

## FSLogix first look #2 - FSLogix Apps and Citrix XenApp working together

So, we had our initial look at FSLogix from a Java remediation standpoint, and it seemed to generate quite a bit of interest. However, let's now concentrate on the core product of FSLogix, which is FSLogix Apps, and see the benefits that this brings to those of us with diverse application environments and user bases.



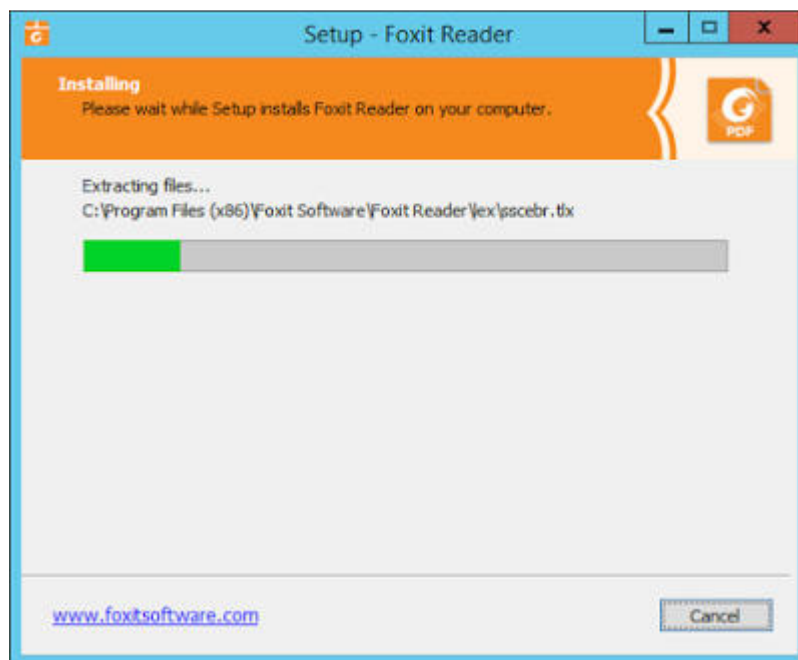
FSLogix Apps is a disruptive technology which almost flips the whole application virtualization ecosystem on its head. Rather than spend time and resource virtualizing applications - whether through traditional app-virt tech like App-V or ThinApp, or newer layering solutions such as AppVolumes and UniDesk - FSLogix allows you to install all applications into the base image and simply "reveal" them to users depending on their entitlements. If you followed my first article on FSLogix, you will know that it isn't just shortcuts and entry points to the applications which are hidden - the entire Registry, filesystem, drivers, services and everything else associated with the application will essentially behave as if it isn't there. No chance of application conflicts, because as far the user running the applications is concerned, the other applications simply aren't there at all. Let's also not forget that applications which are natively installed run faster and more reliably, and that you don't have to spend lots of time virtualizing stuff like device drivers and services because they simply run directly on the operating system. Like everything about FSLogix, the principle behind the FSLogix Apps product is simple, easy and elegant.

We're going to demonstrate this on a Citrix XenApp 7.6 platform to show the value not just for "regular" environments but also virtualized ones. In the XenApp environment, the first benefit you will notice is that image management almost becomes a thing of the past - there's no need for the "image sprawl" that I've seen in so many environments that makes day-to-day support and maintenance incredibly difficult. But even though XenApp and FSLogix seem almost perfectly-suited (particularly in chaotic PVS systems), this isn't just a solution for provisioned Citrix environments - it will deliver the same functionality quite happily to physical machines, VDI instances, standard RDSH, in fact just about anything where there are Windows applications running on a Windows operating system. The possibilities are quite wide-ranging indeed!

For purposes of this demonstration, let's assume we've got a standard Windows Server 2012 R2 image with XenApp 7.6 installed. Rather than publish applications through Citrix directly, the idea is we will present our users with full XenApp published desktops and then use FSLogix to filter the applications available to each set of users. In the interests of keeping this simple, we're going to divide our users into three groups and supply each of them with a specific PDF reader application.

- Desktop 1 - Adobe Reader
- Desktop 2 - SumatraPDF Reader
- Desktop 3 - FoxIT Reader

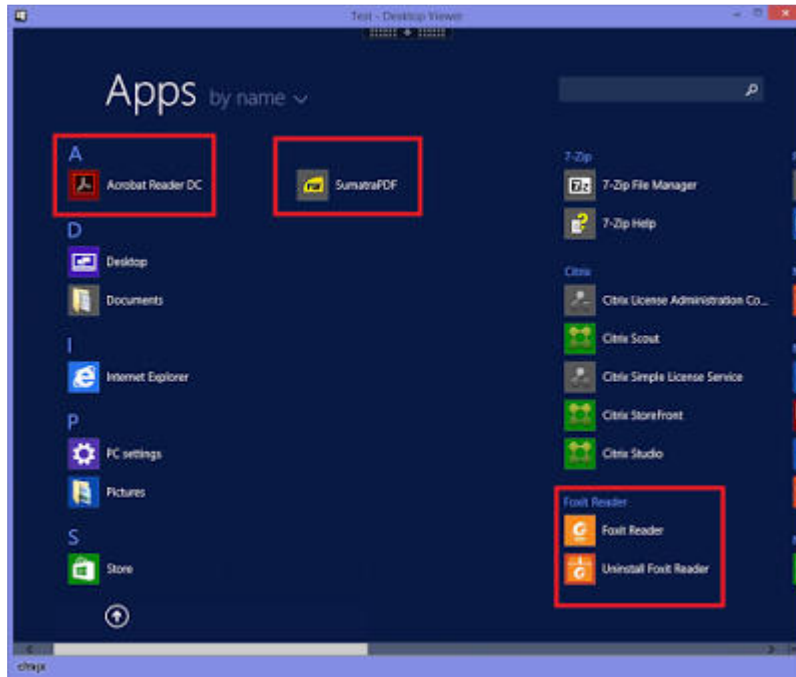
All of the applications will be physically installed onto the XenApp server, but the FSLogix filter driver will reveal the required application sets based around the user's AD group memberships. So the first thing I need to do is get all of these applications installed into the XenApp server base image.



