The Easy Button – A story about Corruption, Portability, Office 365, User Profile bloat, ProfileDisk, non-persistent Data Indexing and more…

20 Min read – User profiles have been around for many years. According to Wikipedia the user-profiling scheme in force today owes its origins to Windows NT, which stored its profiles within the system folder itself, typically under C:\WINNT\Profiles\. Windows 2000 saw the change to a separate "Documents and Settings" folder for profiles, and in this respect, is virtually identical to Windows XP and Windows Server 2003.

Of course, the way that user profiles behave, interact with the underlying Operating System, the different 'types' of profiles available and how they are managed has also drastically changed throughout the years. With many more advancements, currently being developed and on the way.

A few topics I'd like to touch on:

1. Large user profiles
2. Office 365 and its drawbacks, on and offline caching
3. The concept of a ProfileDisk
4. Data indexing for non-persistent
   - *But wait, that's not all – VHD containers*
   - *Size does matter*
   - *Resume*
5. Profile Portability
   - *A bit more detail*
6. A unique combination – ProfileDisk + Profile Portability
   - *Data outside of the user profile*
   - *User-authored data*
7. Profile/data corruption, then what?
8. How to handle existing (Roaming) profiles
9. Infrastructural needs and components overview
10. Integration with ProfileUnity
    - *ProfileUnity agent*
    - *Main configuration steps*
11. Concluding – Why we are different

Large user profiles

Historically, the size of a user profile has always been an issue. Large user profiles take a long time to load making them far from ideal to roam between multiple machines. Especially when working in non-persistent environments this can lead to frustrations. While today there are many reasons (technical as well as non-technical) that can lead to user profile bloat there is one, in particular, I'd like to focus on – Office 365. A challenge more and more companies are dealing with and besides that, it's a perfect example to explain the added value of our ProfileDisk technology.

Office 365 and its drawbacks, on and offline caching

By now, most of you are probably aware of the issues that arise when using Office 365 on non-persistent machines. If not, here's what I'm referring to… Office 365 can be used in two modes; online and offline. When using Office 365 in online mode you have (and need) a direct live connection to Azure where you Exchange e-mail inbox and Outlook Address Book reside.

However, when using this type of set up performance is usually far from ideal. Everything needs to happen near real-time within the online copy of your mailbox and thus a high quality, low latency connection to Azure will be needed to keep things running smoothly. Unfortunately, for many companies today this is still far from a reality. That's why most revert to offline 'cached' mode – which is advised by Microsoft as well.

In cached mode, a local copy of the user's mailbox, including the Offline Address Book will be stored in an offline data file a.k.a. an .OST file, which, by default resides in the user's profile at: C:\Users\<username>\AppData\Local\Microsoft\Outlook. Meaning your users will have direct access to all data, locally without the need for an ongoing network connection. At the same time, when a network connection is present the user's mailbox and address book will be updated continuously.

This all sounds, and works great until we apply the same principle on non-persistent machines, never mind if it's a VDI or an RDSH type of setup. In most cases users are randomly assigned to a virtual desktop, meaning that the user's session is re-created each time a user logs on, and is discarded during logoff, also deleting any data stored locally during the session.

Because of this, storing data locally is not an option and keeping the .OST file, as previously highlighted in the user profile also isn't a preferred approach as it will bloat the user profile and greatly increase the time it will take to (re) load the user profile onto each 'new' machine over and over again.

The concept of a ProfileDisk

This is where our ProfileDisk solution shines. The concept, at least from a high-level perspective is simple: we redirect the entire user profile, including the registry to a virtual disk file, which can either be a VHD or a VMDK.

*The Office 365 caching issue is just one of the many benefits that our ProfileDisk technology brings to the table, below, under **'resume'** I'll highlight a few more.*

Once configured a user profile will automatically be created on the virtual (Profile) disk and connected to the Operating System just before the user fills in his or her username and password. Because of this approach, the Operating System will immediately see and recognize the user profile as being local and as a result won't load a default or other type of user profile instead.

By storing all user profile related data onto a virtual disk, it will be seen treated as single file as opposed to many (100s) separate files and folders, including the additional SMB/CIFS overhead (request, respond, challenge, negotiate, accept open, read, write, close etc. per file, per individual action) that comes with it when using folder redirection together with Roaming Profiles, for example.

