

A New Approach to Per User Application Management

Spanning conventional and
virtual infrastructure

INTRODUCTION

Our premise is simple: existing methodologies for delivering and deploying Windows applications are based upon outmoded ideas and outdated technology. We believe there remains a need for a product that makes it simple for each user to have their Windows® applications individually tailored for their device. When a user logs on they should see only the applications that they are licensed to use regardless of whether they are using cloud desktops, virtual desktops or traditional desktops.

“There is a strong move towards regaining control of corporate activities through re-centralizing desktops using server-based approaches. However, the problems soon come to the fore: uncontrolled images taking up resources and licenses, users needing new applications on their desktop and having to wait for IT to provision them. A single image approach with dynamic provisioning of apps on a ‘need to access’ basis is the Nirvana for IT, the business, and the individual”

— Clive Longbottom, Service Director, Quocirca

Traditional application virtualization solutions that operate by packaging and streaming down to the device have severe limitations:

- **Poor Application Coverage** Due to limitations in packaging methodologies many applications, especially those that use device drivers, cannot be packaged.
- **Poor scalability.** When an application is delivered at runtime “Streaming” servers can be overloaded.
- **Lack of IT agility.** IT staff has no idea if the next version of your business critical application will be able to be packaged until they actually attempt the packaging. Due to poor application coverage it is impossible to effectively plan ahead.
- **Network performance bottlenecks.** Introducing additional streaming servers can bottleneck at the network.
- **Prohibitive labor costs.** The most expensive variable in the equation is labor.

The preferred application installation method for any administrator is to install applications the way the vendor intended them to be installed (e.g., via its .msi file). We founded FSLogix to solve these needs.

In part 1 we discuss the various models available and widely used for per user application management, and the new approach from FSLogix. Part 2 provides an overview of how FSLogix works, including the functional components and the workflow of configuring a system and various applications.

PART 1: PER USER APPLICATION MANAGEMENT

System administrators need to control application availability based on the user who is using the system. Traditionally they have used a variety of tools and approaches based on their specific set of requirements. All of these methods have their drawbacks and the administrator is usually left trying to decide which drawbacks are the least inconvenient.

FSLogix introduces a new way of managing a user's access to applications that removes a lot of these disadvantages.

EXISTING METHODS

Application Virtualization

Application virtualization is the concept of taking the components of a Windows application, bundling them up (typically into a single file), and distributing the bundle to a computer where it can be executed. On the computer an agent makes the application able to be executed by "virtualizing" some of the system resources such as the file system and registry. The agent "tricks" the application into thinking that it is normally installed and that it is running in a normal Windows environment.

"Started by IT veterans Randy Cook and Kevin Goodman, FSLogix Apps is a cool and easy to use Windows application management solution, solving application access control and visibility via easy to use policies. Managing applications on laptops, desktops, SBC and VDI is dramatically simplified with FSLogix."

— Ruben Spruijt, CTO - Microsoft MVP, CTP and vExpert, PQR

Application virtualization can typically make an application visible to a particular user either by controlling access to the bundle or as a capability built into the agent.

Downsides:

1. Repackaging the application. This can be quite expensive as intimate knowledge of the application is required and testing of the bundle must be done to make sure the bundle is complete and functions correctly.
2. Compatibility. Problems can arise because of limitations of the virtualization agent or when an application is expecting to interoperate with other applications on the machine. The typical application virtualization success rate depends on the applications being virtualized and the technology being used but usually it ranges from 50% to 90%..
3. Performance impact. The agent software usually adds overhead because it uses additional CPU cycles, creates additional IOPs, and because it consumes memory.
4. Typically does not handle non-applications well (e.g. Office add-ins, browser plugins, etc.)

Application Streaming

Application Streaming distributes an application “Just in time” and can speed up the initial deployment and launch time of an application. This capability makes it palatable to deploy an application based on a user’s request. Many times this is combined with an Application Virtualization solution.

DOWNSIDES:

1. Does not work well for multi-user systems since the application, once installed, becomes visible to all users on the machine
2. To make the application unavailable, it must be uninstalled.
3. Disconnected computers will not be able to make the application available.
4. Typically does not handle non-applications well (e.g. Office add-ins, browser plugins, etc.)

Controlling access via ACLs

Permissions can be changed on application files so that only the desired users have the rights to be able to launch the program.

DOWNSIDES:

1. Can easily be circumvented by users who have the ability to change file permissions.
2. Depending on the number of ACLs to change, it can be a slow operation.
3. Typically does not handle non-applications well (e.g. Office add-ins, browser plugins, etc.)

Creating/deleting shortcuts

Some products delete the shortcuts to the application for certain users.

DOWNSIDES:

1. Savvy users can browse to the file system location and launch the application directly.
2. Does not control other methods of launching applications (file associations, COM, etc.)
3. Does not handle non-applications (e.g. Office add-ins, browser plugins, etc.)