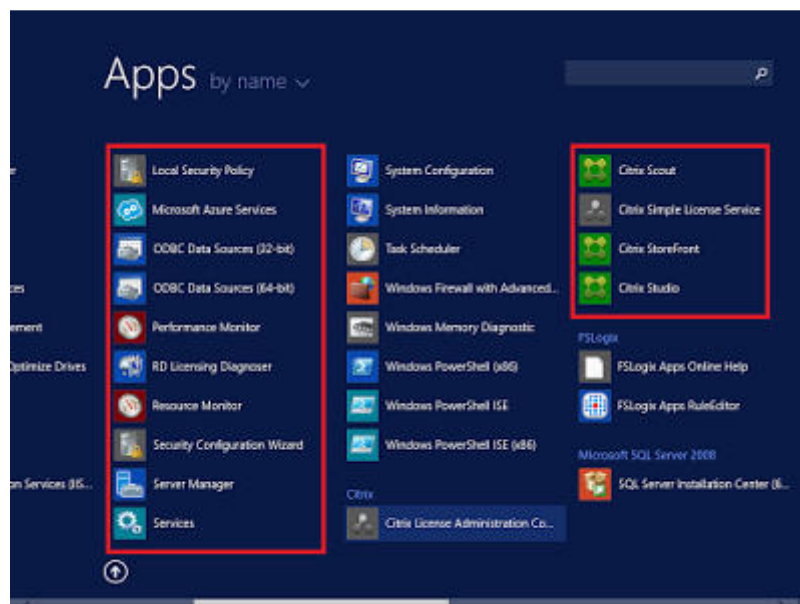
Once we've done this, we need to install *FSLogix Apps* and the *FSLogix Apps Rules Editor*. The beauty again in FSLogix is that it doesn't try to overwhelm you with deployment options of its own - you can leverage existing technologies, for example maybe SCCM and Group Policy Preferences to get the deployment done. I will cover deployment and updates of the FSLogix software and rules in a future article. For the moment, we will just install them the traditional way. They're very lightweight technologies - a single agent that runs as a service, and a console (that could be installed remotely, it doesn't need to be on the XenApp system).



Now we can get to work setting up some rules for our separate desktops. But first, let's publish a Citrix XenApp desktop so we can see precisely what the user would see *without* FSLogix in the mix. We will configure the Delivery Group to host published desktops only and then launch the desktop through the Storefront web site.



All application shortcuts are showing....



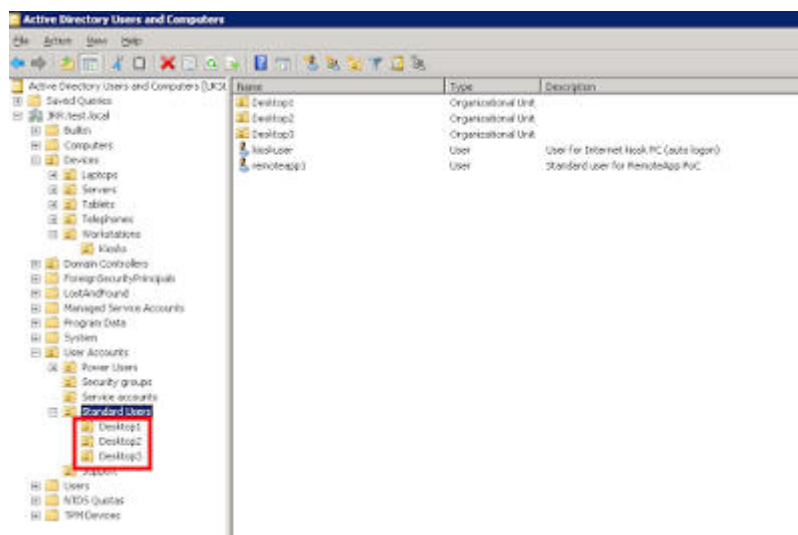
...and we can see admin tools...normally we need to hide these via policies



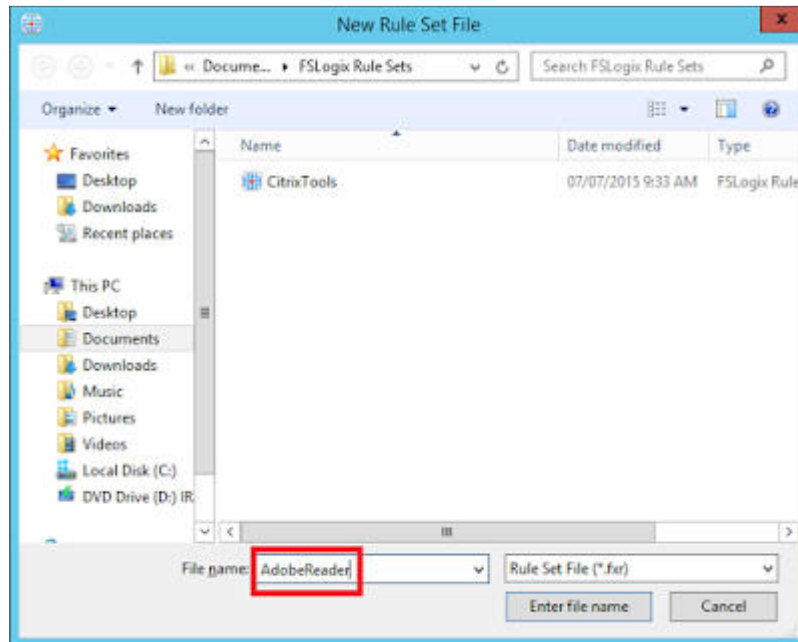
And we've also created custom desktop shortcuts to the apps, which we'll also hide

As we can see, and exactly as we'd expect, the user has access to everything that we've installed into the XenApp base image. Including all the admin tools such as Citrix Studio and the like, and we're going to use FSLogix to hide these too, as well as the desktop shortcuts that we've created ourselves.

So now let's set up some FSLogix Apps rules through the Editor to change this. Firstly, though, we need to create a bunch of Active Directory objects to apply the rules to. I'm going to do it using OUs (directory containers), because a user can only be a member of one OU at a time. However, if you wanted the option of multiple memberships, you could simply use AD security groups instead - both are perfectly viable options (as are usernames, process names, computer groups, etc.)