*Data will be available to the Operating System almost instantly after the user logs in – one of the main reasons why ProfileDisks are suitable to handle large user profiles. We also refer to it as the profile '**Easy Button'***

Just be aware that although folder redirection is no longer needed, it can still be applied as part of your best-practices approach for certain shell folders. In fact, this is made extremely simple since our ProfileDisk technology is part of ProfileUnity – a full-blown user and workspace management suite.

With a ProfileDisk you basically create a user persona/data abstraction layer, which is stored and managed in a centralized manner, making the life of your IT administrator a little easier while achieving the same performance of a local user profile. Data will be natively available to the Windows Operating System without any filters or drivers in between ensuring performance will be optimal.

Data indexing for non-persistent

Everything that happens on a ProfileDisk is persistent, enabling you to create a persistent experience even when most of your users work on non-persistent VDI and/or RDSH type machines.

This applies to data indexing as well. Normally, the Windows Search Service indexes the .OST and creates an index catalog (database local to the system it resides on) to enable search functionality in Outlook – it does the same for other Windows related data as well, depending on your configuration. Although for non-persistent machines this is irrelevant since most, if not all data relevant to the user will be stored on the network, so there isn't really anything to index locally.

Since, in non-persistent environments, this index/database will be 'destroyed' each time a user logs off, or when a system is rebooted all data needs to be re-indexed repeatedly, every time a user logs in, claiming local compute resources in the process. In other words, the index/database does not roam in any way. This is also one of the main reasons (and best practice) why search/indexing is disabled within non-persistent environments.

Good news! Our ProfileDisk/ProfileUnity technology enables the Outlook search index/database to roam, on single as well as multi-user systems – And remember, all data part of a ProfileDisk, including your .OST file is persistent, meaning indexing will take place once and that's it.

But wait, that's not all – VHD containers

ProfileUnity consists out of multiple modules, one of the modules it offers is VHD containers. A simple, yet effective way to offer individual and/or additional VHD disks to your users – these are separate from ProfileDisks, and while both are based on VHDs the technology involved is different. For example, VDH containers can be used for the purpose of caching additional data, like that of OneDrive for Business, or Google Drive, or OneNote, or… you name it.

Size does matter

When using Office 365 in offline mode, as mentioned a copy of the .OST file will be cached locally in C:\Users\<username>\AppData\Local\Microsoft\Outlook. Nowadays having an inbox of multiple GB's in size isn't uncommon, keep this in mind when sizing your file servers where your users' ProfileDisk will be stored.

To help with this you can control the number of months, and thus the amount of data downloaded into your offline .OST file. For example, if you configure the Outlook offline settings to three months, only mail/data from the past three months will be synchronized to your offline .OST file. This as opposed to the default being a year if the disk where the .OST resides on is equal or larger than 64 GB. Of course, indexing will also be limited to the same three months of data, all data/items older than three months will only be available/searchable online, in Azure.

Using the above method might not be ideal in all cases, but at least you have something to play with. Also, when ProfileDisk is thrown into the mix, we can apply compression as well, which will account for another few percent of additional space savings.

Resume

ProfileDisk supports virtual desktop environments such as Citrix XenDesktop and VMware Horizon View, physical desktops as well as cloud and DaaS platforms, and while doing so it provides the following key benefits in these environments:

- **Windows and Microsoft Outlook/Office 365 Indexing and Search** – Windows Indexing is usually off in VDI environments because it negatively impacts user performance. ProfileDisk enables indexing and search to remain on with its high-performance capabilities so that user experience is not impacted even when these Microsoft features are enabled.
- **Microsoft Office 365 Cached Mode –** Office 365 requires caching within the user's profile. ProfileDisk seamlessly enables Office 365 functionality in virtual desktop environments.
- **Large Profiles –** Often user profiles may need to remain large because of applications, such as Office 365, Skype for Business, and Lotus Notes, that may write to profile files. ProfileDisk handles these applications with ease.
- **Speedy Logon times in Non-Persistent/Stateless VDI environments** – Non-persistent VDI environments do not retain user profiles on their own, mandating that large portions of a user profile need to be written at login or throughout the user session. This non-persistence can cause poor login performance or slow application response times. ProfileDisk eliminates the

requirement for large profiles to transfer to end-user desktops, thus significantly speeding up login times and enabling great session performance.
- **Native Windows Profile Performance** is supported by ProfileDisk without requiring a file system filter driver which may slow down native profile performance.

