

## Inside Citrix chapter twenty – The one about application delivery

As we have already seen in Studio, we have a couple of options when it comes to delivering applications to our users. Here it is important to note that I'm not talking about the ability to publish an application; no, I am focusing on the mechanism used to deliver the application after it has been published to a user's desktop. As a side note, I will leave out any potential EUC solutions that might be used to manage the end-user environment, including the ability to assign installed / virtualised applications to your users. This chapter is meant to give you an idea of some of the options you have and to provide you with an overview of the technology and components involved.

We basically have three 'main' flavours to choose from (I know there are more):

- We can leverage locally installed applications. Here I am referring to applications installed as part of the base image (MCS and/or PVS are both optional) of our RDSH and/or VDI deployment (s). Locally installed applications can then be published out to our users. Once launched, these applications will be started from the base image using local compute resources.
- We can apply 'true' application virtualisation using Microsoft App-V, for example. There are multiple software virtualisation products out there, like ThinApp, but App-V is favourite by most and above all supported and advised by Citrix as well. In fact, if we look forward to XenDesktop 7.8 they have made some really nice enhancements to Studio and the FMA with regard to App-V integration.
- Citrix AppDisks is Citrix's application-layering technology, which was introduced with XenDesktop / XenApp 7.8 and will be available with all XenDesktop and/or XenApp editions. Of course other application-layering vendors are optional as well, there are multiple. Since application layering is still a relatively new and unknown concept to most, I will elaborate a bit more on this as we progress.

**FMA fact:** Note how I say 'true' application virtualisation. This is because solutions like XenApp are also often referred to as application virtualisation solutions, so it is really a matter of perspective.

### Locally installed applications

I assume we all know what I mean by this. When we prepare our master image used to provision our RDSH and/or VDI workloads, we manually install certain applications, plug-ins and add-ins etc. And while I say 'master image', this can just as easily be a single virtual or physical machine with a VDA installed, from where our applications would then be published and launched.

## Microsoft App-V

Let's first start with a quick summary of what is involved when we are talking about XenDesktop and/or XenApp combined with Microsoft App-V application virtualisation. App-V consists of the following components:

- Management server – Provides a centralised console to manage App-V the infrastructure and deliver virtual applications to both the App-V Desktop Client as well as a Remote Desktop Services Client. The App-V management server authenticates, requests, and provides the security, metering, monitoring, and data gathering required by the administrator. The server uses Active Directory and supporting tools to manage users and applications.
- Publishing server – Provides App-V clients with applications for specific users, and hosts the virtual application package for streaming. It fetches the packages from the management server.
- Client – Retrieves virtual applications, publishes the applications to the client, and automatically sets up and manages virtual environments at runtime on Windows devices. The App-V client is installed on the VDA and stores user-specific virtual application settings, such as registry and file changes in each user's profile.
- And let's not forget about licensing. When using Citrix XenApp, Microsoft RDS CALs will be needed. As an added bonus, App-V will be covered as part of your RDS licenses as well (note that this does not apply to VDI based environments). This changed when Microsoft switched from Terminal Service (TS) to Remote Desktop Services. With TS CALs App-V still needed to be licensed separately.

By manually adding the Microsoft App-V Management and Publishing servers into Citrix Studio we can publish virtualised App-V applications right onto our users' desktops. These applications will then be 'streamed' over the network using your App-V set-up of choice. Applications can be pre-cached or prepublished and the use of a so-called shared content store is also optional, allowing us to stream directly from a central shared content source without having to stream the App-V package to the local platform.

**FMA fact:** Published App-V applications can be configured to be launched from the Start menu, through Citrix Receiver, using the locally installed (image) App-V client or from the StoreFront web interface.