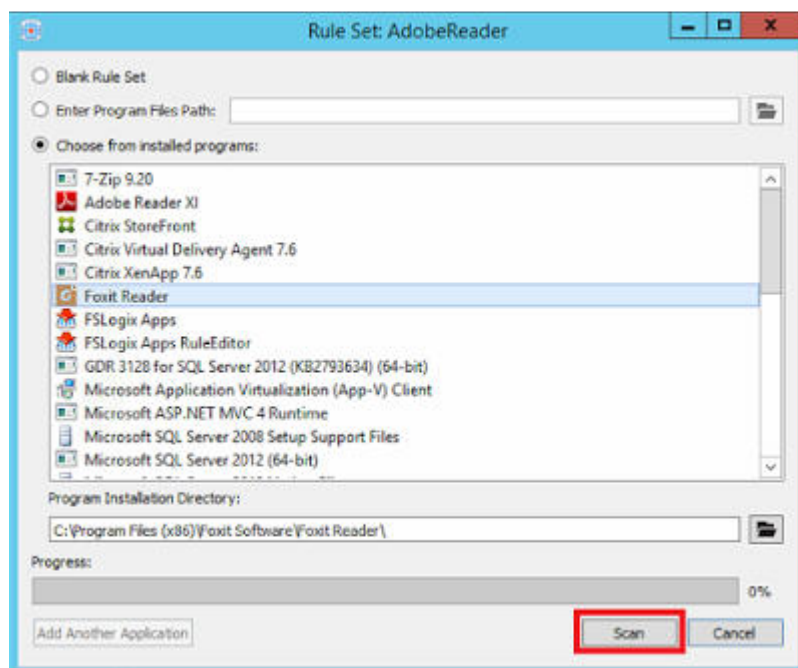


So now we can get on and create the rules. Firstly, we create a new rule called *Adobe Reader*



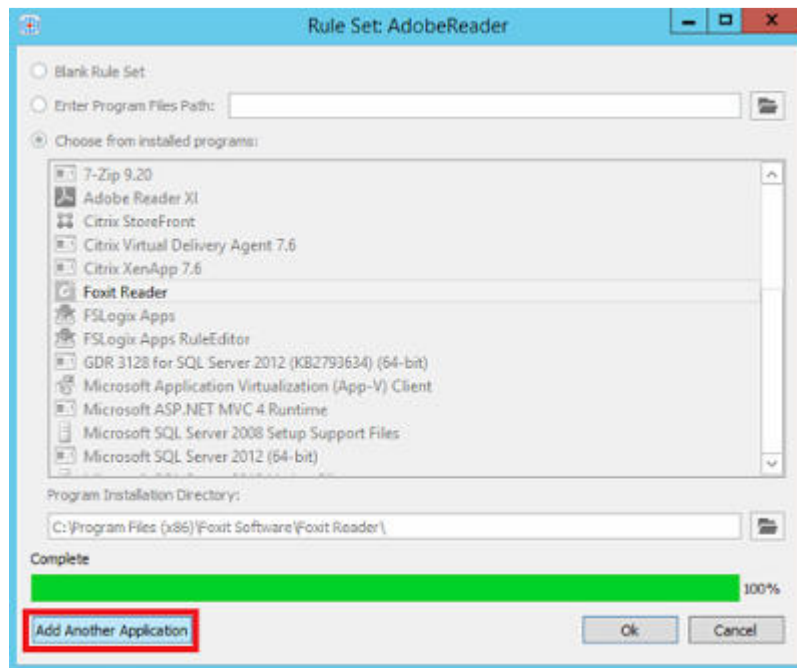
The FSLogix interface seems a bit backward here. The Rule I am creating is intended to show Adobe Reader to the configured users, but to achieve this you have to create *Hiding Rules* for the other two applications. I know there probably isn't a way to present them as "Reveal" rules in the interface, but it just seems more logical to me done like this. Maybe it's just me though :-)

So what we will do is create a Rule to present Adobe Reader, and in this rule will be Hiding Rules for both FoxIT and SumatraPDF.

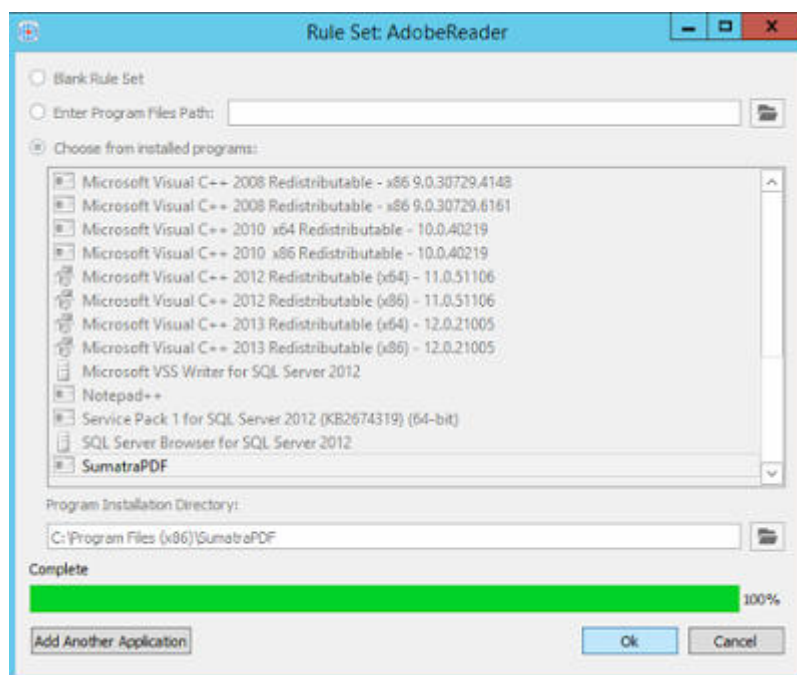


Use the *Scan* button to scan the installed application and automatically detect files, Registry values and shortcuts associated with it.