Disable execution

It is possible to hook into the Windows process creation system and then, based on the user, the application, and a policy, cancel the start of the process.

DOWNSIDES:

1. The user sees that the application is installed and only discovers that she doesn’t have access when she attempts to launch the application.
2. Typically does not handle non-applications well (e.g. Office add-ins, browser plugins, etc.)

Application Distribution products

Application distribution products like Microsoft SCCM, Altiris NS, or LANDesk Total User Management typically have methods to provide applications to particular users. These include doing a normal install of an application at user logon and providing a user portal where applications may be requested.

DOWNSIDERS:

1. Will not work in a disconnected environment.
2. Will not work for multi-user systems since, once the application is installed, it is visible to all users.
3. Depending on the size of the application, may be time consuming.
4. In order to revoke access, the application must be uninstalled.

Remote execution

Some products have the ability to execute an application on one machine but remote the GUI, mouse, and keyboard to the user's machine. Providing access to the application is as simple as showing the icon to the user or not.

DOWNSIDERS:

1. Does not work on a disconnected computer.
2. Some applications don't work well due to bandwidth issues, application interaction requirements, etc.
3. The OS that typically hosts these applications is usually a Windows Server OS, not a desktop OS.
4. Additional licensing costs may apply.

The benefits of a solution like this, besides only maintaining a single image, come in the form of flexibility. Since this works on any form factor, and each user is different, it's an ideal solution for Terminal Servers that previously had to be placed in application silos."

— Gabe Knuth, Industry analyst and blogger

THE FSLOGIX WAY

FSLogix introduces a new way to control application availability by user. The administrator defines application policies and which users should have access to those applications. Once the policies are in place, applications that are on the system will only be available to the appropriate users. To the users that are not supposed to have access, all traces of the application disappear. Application files, application registry settings, uninstall information, file associations, run entries, etc., all simply disappear for the user. They are not simply "hidden". It's as if the application has been uninstalled. The per-user visibility always applies on any kind of system, for example: concurrent multi-user systems, serial multi-user systems, and single user systems.

ADVANTAGES:

- No application re-packaging. The system administrator simply installs the application normally. That may be via MSI, application virtualization, etc.
- Performance impact is low. The FSLogix agent is doing the opposite of application virtualization. Instead of making things appear to be installed, it is making things appear NOT to be installed. This is a much simpler operation and therefore the impact to the system, in CPU utilization, storage utilization (IOPs), and memory utilization, is much lower than other systems. Applications that are allowed to be seen run on the normal file system and registry without any work being done by the FSLogix agent.
- Compatibility. Since the FSLogix agent is not actively altering the behavior of the system, other than making things disappear, application compatibility is not a problem. Applications execute completely normally.
- Dynamic. Because there is no installation, uninstallation, changing of files, etc., the availability of an application to a user is instant. Schedule based, network based, and other types of events can quickly and easily change application availability for the user. Users that are using the same computer will all have the proper visibility, even when they are using the system at the same time.
- Finer grained control (plugins, etc.). FSLogix policies can represent any set of files and/or registry entries. It is very well suited to handle some types of products that other methods simply can't deal with. For example, Microsoft Office add-in products, browser plugins, etc., are all very simple to control.
- Disconnected. Because all of the application files are on the system and the policies are cached on the machine, the system continues to work perfectly in a disconnected environment.

PART 2: HOW FSLOGIX APPS HIDES APPLICATIONS

CONCEPTS

Rule

A rule is a description of logic that should be applied to an operating system object. There are various types of rules, but the rules we are concerned about while hiding an application are "hiding" rules. Such a rule describes an object such as a file or registry key that should be hidden depending on the context of how it is accessed.

Rules are organized into rule sets, each of which is stored in a file (typically with the .fxr extension). These rule sets are commonly used to describe the various objects that need to be hidden in order to completely hide an application from end users, making it seem as though it were never installed.

Rule Assignment

A set of rule assignments determine how a rule set is applied in various circumstances. For example, a rule assignment can say that the rule set applies to users in the "Accounting" group, but not to users in the "Administrators" group. Application of rules can be controlled based on specific user, local or directory group membership, IP address, computer name, etc. Rule assignments are stored in a separate file (typically with the .fxa extension).

COMPONENTS

Driver

The core of the FSLogix Apps software agent is the driver (frxdrv.sys). This component is a file system mini-filter driver. It is primarily responsible for intercepting requests from other software on the system to access objects such as files or registry keys, and changing them.

When an application is being hidden, the changes might involve simply making the object appear to not exist, e.g., an attempt to open a file will fail with a “file not found” error code. Or in a more complex case, it may involve changing the data results of a system call, e.g., an attempt to enumerate all the files in a directory may have one of those files removed from the resulting data.