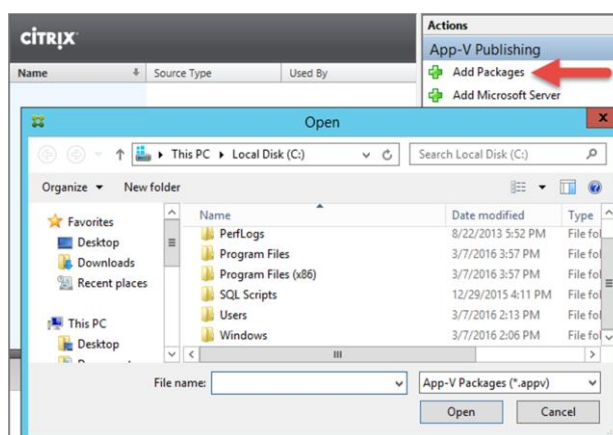
Alternatively the Citrix Connector for System Center Configuration Manager is also optional. It is compatible with XenDesktop / XenApp 7.1, 7.5 and 7.5 and just recently Citrix announced support for both versions 7.7 and 7.8 as well.

The Connector is integrated with the Configuration Manager console to provide a single place where each user's access to applications can be defined and managed. To be a bit more precise, it can be used to:

- Synchronise XenDesktop Catalog and Delivery structures within Configuration Manager
- Deploy software to all types of XenApp and XenDesktop Catalogs
- Leverage MSI and App-V applications already defined in Configuration Manager
- Report application deployment success and failure
- Publish applications to StoreFront and Receiver directly from the Configuration Manager console
- Deploy HDX-delivered applications to PCs already managed by Configuration Manager.

## App-V in XenDesktop 7.8

As of XenDesktop version 7.8 you can use Citrix Studio to manage the distribution of App-V applications to VDI based virtual desktops or XenApp servers without the requirement of a separate App-V server and database infrastructure, greatly simplifying administrative processes and eliminating additional infrastructure costs, a big step forward. When manually adding in App-V packages directly from Studio, all you have to do is fill in the accompanying UNC path and you are good to go, again, no further infrastructural App-V components needed. Ctxappvlauncher.exe will take care of everything.



Adding App-V packages from Studio

## Citrix AppDisks

Throughout the next section I would like to talk about what application layering actually is and also highlight some of the issues we often run into when using the more traditional approach when it comes to delivering applications to our users.

Let me first quote Citrix on their AppDisks technology:

AppDisks is a technology that enables administrators to package and manage their apps independently of their golden images. This in turn reduces the number of golden images required. With AppDisks, instead of installing the application into the golden image, the application is installed (or more specifically the data is redirected) into a virtual disk – a VHD or VMDK depending on your Hypervisor. And once this AppDisk is created, it can be assigned across multiple golden images. Yes, that means

across Windows versions. So long as the application can handle the respective Windows versions, then so will AppDisk.

Of course this is spot on, but, as I mentioned earlier, AppDisks is one of multiple solutions or products that can produce these layers for you. The next few sections will apply to almost every application-layering product out there, with a few exceptions of course.

**FMA fact:** AppDisks will be available with all XenDesktop / XenDesktop editions, Advanced, Enterprise and Platinum. Note that AppDNA will be for Platinum-licensed customers only.

## Application layering in more detail

Application virtualisation (mostly App-V and ThinApp) has been around for some time now; application layering, however, as mentioned, is still relatively new. Although assumed by some and theoretically possible in some cases, application layering is not meant as a replacement for application virtualisation.

In fact, you could say that they go hand-in-hand. Today there are multiple layering solutions available, and while they all take a slightly different approach, the concept is the same, in most cases anyway.

One of the biggest issues when it comes to application virtualisation is that simply not all applications can be virtualised. The more applications a customer has the harder this will be, which makes sense. Note that I'm primarily focusing on RDSH / VDI deployments. Research has shown that being able to virtualise around 50-60% of all applications on average is hard enough, let alone even higher.