Profile Portability

Portability provides us with a unique toolset helping us achieve the following:

- The creation of a universal Windows user profile. Making it possible to work and roam on and between multiple versions of the Windows Operating System using a single version user profile – desktop and server.
- Assists in the migration, or on-boarding process of existing user profiles onto a ProfileDisk.
- It helps protect against profile corruption, providing a backup, or failback mechanism when needed.

A bit more detail

Before explaining how our Profile Portability engine assists with the on-boarding of existing user profiles or how it helps in protecting against profile data corruption I'll first (briefly) highlight what it is all about.

As you might be aware, with new versions of the Windows desktop and server Operating Systems also come new (er) versions of the user profile.
Years ago, as part of Windows XP and Server 2003 we started out with user profile version 1.0 – the technology used by Microsoft Windows to save, load, edit, manage and display the Windows user profile. Today we are at version 6.0 as part of Windows 10 and Server 2016.

The issue with this is that different versions of the Windows user profile are not compatible with each other. Meaning that a user profile originated on a Windows 7/Server 2008 R2 machine cannot be used on a Windows 8/Server 2012 machine or vice versa.

When migrating to Windows 10 the same issue arises. Again, the profile version changes from version 2 in the case of Windows 7 to version 5 or 6 as part of Windows 10. In other words, they are not interchangeable. If a user logs in with an older user profile on a newer version of Windows, he or she will receive a message stating 'User Profile cannot be loaded' and a default empty user profile will be loaded instead. Of course, the other way around the same will happen as well.

By creating a universal, native Windows user profile this problem is easily solved – on-prem and in the cloud

Our Profile Portability engine knows exactly where all data is stored within the user profile and what it is used for, this includes all folders, configuration files, registry entries, personal application settings, and preferences.

It creates an abstraction layer of all data combined and stores it in a native Windows format while compressing the data at the same time.

While a user is logged in, our Portability engine tracks all changes made by the user compared to default Windows user profile. During log off, by default, all changes are saved in a compressed state to a network location of choice, fully manageable by your IT department. So-called Trigger Points can be configured to control when and how many times user profile data will be saved to the network – or loaded into the profile when needed, post login at application launch, for example.

This approach works for all supported editions of Windows Operating Systems out there including the accompanying user profile versions.

When a user logs back in, onto a machine with a newer version of the Windows user profile, our Profile Portability engine knows exactly which data belongs where. Even when migrating from Windows 7 directly over to Windows 10 this will work flawlessly. It will read the earlier saved, and compressed data and load it into the new (er) user profile.

Additionally, we can use context-aware filters (over 300 of them), combined with (post login) Trigger Points and features like Folder Redirection and Privilege Elevation to further streamline the end user’s experience. All of which are modules directly built into ProfileUnity.

A unique combination – ProfileDisk + Profile Portability

A Profile Disk/container with a user profile originated on a Windows 7 machine will not be compatible with the Windows 10 Operating System. Technically the Profile Disk will mount just fine, however, the user profile on the Profile Disk cannot be used with the newer version of the user profile as part of the Windows 10 Operating System, as explained previously.

When we combine Profile Disks with the flexibility of our Profile Portability technology we create a unique combination. No longer will you have to rebuild your user profiles for each Windows Operating System. You simply create two separate Profile Disks and our portability engine will take care of the rest. This way we prevent a ton of manual, repetitive and error-prone tasks that come with migrating user profile data.

Have a look at this video, just to give you an idea of the options you have.

Data outside of the user profile

Not all application and/or user related data is stored in the user profile. Some applications write to, for example, the Program Files directory. Because ProfileDisk is part of ProfileUnity we have a very extensive toolbox at our disposal to help us overcome just about every challenge you can think of. By creating our own, custom-made portability rules we can tell ProfileUnity to go beyond the user profile and apply the exact same principle, using Profile Portability to other folders as well.

In fact, we can get extremely granular if need be, take a look at this article, it discusses some of the options you have when it comes to isolating file and registry settings on a per-application basis.

By applying something we call 'write by application level' we are able to intelligently review the contents of a user's profile at logoff to determine if changes have been made, and to only write back changes to those applications that have been altered. This reduces logoff times and avoids common 'last write wins' conflicts.

User-authored data

ProfileUnity is able to automatically harvest (copy, sync, migrate) so-called user authored data form all shell folders within the Operating System. While doing so we can limit the Bandwidth used, tell it to do an incremental sync, to only migrate existing files present on the client and a few other things. Even if these folders are already redirected this works just fine. Select your folder of choice, give in a new or alternative location, select the options you feel fit and of you go.

Profile/data corruption, then what?

