APPLICATION VIRTUALIZATION

With a specific focus on virtual desktop infrastructure (VDI) initiatives, smart organizations must see applications as the starting point for the overall design. Think about what makes an organization productive. It is reliable access to a specific set of corporate-sanctioned applications. In the end, it's all about the application. Of course, you will need to design and build a robust virtual infrastructure with all the bells and whistles to support your virtual desktop initiative, but if you do not consider the applications in the design, the entire initiative may fail. Application virtualization should be the cornerstone of a well-designed virtual infrastructure. The focus of this four-part series is to highlight application virtualization and the overarching benefits of this technology.

What Is Application Virtualization?

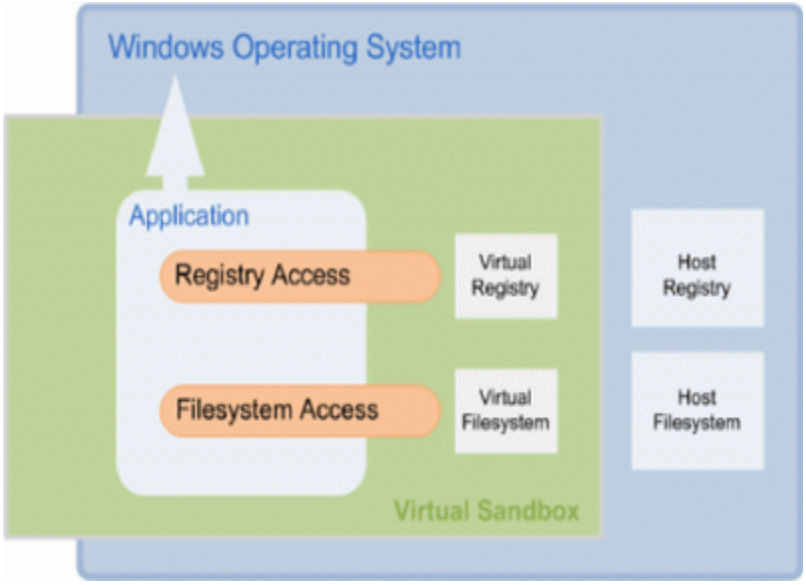
Application virtualization is all about separating the application from the guest operating system (OS). Technically referred to as abstraction or decoupling, application virtualization involves encapsulating the application within a virtual container, including private registry and file system locations for application access.

Traditional application deployments involve installing the application onto the OS, including multiple updates to the system registry and local file system. The application becomes bound to the specific guest OS. Why is this a problem?

- ⊡ Installed applications are inflexible and cannot be easily ported between different hardware platforms or guest OSs.
- ⊡ Installed applications also become susceptible to OS changes through normal updates and patching.
- ⊡ Installed applications do not allow for application contention (two applications requiring different versions of the same system file or supporting applications such as Microsoft's .NET Framework).

Abstraction, Encapsulation, and Isolation! Oh My!

Application virtualization addresses the shortcomings of traditionally installed applications by using a method of abstraction, encapsulation, and isolation as Figure 1 shows.



Each application is abstracted from the guest OS utilizing a virtual registry and file system. The application is also isolated within a virtual sandbox environment from other applications, thus solving application conflicts and allowing for multiple versions of the same application to run on a single guest OS. The virtualized applications are encapsulated in standalone executables with embedded runtimes or CAB files making applications portable and easy to deploy on multiple hardware platforms and OSs. The guest OS also benefits from the deployment of virtualized applications through increased stability and simplified image management.

The management of traditional applications involves a labor-intensive life cycle of testing, deploying, installing, patching, updating, and uninstalling. Add to this the issue of maintaining this life cycle across multiple hardware platforms and OSs. For organizations with large application pools, this translates into increased man hours supporting applications and fewer available hours to respond to business demands and provide needed innovations.

With application virtualization, the management model is greatly simplified. Applications are initially profiled or captured using a clean OS environment. All required application patches are included within the profile and prerequisite applications such as .NET Framework or Java Runtime can be included or linked to the profiled application. The profiled applications are saved as encapsulated files and stored on a secure network file share. From this point, the applications are streamed directly to physical workstations, virtual desktops, or server environments all from

a single encapsulated file. The real value of this process lies within the ongoing maintenance of the applications.

Each application has a single source repository to manage for updates and patching. Thus, IT only needs to touch an application one time when applying updates or patches, greatly reducing the man hours required. Updated applications are immediately available at the next launch by end users without requiring a deployment or installation at the endpoint.

Most organizations do not grasp the fundamental difference in thinking that is required when considering a move to application virtualization. Application virtualization changes the way IT organizations approach application management including life cycle management as described earlier and the deployment of applications.