When that's finished, click the "*Add another application*" button

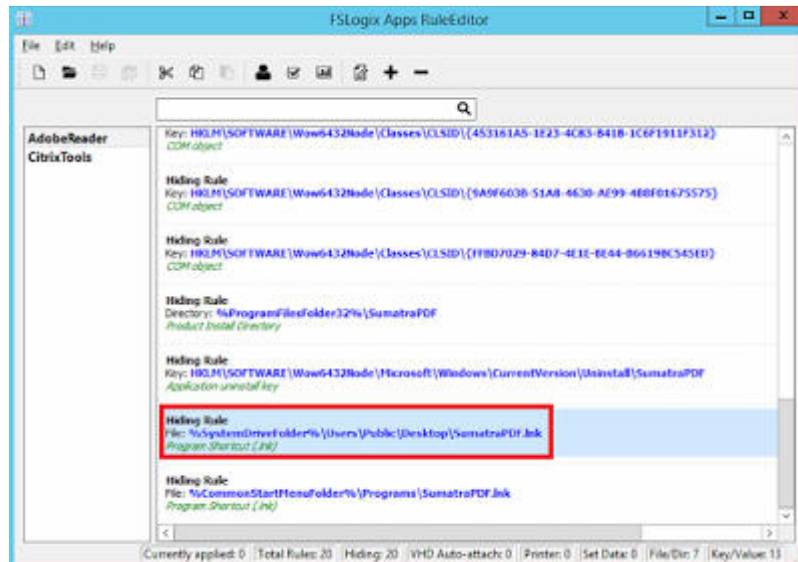


And choose the second application we want to hide, in this case SumatraPDF, and scan that application also.



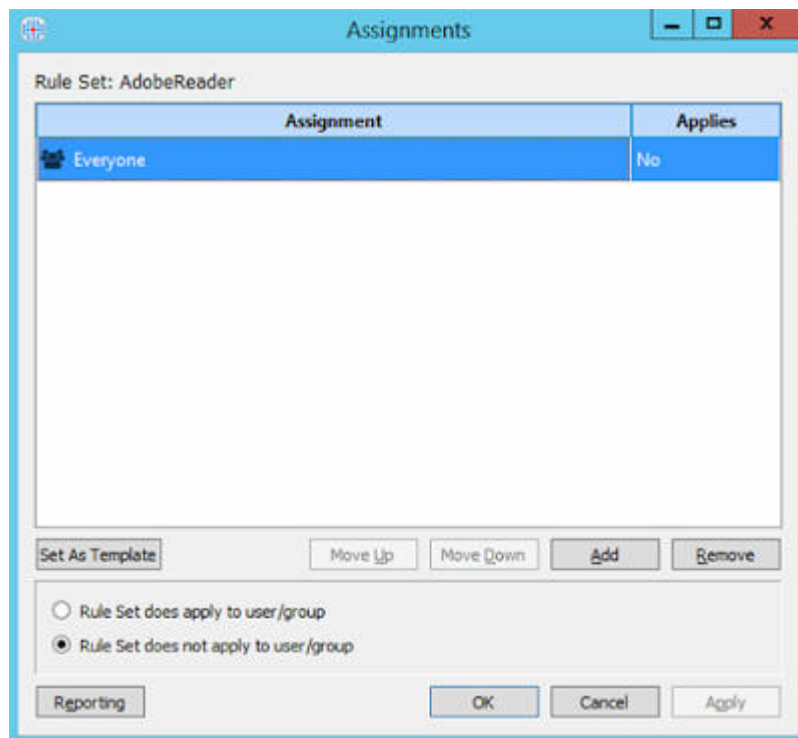
At this stage, click *OK*, and you should be able to review what the scanning tool has discovered about the applications that you want to Hide. As I created shortcuts on the public desktop for these applications, I'm mainly checking that those shortcuts have also been discovered (see below - they have)





Next we need to *Assign* the rule to a our target audience. As we mentioned earlier, we created three OUs and the users in each are going to get a specific PDF reader. So we click on the *Manage Assignments* button (or right-click the Rule and choose *Manage Assignments*)

You will see by default the *Everyone* group has the Rule *not applied*



FSLogix rules are evaluated from the top down, so here's a quick excerpt from the manual to remind us how they operate

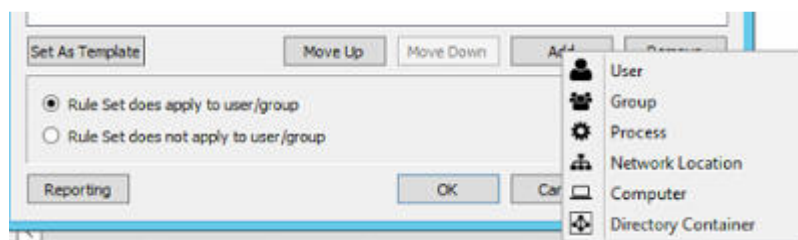
*When adding user and group assignments, pay close attention to the order in which assignments are listed.*

Assignments are processed from top to bottom. For example, if two assignments were listed, the first specifying "Everyone" with "Rule Set does apply" and the second one specifying "User1" with "Rule Set does not apply", then the Rule Set would apply to Everyone except User1. If these two specifications were simply reversed with "User1" with "Rule Set does not apply" being first, then the Rule Set would apply to Everyone.

It is important to note that a user's group membership list is calculated at log on and will remain the same while that user is logged on. This means that changing what is visible to a user by manipulating their group membership requires the user to log off and then back on again in order for the changes to be seen. This is not the case for changes in the Rule Files. Those changes are immediately active upon distribution of the Rule File.

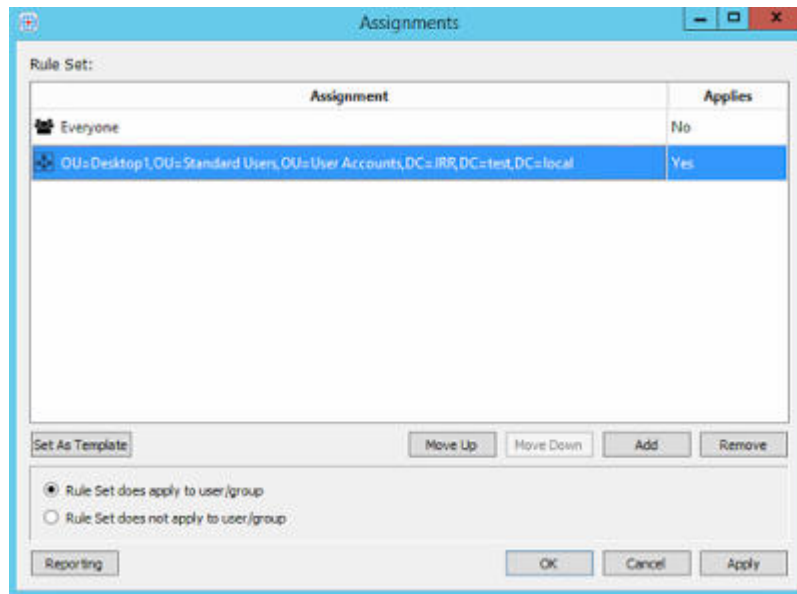
With this in mind, we will leave the Everyone "not applied" Rule at the top and apply a separate rule for our OU underneath, meaning that only a user in the OU specified will have the rule applied to them.