User profiles do tend to get corrupted from time to time, even when stored on a ProfileDisk. Profile corruption is something I've dealt with on many occasions during my first and second line helpdesk career, years back. The most effective and quickest way to recover? Throw away the entire user profile and let him or her start from scratch. Far from ideal but it got the job done almost every time. Even today this remains one of the go-to solutions saving tons of time, though frustrating your users.

By combining our ProfileDisk technology with our Profile Portability engine we create a unique combination – a user profile fall-back mechanism if you will. Even after completely deleting a ProfileDisk/user profile. The logic behind it is simple, while the ProfileDisk will hold everything relevant to the user, in the form of a user profile, the Profile Portability engine will continue to track all changes made to the user Profile and save it to a location/share of choice, as highlighted earlier.

If we decide to delete an entire ProfileDisk a new one will be created from scratch the next time the user logs in. (Almost) at the same time, Profile Portability will write back all data captured and saved as mentioned above, bringing the user profile back, or near to its original state.

Another option would be to configure one, or multiple backups of the user profile data generated by ProfileUnity. We can set any number between 1 or 99, and because we do not alter the data in any way (we keep it in a Windows native format) it can easily be used to restore parts of or the entire user profile. When configured, a backup of the user profile data will be created at user logoff.

How to handle existing (roaming) profiles

Migrating user profile data has never been easy. Even though ProfileDisk automatically creates a fresh user profile to start out with, it's also empty. Great if you're going greenfield or decide to start from scratch for other reasons, however, this usually isn't the case.

Let's say you are using Roaming Profiles combined with folder redirection – like thousands of companies still do. Since we would like to keep what we already have, here are the steps you need to go through when using ProfileDisk combined with Profile Portability. The best thing is, the whole process will be completely transparent to your users, they simply won't notice a thing.

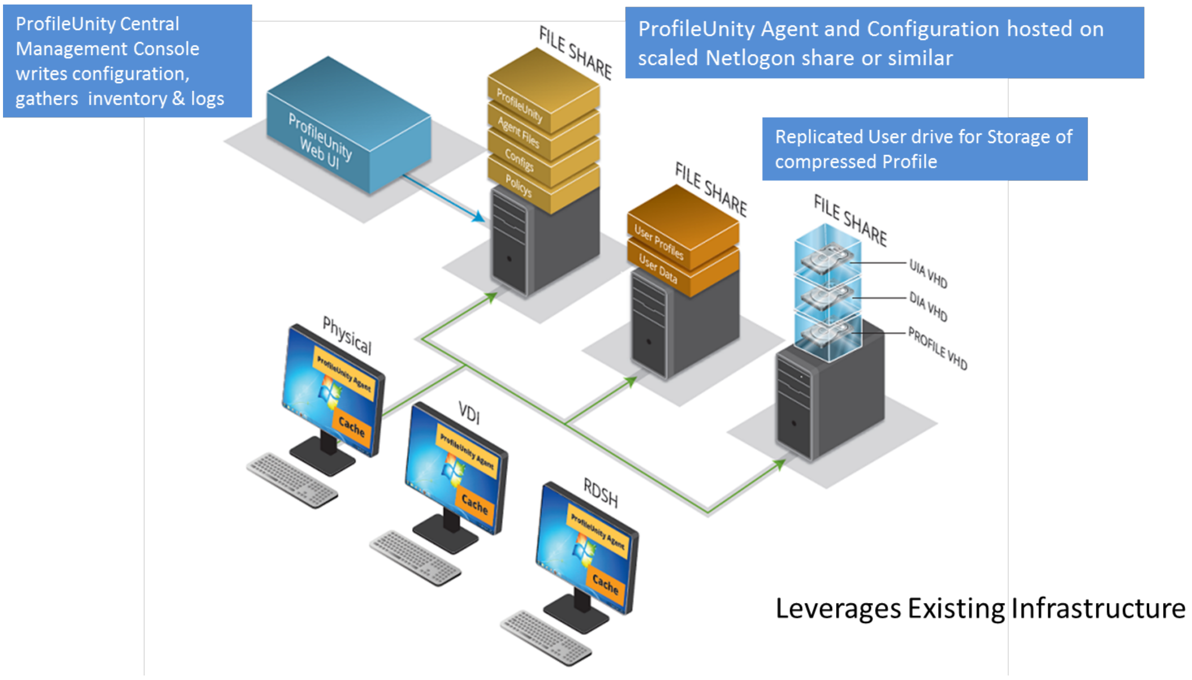
- First, enable and configure Profile Portability – No ProfileDisk just yet.
- Let your users work for a couple of days (using Roaming Profiles) so that all changes are monitored, captured, saved, and compressed to a share/location of choice.
- At some point, you enable ProfileDisk and disable Roaming Profiles. Start out small with a few test users, for example, and take it from there.
- Next, during login, a ProfileDisk will automatically be created including a brand new and empty user profile.
- At (almost) the same time, and depending on your configuration Profile Portability will start to write back/restore your user's personal settings into the users' profile. The Portability engine will do a resynch (incremental) of all data. It will only restore settings not already there, however, since the user profile will be empty to start with, all data collected by Profile Portability will be written back to user profile residing on the ProfileDisk.
- From here you can start fine-tuning as you feel fit. ProfileDisk is designed to handle large user profiles. You might decide to not redirect any folders whatsoever. This would be a judgment call. Often, using folder redirection for certain shell folders is still seen as a best practice.
- Done!

