

# Best Practices for Application Virtualization



**By Brien M. Posey**

TABLE OF CONTENTS

Introduction ..... 1

Can an Application be Virtualized? .....2

Should Your Application be Virtualized? .....5

Third Party Software is Essential..... 7

Conclusion.....8

*Application virtualization has become increasingly popular over the last few years, but organizations have discovered it isn't quite as simple as other virtualization technologies.*

**O**ne of the ongoing trends in enterprise IT is that organizations are increasingly choosing to virtualize their applications, especially those that implement virtual and remote desktop solutions. Traditional repackaging processes to prepare applications ready for enterprise deployments rely on Windows Installer (MSI) installation packages. Application virtualization solutions, such as Microsoft App-V, VMware ThinApp, or Symantec Workspace Virtualization; build virtual packages in proprietary formats. Application virtualization has become increasingly popular over the last few years, but those organizations that have chosen to virtualize applications have discovered that application virtualization isn't quite as simple as other virtualization technologies such as server virtualization. Though many applications can be virtualized without issue, there are others that aren't suitable for virtualization or are difficult to virtualize.

According to some estimates, organizations that implement this technology are typically able to virtualize 40-60 percent of their software. There have been some organizations that have reportedly achieved a conversion rate of nearly 95 percent, but achieving that benchmark is difficult and time consuming. Many organizations choose to set boundaries on how much effort they will put into application virtualization and if after a certain amount of time, the app can't be successfully converted to a virtual format, they will fall back to traditional installation technologies.

The application virtualization process creates a layer of isolation around virtualized applications. Some of the advantages to this approach include better protection against malware and the ability to run otherwise incompatible applications side-by-side on the same computer.

Unfortunately, application virtualization presents some challenges stemming from the way the abstraction layer isolates virtualized applications from one another. To put it simply, some applications have a symbiotic relationship with one another, and the isolation that comes with application virtualization interferes with that relationship.

Just because an application can be virtualized, it doesn't always mean that it should be virtualized. Organizations that want to virtualize their applications must evaluate each individual application to determine

***Organizations that want to virtualize their applications must evaluate each individual application to determine whether virtualization is technically possible.***

whether virtualization is technically possible, and whether or not it is practical. Whether an app should be virtualized is different than whether it can be virtualized. Consider both aspects.

When considering whether an app can be virtualized, look at behaviors like:

- Applications that start services at boot time
- Applications that require device drivers
- Applications that exist as shared dependencies
- Applications that are a part of the Operating System
- Applications that use COM+
- COM DLL surrogate virtualization
- Applications with complex licensing considerations

Flexera's AdminStudio Suite can test many of these things programmatically, identifying whether or not an app can be virtualized in seconds, instead of requiring hours of testing and investigation.

When considering whether an app should be virtualized, look at behaviors like:

- Applications with third party integrations
- Applications that support plugins
- Applications that are plugins
- Applications that are frequently updated or patched

### **Can an Application be Virtualized?**

Whether or not an application can be virtualized depends both on the application and on the virtualization software. A vendor agnostic third party solution can support Microsoft App-V, VMware ThinApp, and Symantec Workspace Virtualization, but since the dominant product in the market right now is App-V, this article will focus on App-V specific virtualization. Here are some of the behaviors to consider when deciding whether an app can be virtualized.

*Applications that come bundled with device drivers can present a significant challenge with regard to application virtualization.*

**Applications that Start at Boot Time:** Many systems require certain applications to start at system startup. These applications range in scope from anti-malware software to backup agents to databases. As a general rule, the current version of App-V can handle applications that start at boot time. However, you must ensure that the application starts in a timely manner. This shouldn't normally be a problem, but some have reported problems with virtualized applications that function as a system service. In some cases, it may be possible to improve the performance of such applications by experimenting with the service's startup type.

**Applications that Require Device Drivers:** Applications that come bundled with device drivers can present a significant challenge with regard to application virtualization. The reason for this is that device drivers are usually designed to run at the kernel layer, which means that if the driver is virtualized along with the application then the driver probably is not going to work correctly. For some apps, packaging engineers may be able to separate the application and its drivers into separate packages, leveraging virtualization for the application and traditional installs for the driver.