Click on the Add button to add a new assignment, in this case I choose Directory Container, but as you can see there are quite a few different assignment conditions to choose from.

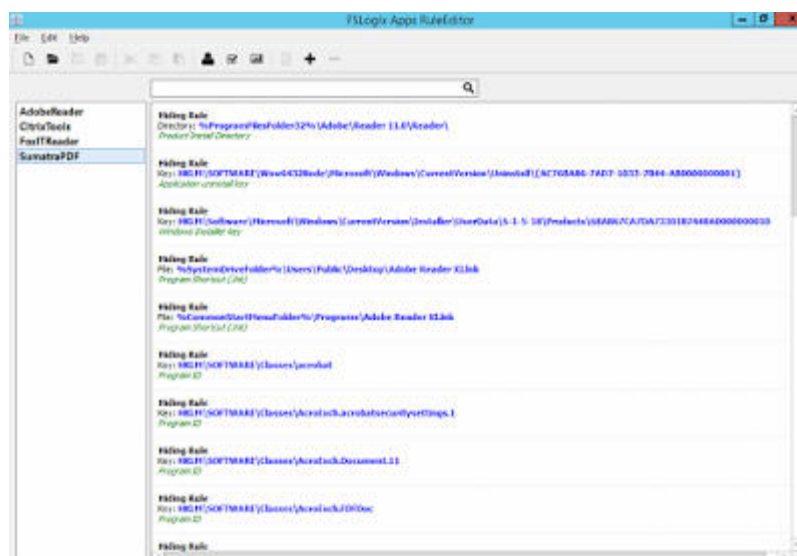


There is a browse button attached to the dialog box for the AD Distinguished Name, but this seems to be a bit buggy, in that it won't let me drill down more than three levels. No matter - we can simply add the required OU in the path for the AD DN, but this could probably do with sorting out. Anyway, we should now have Assignments looking something like this for the Rule

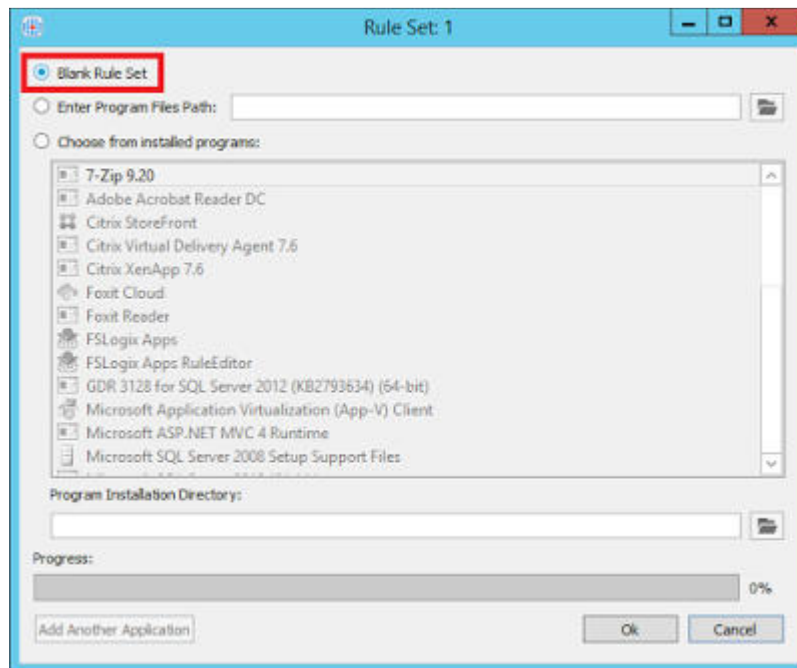




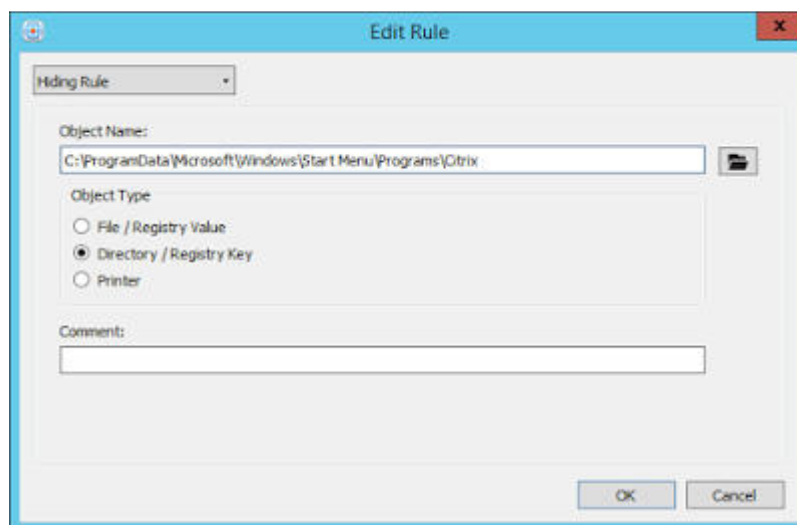
Next we can create matching rules for the groups that will apply FoxIT Reader and SumatraPDF to the OUs for Desktop2 and Desktop3. This is quite straightforward and only takes a few minutes.