Traditional Application Deployment

Traditional application deployments involve a complicated process of installing, tracking, and supporting an application on individual workstations and laptops. For many organizations, this means the use of a dedicated infrastructure of deployment and management tools and often dedicated IT resources to manage the entire process.

In the end, the organization is left with a cumbersome patchwork of imaging, deployment, metering, and patching tools along with applications that reside locally on a disparate group of endpoints and OSs. The following list highlights further details regarding the shortcomings of traditional application deployment methods:

- *Inflexible*- Inability to quickly respond to business fluctuations and application changes.
- *Poor application security and control*- Traditional application deployments do not allow for dynamic control of application access.
- *Incompatibility issues*- Many organizations face application compatibility issues when attempting to use a traditional application deployment.
- *Complicated desktop images*- Due to the nature of traditionally installed applications and disparate OSs, many organizations opt to create base desktop OS images with pre-installed applications. This leads to management headaches for multiple desktop images with regard to updates and patches.

Virtualized Application Delivery

Application virtualization changes the entire method from deployment to delivery. At first glance, these two words may seem very similar, but in reality, these two methods differ greatly. Application deployment was all about installing and managing applications on multiple endpoints, but application delivery is about packaging an application once and delivering the application to multiple endpoints, for use, without the need for installation.

The power of application virtualization is the separation of the application from the endpoint OS. This alleviates all the shortcomings that come with a traditional deployment and provides a number of key benefits:

- *Ease of management* – Applications are now managed using a single image, so updates and patches are only applied once and delivered everywhere.
- *Dynamic applications* – Because of the ease of packaging and updating virtualized applications, IT can quickly respond to business changes.
- *No compatibility issues* – Every virtualized application is delivered into an isolated environment or sandbox on the endpoint. Thus, multiple versions of the same application can run on a single endpoint.
- *Data center centric* – All virtualized applications reside in the data center improving overall security and compliance of data and applications.
- *Branch and remote access* – Remote users have quicker access to updated applications without the lengthy process of deployment on the remote endpoints.

There are two primary methods of application virtualization in use across a growing number of vendors. The two methods can be categorized as agent-less and agent-based.

Agent-less application virtualization involves the use of an embedded virtual OS that is deployed as part of the virtualized application. These virtualized applications are fully encapsulated and able to run as a standalone executable from multiple locations such as a network drive, local drive, or USB drive.

Agent-based application virtualization utilizes a combination of a profiled or packaged application, a centralized delivery server, and a locally installed agent on the endpoint. The agents themselves utilize a kernel-mode driver or service. Some agent-based methods do not require the centralized delivery server and allow for shortcuts to be presented from a network share.

Both methods have the ability to stream the applications across the network to the local device. Streaming allows for immediate execution of the virtualized application from the endpoint while

data blocks are streamed on-demand across the network or Internet. As the application is used, additional blocks will be streamed for the features required. The streaming of applications helps to minimize the network overhead required to run virtualized applications.

Comparison

Outside the obvious architectural differences of the two methods, there are a number of key management and performance differences that should be taken into account when selecting a specific vendor. Two of the key differences I would like to focus on are centralized management and user mode execution.

Centralized management is the use of a centralized delivery server. This server allows for the publishing of virtualized applications and setting access controls on the applications using Active Directory (AD) group security. The ability to set access security on virtualized applications is typically a base requirement for most organizations and should be considered when looking at application virtualization vendors.

Note: Some agent-less vendor solutions do allow for group-level security to be set when the application is packaged, although this option is not as dynamic as the centralized management option.

User mode execution is the execution of the virtualized application within the user-mode of the endpoint OS. The advantage to this architecture is that there is no interaction with the kernel of the endpoint's OS, so if the virtualized application crashes, it will not affect the endpoint. Agent-based methods by their very nature utilize a kernel-mode driver and can cause an OS interruption if a virtualized application or agent were to crash.

Other Key Features to Look For

As you begin to shop around for your new shiny application virtualization solution, there are a few important features you should look for. I would consider these features to be very important for a well-rounded application and desktop virtualization initiative:

- *Off-line usage* – This feature allows virtualized applications to be launched even when the endpoint is disconnected from the network.
- *Streaming over HTTPS* – The ability to deliver virtualized applications over a secure SSL connection is vital in supporting a remote user base.
- *Broad platform support* – An important feature is the ability to run virtualized applications across a wide range of platforms including the option to deliver applications to a server-based computing environment.

Source :http://knowcitrixx.wordpress.com/xenapp-5-topics/application-virtualization/