support and IT's desire to be productive for years to come.

Resources

- **Windows 7 Compatibility Center** - www.microsoft.com/windows/compatibility/windows-7/default.aspx
- **Application Compatibility Toolkit** - technet.microsoft.com/en-us/windows/aa905066.aspx
- **Virtual PC and XP Mode** - www.microsoft.com/virtual-pc

More information on the ScriptLogic solutions listed in this document, as well as a 30-day fully-functional evaluation is available on our website.

- **Desktop Authority** – www.scriptlogic.com/da
- **MSI Studio** – www.scriptlogic.com/msi

About the Author:

Nick Cavalancia, MCSE/MCT/MCNE/MCNI, is ScriptLogic's VP of Windows Management where he assists in driving innovation and the evangelism of ScriptLogic solutions. He has over 16 years of enterprise IT experience and is an accomplished consultant, trainer, speaker, columnist and author. He has co-authored and contributed to over a dozen books on Windows, Active Directory, Exchange and other Microsoft technologies, and is the author of "Microsoft Exchange Server 2007: A Beginner's Guide."