Service

The second major component in the Apps software agent is the service (frxsvc.exe). This component is a Windows service running as system. It is responsible for communicating various data about the system state to the driver, e.g., informing the driver when new users login to the machine. It is also responsible for taking rule files and rule assignment files and “compiling” them to a format that is consumable by the driver. One additional responsibility of the service is that it provides an API for other programs to do things like add rules dynamically, query the rules in place on the system, etc.

User Interfaces

The other components of the software agent are various user interfaces and supporting files. There is a GUI rule editor that assists administrators in creating rule sets, hiding applications installed on a system, making rule assignments to control how rules are applied, testing rules, etc. There is also a command-line interface that can be used for scripting many of the functions that the software can perform. A Windows Event Viewer integration module is also provided to assist administrators in the management and audit of the software.

WORKFLOW

Create Rule Sets

To explore the workflow of the agent, we will choose a real world problem scenario and trace how the software components of the agent work together to solve the problem. For our problem scenario we will imagine a shared system (Remote Desktop Services/Citrix Server) where we would like to have users from the accounting team able to use Google Chrome (because they have some fictional Chrome plug in that assists them in their job), while other users are not able to use Chrome because they have not been trained and only require the built-in Internet Explorer browser functionality.

We will assume that the target system already has the Apps agent installed, and additionally has the rule editor GUI installed. In a real-world situation, the rule editor would likely not be installed on the production machine.

The first step in setting up this scenario is the installation of the Google Chrome software. This is an area where the FSLogix Apps agent does not get in the way at all. Installation is accomplished normally through traditional means (sneaker net, SMS, application virtualization software, etc.)

Second, we would launch the rule editor to create a rule set for hiding the Chrome software. In the rule editor, we choose to create a new rule set, select the Chrome software from the list to auto detected applications, and press the scan button. The rule editor discovers all the system objects that are a part of the Chrome software and that will need to be hidden, and it creates rules for each of these objects.

Third, we use the rule assignments dialog in the rule editor to determine to whom the rules will apply. By default there is an assignment that the rule set does not apply to "Everyone." We will change the "Applies" setting for this rule to "yes", because we want the Chrome software to be hidden for (almost) everyone. We add a second rule assignment for the accounting group, and set applies to "no", meaning that the software will not be hidden from these users.

At this point we now have a rule set and assignments that will solve our problem. In the real world, this would likely be accomplished on a non-production environment, and the following steps would be used to apply the rules to the production machines.

Note that in addition to creating rule sets in this fashion, there are a number of rule set files available for download on the FSLogix website. Rule sets for many of the popular applications, including Google Chrome for this example scenario, can be found there.

Deploy Rule Sets

We will next copy the rule and rule assignment files from the "My Documents" area where they are saved to the agents working directory at %ProgramFiles%\FSLogix\Apps\Rules. At this point, the FSLogix Apps service detects that new rules and assignments have been added to the system. It compiles these rules for use in the driver, and during this process it fixes up differences that may occur between two machines. For example, if you created the rules on a system where your system drive was C:, but the system where you are installing the rules has a system drive of D:, this all gets fixed up so the rules will work. Additionally, things like the "Accounting" group will be resolved to a SID, in order to make the driver's workload easier (so it will be as fast as possible). The compiled rule files are placed in the %ProgramFiles%\FSLogix\Apps\CompiledRules directory, and the service notifies the driver that these rules need to be loaded.

The driver will load the compiled rule files into memory and immediately apply the rules to the system. If you are doing all this as a user who is not in the Accounting group, you will see the shortcut link for Google Chrome immediately disappear from your desktop.

Conclusion

CIOs and IT managers are faced with ever increasing complexity in their desktop infrastructure, often requiring multiple sets of specialized tools to complete simple management tasks. New tools for application management should remove complexity and cost, cover a wider spectrum of form-factors and environments, and create greater flexibility and portability. Simplifying and standardizing environments helps administrators take better advantage of new models and industry innovations like DaaS and VDI, without creating even greater complexity or isolated 'islands' of technology.

FSLogix Apps dramatically reduces the number of required images in an enterprise, helps to eliminate silos, and provides a solution to standardize application management for all users, creating greater flexibility in infrastructure design and evolution. This approach reduces complexity, minimizes costs, and allows administrators to focus resources on more critical projects that foster business agility and more strategic IT value.

ABOUT FSLOGIX

FSLogix Apps is an advanced application filtering solution that enables administrators to dramatically reduce the number of images required to support any enterprise, with any deployment system, on any Windows based infrastructure. FSLogix Apps delivers Dynamic Application Visibility, and policy based control of application visibility to individual users or groups from a single gold image.

To learn more about FSLogix, visit us online at:

www.fslogix.com



FSLogix
573 West 1830 North
Suite 3
Orem UT 84057