From what I have seen personally at various customer sites it's not uncommon for a mid-sized to larger company to have (at least) a few hundred different applications. And if 'only' 60% of those applications can be virtualised (again, on average) you are left with dozens, potentially hundreds even, of applications that you will have to 'present' to your users in some other way. And while cloud-based applications (SaaS) are becoming more and more popular, they won't fit all use cases and primarily apply to new(er) applications in general. Many older, legacy (Windows) applications simply can't be replaced that easily, meaning you will still need to come up with a way to 'present' those apps to your users. This is where application layering can help.

## The big, bad image

So what happens with applications that cannot be virtualised or replaced in another way? Right, they end up in your 'base' image. Not something to get very excited about. Let's have a brief look at some of the cons associated with this approach:

- It will make your applications harder to maintain and manage. When an application needs to be updated / patched you will always have to make those changes on your 'production' image. This could also be true for some of your virtualised applications, which might be pre-cached, for example, but at least you have a choice.

- Although, based on permissions, your users will only be able to see and start what they are allowed to (at least that's what they think): there will (potentially) be a whole bunch of applications on the base image that need to be shielded and/or hidden so users will be unable to locate and start them. Not to mention that, although they are not used by everyone, all applications will still need to be patched and updated.
  - The same applies to adding new and removing old applications: it's not dynamic at all.
  - You will have a hard time installing and maintaining multiple versions or editions of the same software.
  - Your image might potentially grow up to hundreds of GBs. This doesn't have to be an issue per se, but it sure can be.
  - I have come across (multiple) applications that needed to be updated on a weekly basis, or twice per week even. Queuing the updates and applying them once every two weeks or perhaps only once a month is far from acceptable in most cases. Can you imagine the horror?
  - It's not that hard to see how all this will negatively impact production and not to mention your users. Think about it, when updating and thus changing a production image, with most companies it will need to be retested and 'accepted' each time before taken back into production as part of their standard 'change' protocol. This will take one or two days at a minimum, and that's fast: trust me!
  - This could easily force you to create multiple images / silos, based on departments, for example, while it should be 'less is more'.
  - Another consideration might be the use of fat clients (niche software) or another form of persistent desktops, which you might not want to do, preferably.
  - So-called 'Reverse-Seamless' technology could also be used. Here an application will be installed on the end point device and made available as part of the virtual sessions. Again, not ideal but worth considering.
- Application life cycle management becomes much harder.

## Bring on the layers

Although this section isn't meant to give you a detailed technical explanation on how layering works, and this will also differ per product (vendor), here's a simplistic explanation on what it is about... First an application gets installed / captured on a VHD or VMDK file (virtual hard disk). This virtual disk will then be mounted onto a virtual machine (assigned to a user or machine; this also differs per layering solution). In some cases an in-guest mount is also optional, making layers available to physical machines as well.

Almost without exception, all application layer vendors make use of (mini-) filter drivers a.k.a. write filters residing in the base Operating System to 'manage' the file system containing C:\ etc. These 'filters' will also make sure that all application layers mounted to a machine will be seamlessly merged into the file system, making them appear as if they were installed locally, including any .ini, .dll, registry entries and other files that may come with it. From there on, any 'calls' made to the file system, when launching an application, will be 'filtered' and directed to the appropriate application layer, or VHD / VMDK file.

## How they help

Some might argue that using an application-layering solution adds yet another product to the stack: it's another 'technology' and potential interface that admins need to get familiar with. It will also come at a certain cost with regard to purchasing the product, licenses and so on. Applications will not be isolated like with app virtualisation. However, these cons, if you can call them that, are soon forgotten when you have a look at some of the pros that come with application layering. To name a few...

