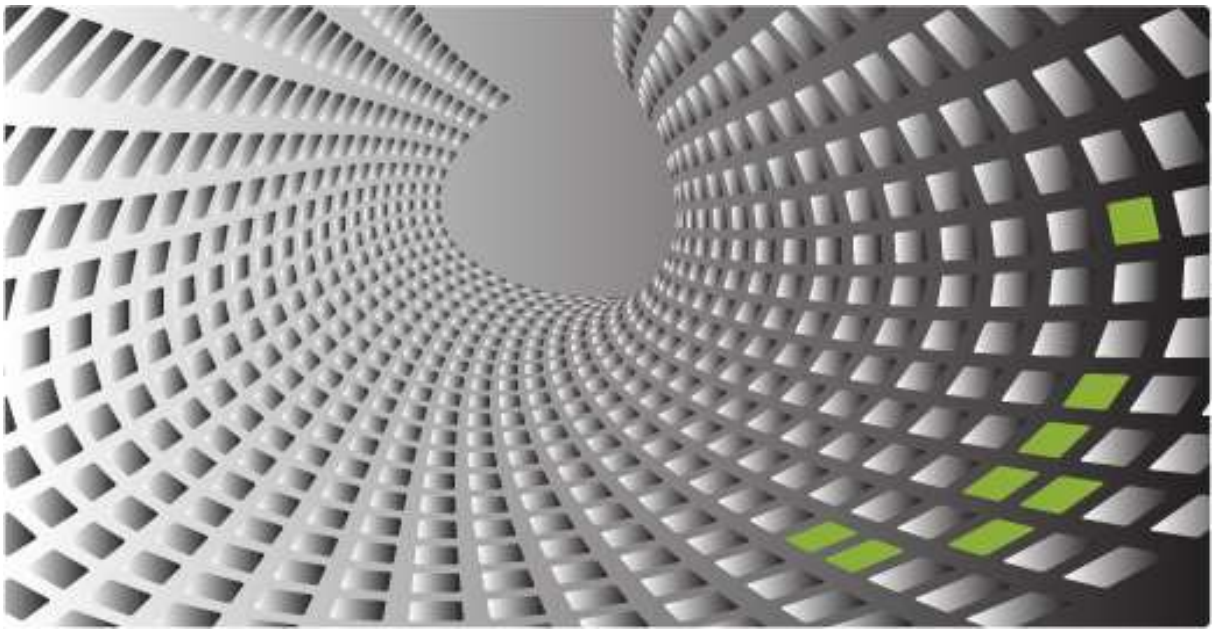

Immidio White Paper – Things You Always Wanted To Know About Windows Profile Management

Abstract

Why are Windows user profiles so critically important for corporate IT environments and how can they be managed efficiently? If you always wanted to have a profound answer to this question, this white paper is for you. Be prepared to get some technical insights that go beyond what Microsoft calls User State Virtualization. The successful management of user profiles turns out to be of growing relevance, both for physical and virtual Windows desktop infrastructures.

Version 2

Things You Always Wanted To Know About Windows Profile Management

Windows user profiles are about individual workspaces, each one reflecting its owner's personality. Such user profiles may well include business critical information represented by unique, user-specific data and settings related to desktops and applications. In the past, when users only needed to deal with one physical desktop and one profile, things were relatively simple. But now they tend to have access to corporate and private applications through multiple physical desktops.

In addition to that, remote desktops and the advent of virtual desktop infrastructures are making things even more challenging. Using a wider spectrum of applications from multiple desktops, with each desktop optimized for dedicated tasks, is the way how I'm working today both at work and at home. And I don't regard myself as an early adopter. I'm convinced that many people around the globe use computers in pretty much the same way I do. And most of them are struggling with their Windows user profiles.

User Profile Fundamentals

At a first glance, Windows user profiles seem to be relatively simple objects containing registry and file system information in a predefined hierarchical structure. Such profile objects are like container boxes, used to preserve all application settings and system configurations for individual users. By default, each profile – a predetermined folder structure and accompanying registry data – is stored locally in a user-specific system folder, with the file Ntuser.dat containing the relevant registry data. Windows loads the profile when the user logs on and unloads the profile when the user logs off. At logon, a user's previously stored profile information is applied to his or her interactive user session, commonly called the "desktop". At logoff, all user-specific settings are saved persistently in the profile folder structure. If a user logs on for the first time, a default profile is used as a blueprint for creating a new individual profile for future usage. It is important to note that exactly one individual "container box" folder structure represents a complete profile, independent of the number of applications used or desktop settings modified by the associated user.