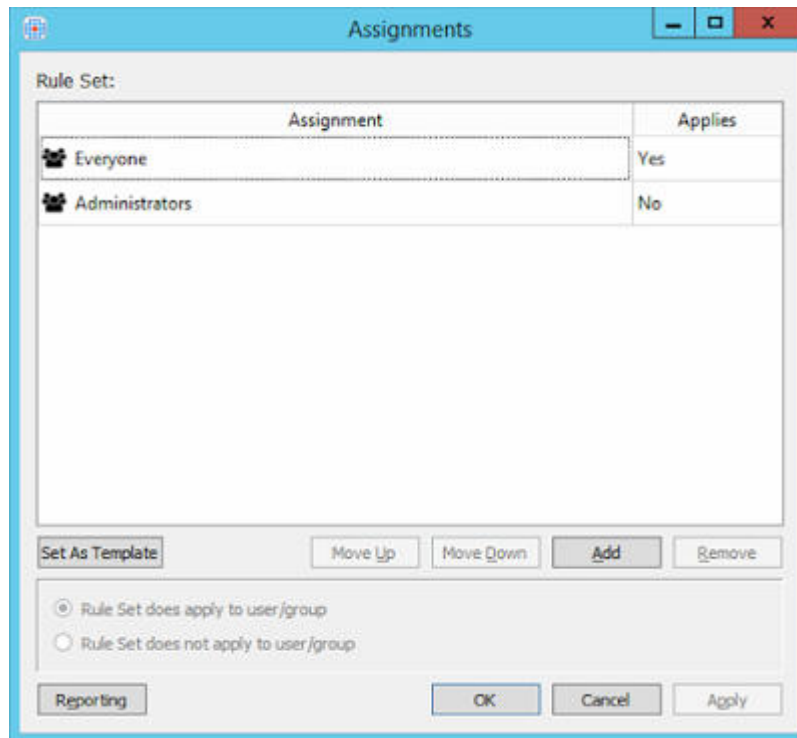
Now that we've got these done, we're also going to create a rule to hide the Citrix admin tools from all users except Administrators. This is a custom rule, so you choose *New Rule* but select *Blank Rule Set* from the dialog



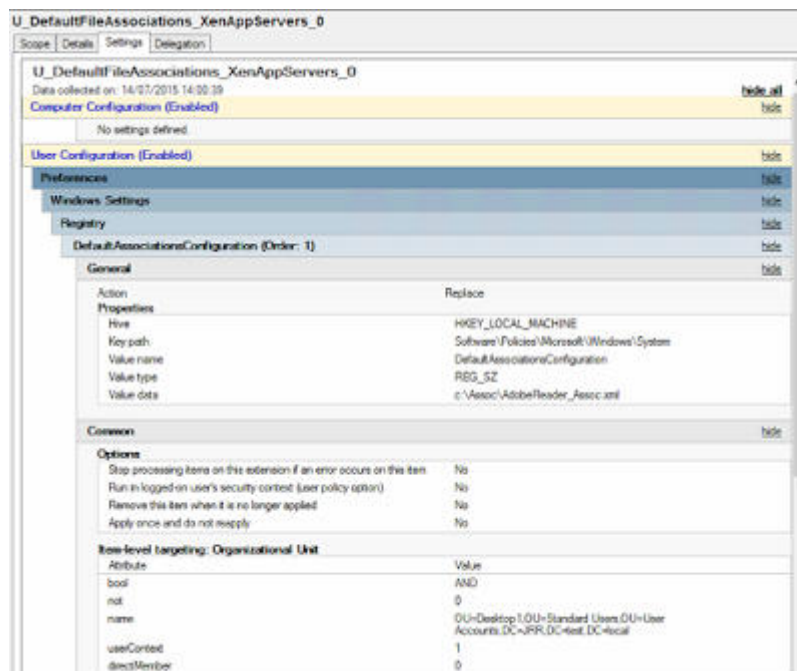
And then you can simply right-click on the right-hand side, choose *New Rule*, and add a *Hiding Rule* that picks up the directory you want to conceal (see below)



The CitrixTools rule we will apply to everyone except Administrators, so that no-one gets to see these shortcuts except admins, so the Assignments we have created as below



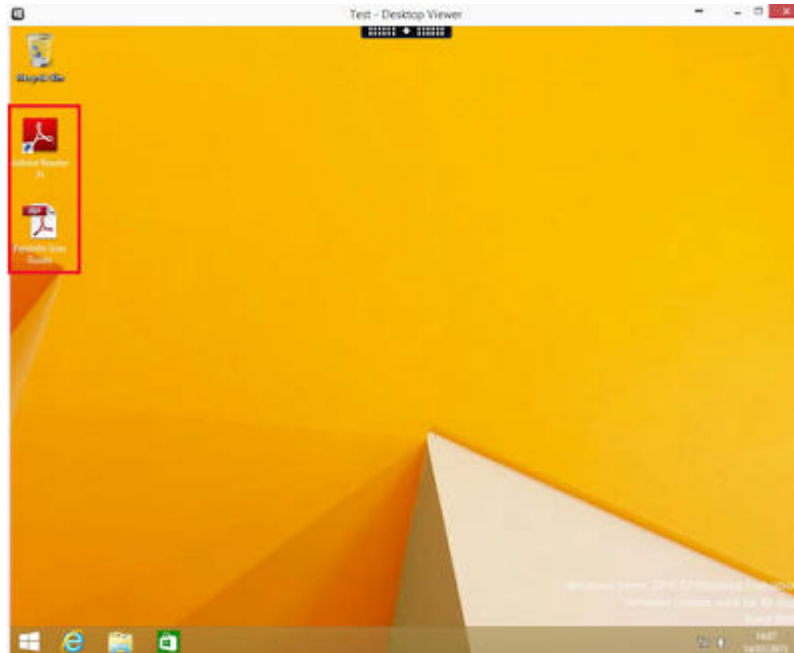
Now we've got all our Rules set up, the final piece we need to put into place are some file type associations (FTAs), so that each user automatically has the application associated with the PDF file type. Regular readers of my blog [can now see where yesterday's post originated](#), so rather than dwelling on this again we will simply use *Group Policy Preferences* and *Item-Level Targeting* to push out a file type association based around the OU of the user logging on, which matches nicely with the way we've deployed the FSLogix Rules. If you need to set this up, please reference yesterday's article for this!