**Applications that Exist as Shared Dependencies:** Shared dependencies have long been a common problem for virtualized environments. The reason for this is that a virtualized application exists within what Microsoft refers to as the App-V bubble. This "bubble" is an isolation layer that may prevent the virtualized application from being seen by other applications.

Consider Java for example. If you virtualize Java then an application that depends on Java might not be able to detect that Java is installed on the system. Sure, Java can be sequenced in conjunction with the application that requires it, but if you have multiple applications that depend on Java then you really don't want to have to have a separate copy of Java for each.

The best solution to this problem is to take advantage of an App-V 5.0 feature called Run Virtual. By doing so, you can publish a package globally and then use registry keys to force the application to run within

*Still another reason why a particular application can not be virtualized is because the application has known compatibility issues.*

a virtual environment. Microsoft provides instructions for doing so at: <https://support.microsoft.com/en-us/kb/2848278>

**Applications that use COM+:** Conventional wisdom has long held that applications with a dependency on COM+ components cannot be virtualized. In actuality, it is often possible to virtualize such applications using App-V 5.0, but some additional work is required.

When you attempt to virtualize an application that requires a COM+ application, the sequencer will typically tell you which COM+ application you are going to need. Once you have that information, you can access the Windows Component Service Management Console and use it to export the COM+ application and add it to the package. You can find instructions for doing so at: <http://blogs.technet.com/b/gladiatorsft/archive/2013/11/08/app-v-on-com.aspx>

**COM DLL surrogate virtualization:** COM DLL surrogates pose a special problem for application virtualization. For those who might not be familiar with the term COM DLL surrogate, it refers to a DLL that runs within Dllhost.exe.

In the case of an application that relies on a COM DLL surrogate, it is necessary to determine how important the DLL is to the application. Some applications can work without the associated DLL surrogate and may provide slightly less functionality. If this loss of functionality is acceptable, then the application can be virtualized without the associated COM DLL surrogate. If the functionality is required or if the application cannot work without the COM DLL surrogate then the application should not be virtualized.

**Applications with complex licensing considerations:** Still another reason why a particular application can not be virtualized is because the application has known compatibility issues with the virtualization platform in question or because the application's license specifically forbids the application from being virtualized.

One of the best-known examples of such an application is Microsoft Internet Explorer. Microsoft has long cautioned its customers to avoid

*You should consider a third party solution, which can test your applications programmatically so that you can make a determination in seconds as to whether or not an application can be virtualized.*

virtualizing Internet Explorer. It's worth noting, however, that there are some application virtualization vendors that claim to be able to safely virtualize Internet Explorer.

As you can see, it is possible to virtualize applications that were once considered to be impossible to virtualize, thanks to some of the functionality that was introduced in App-V 5.0. Even so, virtualizing these types of applications can be tedious and time consuming. If you plan on virtualizing such applications then you should consider a third party solution, which can test your applications programmatically so that you can make a determination in seconds as to whether or not an application can be virtualized, rather than spending hours using trial and error.

### **Should Your Application be Virtualized?**

Just because an application can be virtualized does not mean that it should be. Some applications are poor candidates for virtualization in spite of the fact that it is technically possible to virtualize the application.

**Apps with third party integrations:** Some applications require third party integration. For example, there are a number of applications that require the Microsoft C++ run time. Although such applications can often be virtualized, the interaction between the application and the required third party component can sometimes make virtualization impractical and you must therefore consider the implications of virtualizing such applications.

**Apps that Support Add-ins or Plug-ins:** Some applications, such as Microsoft Office and various Web browsers support the use of add-ins or plug-ins. Although it may be possible to virtualize such applications, the virtualization process can make it difficult or impossible to take advantage of add-ins or plug-ins later on. These types of applications can be sequenced with all of the commonly used plug-ins, but linking new plug-ins to the application later on may require the application to be re-sequenced.

**Apps that are Add-ins or Plug-ins:** Plugins are easier to virtualize than they once were. In fact, App-V allows you to specify an application type as an add-in or plug-in. The trick to virtualizing add-ins or plug-ins is

*Virtualizing Java can prove problematic not just because of the way the software is isolated, but also because the frequent updates may require the application to be repackaged on a regular basis.*

that the parent application must be in place before the application is sequenced. If the parent application has been virtualized then the virtualized copy should be used. Another consideration is that if updates are made to the parent application then the updates could potentially break the virtualized add-ins or plug-ins.