Saving and restoring Windows user workspace settings and profiles is not as simple as it seems to be. Several predefined steps need to be done properly and in the right order during the logoff and logon sequences, taking many aspects into account, such as Explorer settings, language preferences, themes, drive mappings, printer associations, icon positions, wallpaper selections, taskbar configurations, documents stored on the desktop, and application settings.

Locally stored profiles are perfectly suited for home users with only one single computer or for users with laptops that are not permanently connected to a network. Many corporate users, however, are facing a different situation. In many enterprise IT environments, users need to roam between locations and client devices, meaning that they are using multiple computers for work. In addition to that, remote desktop and

virtual desktop concepts may introduce additional platforms corporate users are interacting with on a regular basis. It is easy to understand that multiple locally stored profiles result in an inconsistent user experience in such environments.

Storing individual users' profiles on a central file share is the solution to this problem although requiring a little extra system configuration. Centrally stored profiles are commonly referred to as roaming profiles. They can be configured through per-machine group policy settings or through the attributes of Active Directory user objects. Even if the assignment of roaming profiles through a group policy is preferably, this is not possible for all Windows versions. Roaming profiles for Windows XP and Windows Server 2000 Terminal Server can only be configured through Active Directory user account settings.

Roaming profile data is copied from a central file share during logon and transferred back to the file share during logoff. This concept of roaming profiles is particularly popular in corporate networks. In case the roaming profile is inaccessible during user logon, a temporary profile with default settings is generated instead – but all profile modifications are discarded at logoff.

If all was good, the story would end at this point. But unfortunately, user profiles produce a number of challenges in many Windows environments.

1. Profile corruption: During an interactive user session drivers and applications read and write registry values. When the user logs off it may happen that open connections to the registry don't close, preventing the user's individual portion of the registry to unload properly. In such a case Windows cannot copy the profile back to the central file share, producing an error or a profile corruption when the user logs in next time. In the past Microsoft provided the User Profiles Hive Cleanup Service (UPHClean) as a solution which checks for open registry connections during logoff and cleans them up. Windows Vista and Windows Server 2008 introduced the built-in User Profile Service, providing the same functionality like UPHClean. But still rogue registry connections initiated by applications, service processes or drivers may lead to profile corruption. Unfortunately, creating a new user profile is the only way to solve a corrupted profile issue in many cases, resulting in lost personal data.

2. Profile bloat: There are several reasons why user profiles may grow significantly over time. Uncontrolled profile bloat is caused by growing numbers of customized application and system settings, by sprawling icons and documents on the desktop and by remaining data from uninstalled applications. As a result the performance of user logon and logoff sequences may degrade significantly over time, leading to unacceptable logon times in many cases. This effect is even amplified when user profiles include many small files due to the fact that file servers are optimized to transfer only large files at the fastest possible network speed. All together these issues may create an unacceptable user experience during logon and logoff.

3. Profile incompatibility: The original Windows profiles version – referred to as V1 – is used in Windows 2000, Windows NT, Windows XP and Windows Server 2003. With Windows Vista and Windows Server 2008, Microsoft introduced Windows profiles version 2, changing standard profile path, folder names, localization concept and management of application-specific data. Although the changes may

have been necessary for several reasons, the different profile versions are incompatible, resulting in the fact that they have to be maintained side-by-side in mixed environments. This means that you cannot use a Windows XP profile when logging in to Windows 7.

4. Mix of system and application settings: Settings of the operating system and all applications are stored in one profile container. In other words, system settings and application settings are not decoupled. Due to this fact standard Windows user profiles are likely to create issues when using them across different operating system versions.

5. Late profile write: The concept of roaming profiles was designed while assuming that a user is only logged in to one interactive Windows desktop at a time. Roaming profiles are not copied to the central file share until logoff. This means that the most recent versions of files and settings of roaming profiles are only cached locally and not transferred to the central file share as long as the user doesn't log off. A user simultaneously working on multiple physical computers or virtual desktops may start and terminate multiple applications in random order from different desktops. Write conflicts are hard to avoid in such a scenario, most probably producing inconsistent profile data. This is also referred to as the "last write wins" issue.

This list makes it obvious why many Windows administrators are struggling with centrally stored user profiles. Now the question is, how can you master such beasts?

Solutions Provided by Microsoft

Microsoft is well aware of the profile issues described earlier. This is why they provide a number of solutions. UPHClean and the newer User Profile Service are intended to fix the issue of corrupted profiles. But unfortunately they can only resolve some of the most annoying symptoms and do not work at the level of the root cause.