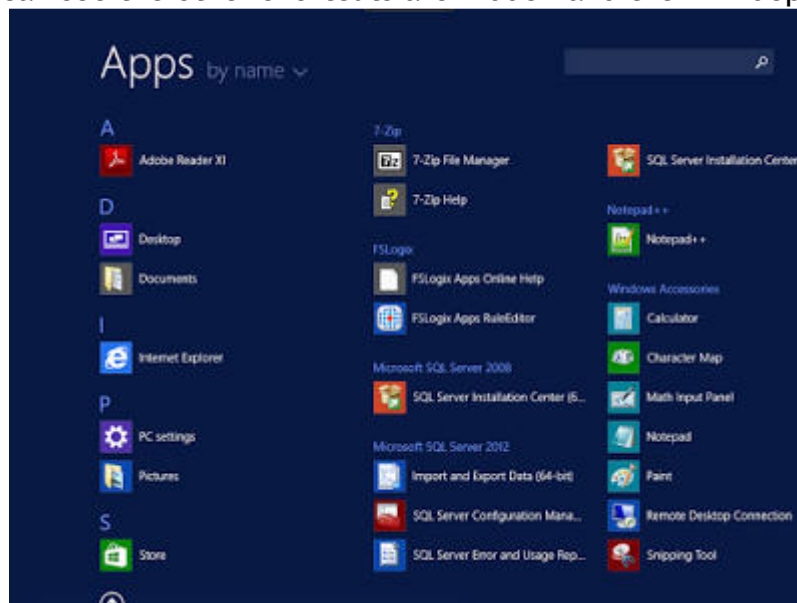
And all that's left to do is copy the Rule files into the required folder. I love this bit about FSLogix - it's just so simple. In this case, I just copy the files from %USERPROFILE%\My Documents\FsLogix Rule Sets to %PROGRAMFILES%\FsLogix Apps\Rules and they are all

instantly deployed and active. You can imagine how simple this is to automate as well, even at scale.

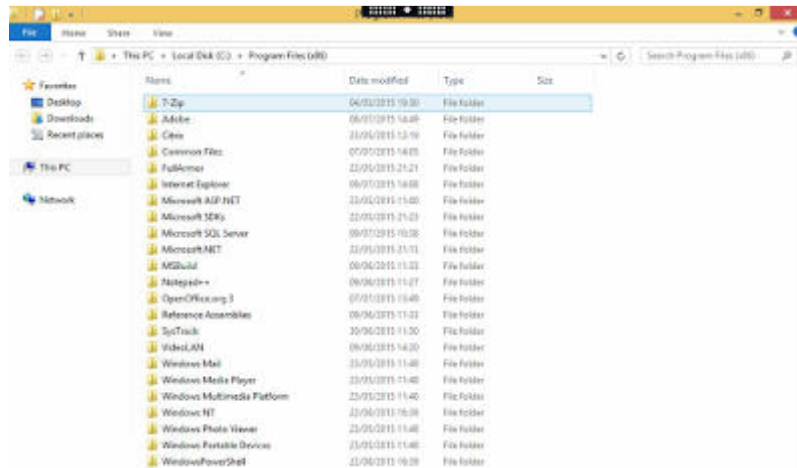
OK, we've got it all set up. Let's log on a user from the *Desktop1* OU (which should just show Adobe Reader and the associated FTA, as well as no Citrix admin tools). We've bunged a PDF file onto the public desktop so we can easily see whether things are working as intended.



You can see the other shortcuts are hidden and the FTA deployed



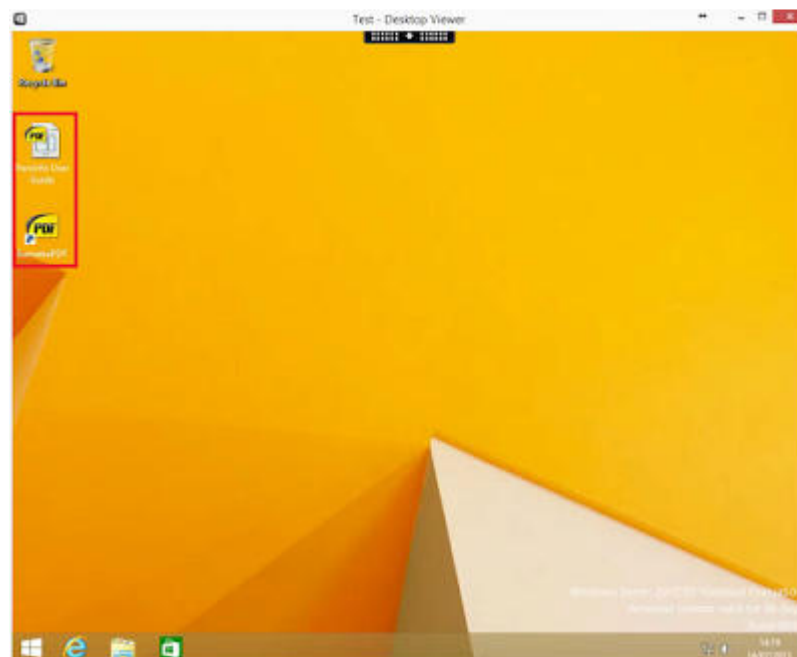
And compared to the earlier screenshot, the other PDF readers and admin tools are gone

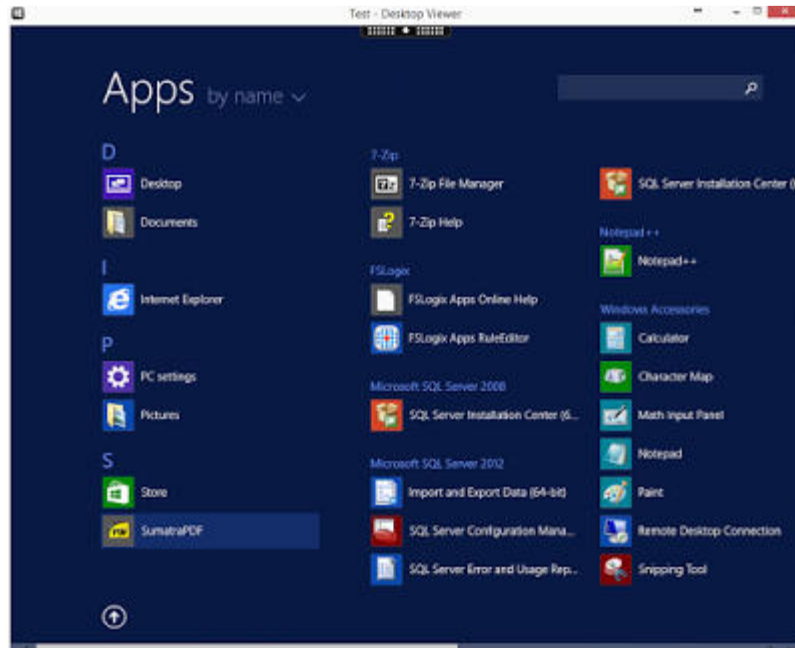


And in the filesystem, the other entries are simply gone...