Infrastructural needs and components overview

We couldn't make it any simpler. Besides a single management machine, we do not require any additional infrastructural components. The installation of ProfileUnity contains a built-in webserver as well as a database so no need for additional machines and/or licenses – saving time and money.

Furthermore, ProfileUnity leverages your Domain Controllers' NETLOGON share to store ProfileUnity (and thus ProfileDisk) configuration and agent files, which makes them highly available by default since we all have at least two or more Domain Controllers, right? If, for whatever reason, you would like to use a different file share that's optional as well.

The user profile data and/or ProfileDisk VHDs are stored on a file share of choice, or multiple. Existing, new, you decide. Just apply Microsoft's best practices when it comes to secure file share configurations and you will be good to go – again, straightforward and highly available by nature, have a look at the below overview.



ProfileUnity Central Management Console writes configuration, gathers inventory & logs

ProfileUnity Agent and Configuration hosted on scaled Netlogon share or similar

Replicated User drive for Storage of compressed Profile

Leverages Existing Infrastructure

With ProfileDisk everything is configured directly from our user-friendly management console, no need to manually edit the registry or configure Group Policy Objects. Even better, when PU is installed we provide you with an easy to follow wizard consisting of multiple templates to choose from, depending on what you would like to achieve. All you have to do is fill in the file server/shared folder location where the ProfileDisks and/or Portability data will be stored. Once finished, PU will take care of any GPO's that need to be configured, fully automated.

Integration with ProfileUnity

Because ProfileDisks are part of ProfileUnity you have direct access to all major end-user environment and workspace management features you can think of, offered from a single pane of glass, think: folder redirection, drive and printer mappings, our context-aware filter engine, privilege elevation, application rights management and much more, including our award-winning Application Layering technology, FlexApp.

ProfileUnity Agent

Cached mode. ProfileUnity's desktop client is lightweight and can be configured to operate in either direct or cached mode, which is ideal for even quicker login times. Both leverage a Group Policy template to apply the .exe, no software needs to be installed on individual desktops.

Main configuration steps

To set up ProfileUnity here's what needs to be done:

- Create an Organisational Unit – give it a name. Will hold machines managed by ProfileUnity.
- Create a Security Group – give it a name. Will hold users managed by ProfileUnity.

- When purchased, replace existing evaluation license file.
- Download and install ProfileUnity software.
- Start wizard, choose a template, follow steps.
- Done.

Concluding – why we are different

At this point you may think, surely you are not the only ones doing this?

Well, yes and no. Yes, there are other profile disk/container and/or user/workspace management solutions out there, but no, they do not offer Profile Portability, separate VDH containers (on top of ProfileDisk) as part of the same solution. Built-in data recovery options, the ability to migrate existing profile data, direct integration with a full-blown user profile and workspace management suite, including the industries Nr. 1 application layering solution – FlexApp. VMDK's instead of VHD's and everything fully configurable and manageable through a rich Graphical User Interface.

This paper/blog was meant to provide you with a more detailed overview on some of the challenges we as IT folks face and how to overcome them using our ProfileDisk technology, optionally paired with Profile Portability and of course, all other features and functionalities offered by ProfileUnity. Our roadmap is full of exciting new features and developments and I'm already looking forward to sharing more with you guys on short notice, to be continued…

If you would like to know more, or perhaps are interested in trial licenses, please, do not hesitate to give me, or one of my colleagues a shout.

Contact me directly over at: bas.vankaam@liquidware.com