The most popular solution suggested by Microsoft to fix the problem of profile bloat is to redirect user shell folders. This is accomplished by rerouting folders like Desktop, Personal Start Menu and Application Data to a network share outside of the user profile, preventing them from being copied at each logon and logoff. The path of each user shell folder is defined in the registry under HKCU \Software \Microsoft \Windows \CurrentVersion \Explorer \User Shell Folders which can be modified manually. It is, however, recommended to configure folder redirection through a group policy. This is done under User Configuration \Policies \Windows Settings \Folder Redirection in the console tree when using the Group Policy Management Console.

Folder redirection results in a reduced size of the user profile and, as a consequence, in shorter logon and logoff times. It may also reduce the effects of the "last write wins" issue described earlier. On the other side, some of the redirected files may be required at application launch. In this case, depending of the network bandwidth, the time to launch an application may increase significantly. In addition to that, the permanent load on file servers and the network utilization is higher when user shell folders are redirected. In other words, the problem is not solved; it is only shifted to a different time within a user session or to a different place in the IT environment.

There are even more options to control the user profile through group policies. Excluding folders in roaming profiles is one of them, discarding selected files and preventing them from being saved on the central store is another. But this also means that the files are not available when the user selects another workstation or virtual desktop at the next login.

Incompatibilities between version 1 and version 2 of user profiles can be addressed by the User State Migration Tool provided by Microsoft. It consists of two command line programs to copy user files and settings from one Microsoft Windows computer to another. USMT 4, which allows user state migration from Windows XP and Windows Vista to Windows 7, is included in the Windows Automated Installation Kit. One program, named Scanstate, scans the source PC for user accounts, files and folders, Windows settings, and program data files and settings. It stores the results in a file which is then used by the second program, named Loadstate, to transfer all scan results to a target PC. Although USMT is very powerful, it has a complex syntax which prevents it from being a straight-forward solution for user profile migration. In addition, it cannot be used to transfer user profiles from Windows 7 down to Windows Vista or Windows XP which makes it incompatible to scenarios where a concurrent support of multiple Windows versions is required.

If a user profile is corrupted or if it grew to an unmanageable size, it may become necessary to delete it. But this is not as simple as it seems to be. In Windows Vista, Windows 7 and Windows Server 2008 (including R2), deleting a user profile in the %USERPROFILE% folder results in an error message the next time the user logs on, saying that he was logged on using a temporary profile. The bad news is that upon logoff a temporary profile is deleted and all personal profile data is lost. Using temporary profiles in such a case can be seen as an emergency procedure initiated by Windows when the user profile cannot be loaded, but resulting in an undesired system behavior. All this is due to the fact that Windows keeps track of local profiles in the registry key HKLM \SOFTWARE \Microsoft \Windows NT \CurrentVersion \ProfileList. For each profile a subkey with the name set to the profile owner's SID is maintained in this list, which is keeping track of the corresponding profile directories. Before creating the temporary profile, Windows renames the original ProfileList subkey to SID.bak.

Windows XP and Windows Server 2003 are more simplistic here. If a ProfileList registry subkey exists but the corresponding profile path is not accessible, they simply create a new local profile. As a result the data in the ProfileList key is overwritten, making the original profile useless without logging a corresponding error message in the Windows event log. For proper profile clean-up it is recommended to open the control panel applet "System Properties" (by running sysdm.cpl) and deleting the profile from there.

At this point you may ask yourself why Microsoft does not completely change the way profiles are working. The primary reason is compatibility. For all their operating system modifications and enhancements, Microsoft needs to make sure that they work for a large range of users, applications and computers. User profiles are such a critical component hooking into so many other operating system components that the underlying architectural concepts cannot be changed so easily. This is why there is enough room for several third-party vendors providing user profile management solutions addressing specific technical challenges or market segments.

Profile Streaming and Profile Segmentation

There are two fundamentally different methods when managing Windows user profiles in a way that goes beyond the standard solutions provided by Microsoft: profile streaming on one side and profile segmentation on the other. Although different, both methods aim at solving the profile challenges described earlier.

Profile streaming – also referred to as profile virtualization – redirects all local profile read and write operations to an alternative location in the network. Technically, this can be done through a filter driver or a service communicating with a database, a streaming server or a file server in the network. The advantage of this method is that its initial installation and configuration is simple. Instead of exchanging profile data with the local profile cache, the Windows system reads from and writes to a central profile store. In addition to that, profile corruption can be eliminated by using advanced database storage concepts rather than an old-fashioned container box file structure including one single file with all registry settings.