**Patching/updating strategy:** Organizations using Microsoft App-V or a similar virtualization product should typically avoid virtualizing applications that are updated on a frequent basis. Java, for example, is updated very frequently. Virtualizing Java can prove to be problematic not just because of the way that the software is isolated, but also because the frequent updates may require the application to be repackaged on a regular basis. The repackaging process tends to be time consuming, but it can also be error prone unless the administrator strictly adheres to a long list of established best practices, and even then the newly created package requires extensive testing.

So why can't you simply update an application in the usual way? It's because of the isolation layer and the way the virtualization software makes the application available for use. Once again, application virtualization vendors have their own way of doing things, but oftentimes application updates or modifications are non-persistent unless the application is repackaged. For example, if a user were to download and install a patch for a virtualized application, that patch might only remain in place for the duration of the user's session. When the user signs out, the virtualized application may be reverted to its previous pristine state (depending on what virtualization software is being used and how it's configured).

At one time, it was impractical to virtualize dependency applications such as Java because such applications had to be sequenced with the applications that depended on them. This meant each time an update was released, both Java and the application that required Java had to be re-sequenced.

In App-V version 5, it is possible to sequence an application such as Java separately from other applications. However, a Java update would still require Java to be re-sequenced (or for the package to be

*When virtualizing applications, it's a good idea to adopt a third party solution such as Flexera's AdminStudio Suite to automatically evaluate applications for suitability with virtualized formats.*

updated). Although this might not involve quite as much work as what was required in the past, all of the applications that depend on Java must be tested to ensure that they still function correctly.

Keep in mind that the frequency with which an application is updated is not the only consideration that must be taken into account. The patch size can also make patching a virtualized application impractical.

### **Third Party Software is Essential**

Application virtualization offers tremendous benefits, but evaluating the could's and should's can be challenging. Often, packaging engineers have to take a trial and error approach to application virtualization, and just see which works.

When virtualizing applications, it's a good idea to adopt a third party solution such as Flexera's AdminStudio Suite to automatically evaluate applications for suitability with virtualization formats. AdminStudio Suite is designed to complement Microsoft App-V, VMware ThinApp, and Symantec Workspace Virtualization to help you determine whether an application can be virtualized, what specifically extra steps you may need to take to successfully convert the application into a virtual format.

The main reasons why you should consider using a third-party tool for the application virtualization process is because there are direct costs associated with virtualizing applications. Packaging, editing and testing an application tends to be a time-consuming process. This is especially true when it comes to the trial and error that is so often required for virtualizing tricky applications. Enterprise IT must consider the cost associated with the time it takes to create these packages. If the AppV repackaging time can be reduced, then costs are also reduced.

Costs also come into play with regard to package troubleshooting. The virtualization process can sometimes deliver results that are completely unexpected and when this happens, it's up to the administrative staff to figure out what went wrong. Depending on the complexity of the application, this can be a time-consuming process. A good third-party tool should not only offer help with the troubleshooting process, it should include features and capabilities

***Leveraging third party tools to proactively identify possible issues up front helps focus efforts for faster and higher quality conversion.***

that help the IT staff avoid making mistakes in the application packaging process in the first place.

## **Conclusion**

Most software vendors don't specifically design their applications to be virtualized and aren't familiar with the virtualization process. This leaves the packaging engineer on their own to resolve issues with virtualizing an application. Leveraging third party tools to proactively identify possible issues up front helps focus their efforts for faster and higher quality conversion.

AdminStudio Virtualization prepares software for reliable application virtualization deployment with a complete set of virtualization compatibility testing, automated conversion, validation, editing and management reporting capabilities. With support for all leading virtual application formats in a single tool, AdminStudio Virtualization fast tracks application virtualization and Virtual Desktop Infrastructure (VDI) projects.

Download a free trial of AdminStudio Suite <http://learn.flexerasoftware.com/content/AR-EVAL-AdminStudio>

---

*Brien M. Posey is a seven-time Microsoft MVP with two decades of IT experience, with expertise in SharePoint administration. During that time he has published thousands of articles and has written or contributed to dozens of IT books. Prior to becoming a freelance writer, Posey served as chief information officer for a national chain of hospitals and healthcare facilities. He has also worked as a network administrator for some of the nation's largest insurance companies and for the Department of Defense at Fort Knox.*

---