You get the picture - I won't bother delving into the Registry as well, those entries will also all be gone. Software that just works breeds confidence :-)

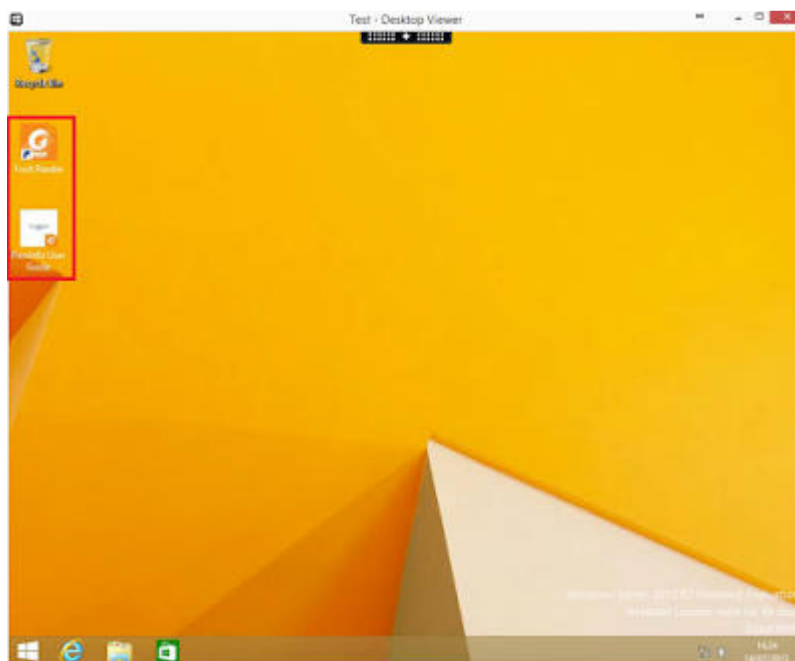
So now let's log on (while the current user is logged in, to give it the normal XenApp multi-session experience) as a user in the OU for *Desktop2*, which should allow them to see SumatraPDF as the PDF reader software.



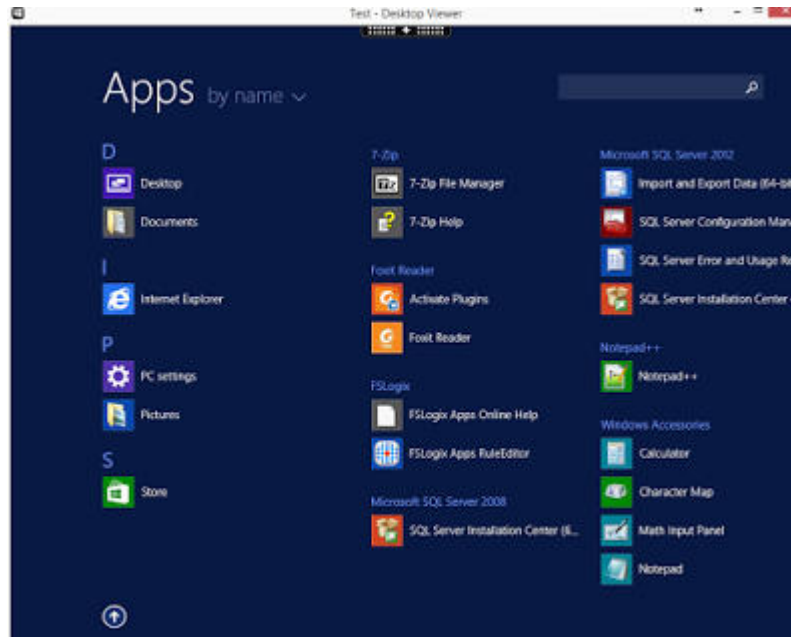


Wow! That seemed to work beautifully - our second user is happily assigned SumatraPDF for his use, and he isn't even aware that Adobe Reader or FoxIT are installed on the system. In fact, effectively, as far as his session is concerned, they don't exist on there - even though in another session on the same XenApp server, another user is running Adobe Reader.

So let's log on with a user from the OU for *Desktop3*, who should be assigned FoxIT Reader as his PDF viewing application.







Clean as you like - the third user on the server can only see FoxIT Reader, and only use FoxIT Reader.

I have to say I am mighty impressed by this, it's dead easy to deploy and manage, and we can direct users to sets of applications based around the context they're operating in without much fuss. As all the applications are actually natively installed, they will run cleaner and more efficiently than if they were packaged, sequenced, siloed, cloud-hosted or any one of the many ways you can deliver apps these days. And those environments with the images upon images that are frankly a blight on a lot of XenApp deployments these days - you can kiss all that goodbye.

You may be thinking about base image bloat, but FSLogix can also mount applications as VHDs to avoid this, effectively redirecting the applications onto the network or a secondary drive to become what FSLogix call "app containers". All in all, it's an elegant solution and one that I feel has a lot of mileage. However, as I said in my previous post, the interface could do with a lot of tidying up and making a bit more logical, but those are cosmetic improvements and I'm sure they will be on the roadmap.

So that covers the base core functionality of FSLogix - filtering application sets based around configurable parameters. We've [already looked at it doing Java remediation](#), but in future I intend to take a closer look at app containers, profile containers and vulnerability mitigation - but not before I get back to some AppSense stuff over the rest of this week. Stay tuned for more EUC fun!

Posted by [James Rankin](#)