With profile streaming, a user's profile is updated in real time when he or she is running multiple sessions at the same time, turning the "late profile write" issue described earlier into an "immediate profile write" advantage. If a user creates a new document in one session it automatically pops up in the other one. Changing system or application settings from two different user sessions immediately saves the new settings at a central location, avoiding the "last write wins" issue at user logoff.

The disadvantage of profile streaming is that profile bloat is not necessarily prevented without introducing advanced filtering functionalities and configuring corresponding exclusion rules. Just streaming unfiltered profile information means that system and application settings are still unmanaged. This still results in large profile archives consuming significant network resources, even though logon and logoff times may have been affected positively by moving profile data transmission to the time when required by the application. Depending on the nature of the backend system, offline capabilities may be very hard to implement or require additional local services.

It is interesting to note that profile streaming cannot meet the expectation that modified application settings are always reflected to applications running in different user sessions immediately. But why is that? On changing a specific application setting, a user would expect that these changes are reflected to the same application running in another session that user is logged in to. In fact, this is not the case for most applications as they only read profile information at launch and save modified settings at termination. Updating settings during application runtime is just not coded into the logic of most application. Permanently streaming modifications of application-specific profile portions does not change this application standard behavior. This means that there is still a chance of running into unexpected conflicts when storing application settings from multiple sessions.

Depending on the implementation, immediate profile streaming may also introduce challenges when dealing with open documents included in the user profile. A good example is a Microsoft Word document stored on the desktop. When opened, such a document creates temporary files that are instantly available on other desktops through profile streaming, creating additional network traffic but being completely useless in the target sessions. When a user wants to open the already opened document from another session, the permission to access the file for read and write operations will be denied by

Windows preventing a data synchronization conflict. The result is a read-only error dialog box that does not provide additional information about the nature of the problem. The behavior is pretty much the same as it can be observed when conventionally redirecting the user profile's document folder to a central file share.

At this stage it becomes very obvious that profile streaming or profile virtualization has a great potential to solve many profile issues, but still has its limitations. It is extremely powerful when an easy-to-install solution is required that solves challenges encountered with multiple concurrent sessions used by one individual user. However, profile bloat, profile incompatibility and issues with the mix of system and application settings cannot be solved with the concept of profile streaming alone.

In contrast to profile streaming, the concept of profile segmentation – also referred to as atomic profiles, profile granularity or profile decoupling – introduces a completely different method when managing user profiles. Segmented profiles work on all Windows versions supported by Microsoft today, both 32 bit and 64 bit. The general idea behind this concept is using a rather static base profile for common settings and store individual system or application settings in individual files or archives. This means that all settings of one application are stored in its own archive while settings of a second application are stored in another archive, with all archive files stored in a central file share – ideally the user's home directory. An individual application's profile archive file is created on the same platform where the application is installed.

This allows storing the complete, application-specific profile archive in a single container file, preventing the transmission of a potentially large number of small files containing application settings and data. The same concept can also be applied to sets of desktop settings, providing the option to separate printer settings from Explorer settings or wallpaper settings and saving these to individual files. Combined with smart compression algorithms, profile segmentation reduces the size of user profiles significantly, eliminating the effects of profile bloat.

Preselected groups of profile segments can be stored and saved during user logon and logoff. Profile settings related to individual applications – independently if installed locally or using application virtualization – can be applied at application launch and saved at application termination. In combination, this allows full control over each user's profile, leading to minimum logon and logoff times and requiring minimum network resources. Offline scenarios can be covered easily by using standard file replication mechanisms provided by Microsoft.

Profile segmentation doesn't come for free; it requires the administrator to configure all desktop, system and application settings to be taken into account, enabling the integral parts of a user's profile to be transferred to the central storage. If done smartly, this leads to resolving the incompatibility between V1 and V2 user profiles and all issues related to the mix of system and application settings. Profile segmentation supports user sessions running side-by-side on local Windows, Remote Desktop Services (aka Terminal Services), Citrix XenApp or virtual desktops. In addition, file corruption cannot influence complete user profiles anymore as it only may affect individual application or desktop settings – making profile corruption a problem of the past.

Immidio Flex Profiles

Immidio Flex Profiles is today's most popular profile management product implementing profile segmentation. The graphical Flex Profiles Management Console helps administrators and IT professionals to interactively configure all necessary settings. The Immidio Flex Profiles Assistant – embedding Microsoft SysInternals Process Monitor – is the ideal product component for identifying the relevant profile information that needs to be taken into account for individual application or desktop settings. For more details, check out www.immidio.com/flexprofiles.