- As for the 'another product to the stack' argument, yes that's true, but if you forget what is going on under the hood technology-wise, which is easily done, then learning how to deal with layers (creating, assigning and updating them) will probably only take you a couple of hours max.
- Another GUI? Perhaps. But if you are a VMware or Citrix shop, for example, then go with App Volumes or AppDisks (which will be integrated into Studio). The impact on your existing product stack will be minimal. And who cares about 'another' GUI if it helps you solve some major issues.
- About the costs. I don't have any list prices to share, (AppDisks will be included for free as part of all editions) but just think about what you would have to do, and how much time and effort it will take managing all your applications, or environments that you might be managing as a consultant, when taking the 'traditional' route.
- No isolation. Yes. But if applications can't be virtualised there's usually a reason for this. I mean, being able to interact with the underlying OS and any applications or other resources that might reside on there can be an advantage just as easily.

What about some of the (other) pros...?

- No more applications in your base image, with a few exceptions free for you to choose.
- You might end up with just one 'golden' image, maybe two.
- Applications can be managed completely independently from each other, on a per layer basis if you will, even on different time schedules or 'on the fly' if needed, without impacting your production environment.
- It also works for drivers, add-ons, plug-ins and so on.
- Depending on the technology used, applications can be added and removed dynamically: the user does not have to log out, reboot etc. When a layer has been added the application icon will appear almost instantly on the user's desktop, same with removing layers.
- They all work on top of your existing infrastructure, leveraging Active Directory for layering assignment to users, groups of users, machines and so forth.
- Non-persistent desktops are now more 'achievable' than ever.
- Support for all Hypervisors (does depend on solution chosen) and works for all major RDSH and VDI platforms out there.
- When combined with a personal data / AppDisk (optional in most cases), you can achieve persistent desktops but still maintain all the advantages of non-persistent.
- Most application-layering products have built-in application life cycle management



capabilities.

- Reduces storage, compute and network costs, and not to mention the time it takes to rapidly ‘package’ and provision applications to users, including overall management.
- Layering and application virtualisation go hand-in-hand. Application Layering is easy; you don’t need to have any specific knowledge (or a lot less anyway), as apposed to App-V packaging, which can be an art on its own. As an added advantage the application layering packaging process can easily be delegated to other teams when needed.

## Layering questions to ask

Although the primary focus is on AppDisks, when thinking about implementing an application-layering solution in general there will always be a few (basic) questions you have to ask yourself (or out loud) with regard to the platform you are supporting or going to support.

- To start, is it Citrix XenApp / XenDesktop, VMware Horizon / View or maybe ‘plain’ Microsoft RDSH that you want to support?
- The same goes for supported Hypervisors. Depending on what you are using, your options might be limited to a few vendors.
- Is there a steep learning curve?
- Does it need to work for persistent as well as non-persistent?
- Do we need to be able to assign layers to users, machines or both?
- Does it support both VHD and VMDKs?
- Can we assign layers to virtual as well as physical machines?
- How about the rest of the vendor product portfolio: do they offer anything else that might benefit us combined with application layering?
- Do they offer a (persistent) personal data / app disk solution? Would I like to use it?
- A user profile solution (based on layering as well) perhaps?

**FMA fact:** Citrix AppDisks is available as of XenDesktop / XenApp version 7.8

## Key takeaways

- Although I narrowed it down to three ways of application delivery, there are of course a lot more flavours to choose from, especially when talking virtualisation, layering and containerisation. Search for the Application Virtualization Smackdown whitepaper, or visit rorymon.com. You will be amazed by the options you have.
- AppDisks is Citrix's approach to application layering.
- AppDisks will be available for all licenses. AppDNA integration with AppDisks will be for Platinum customers only. AppDNA will automatically check your AppDisks for any potential compatibility issues with the underlying Operating System and/or any other software, including applications, security updates and patches etc. already installed and running. When applicable it will tell you what is wrong and how to correct it.
- Application layering is not meant as a direct replacement for application virtualisation: they go hand-in-hand. In practice you will probably use all three, base image-installed applications, virtualised and layered apps.
- Application layering does not isolate applications like App-V does, for example.
- Think of it as just another tool in the toolbox to make life a little easier.
- Remember that, although a single master image is great to have, it is also a utopia in most cases. Just don't go nuts: keep the number of images to manage to a minimum. Less is more.