

Virtualization for Windows:

A Technology Overview

 David Chappell, Chappell & Associates

September 2009

© Copyright Microsoft Corporation 2009. All rights reserved.

Contents

[Understanding Virtualization 3](#_Toc242081880)

[Virtualization Technologies 3](#_Toc242081881)

[Hardware Virtualization 4](#_Toc242081882)

[Desktop Virtualization 5](#_Toc242081883)

[Application Virtualization 7](#_Toc242081884)

[Other Virtualization Technologies 8](#_Toc242081885)

[Managing a Virtualized World 9](#_Toc242081886)

[Microsoft Virtualization Technologies 10](#_Toc242081887)

[Hardware Virtualization 10](#_Toc242081888)

[Hyper-V 10](#_Toc242081889)

[Windows Virtual PC 12](#_Toc242081890)

[Desktop Virtualization 13](#_Toc242081891)

[Remote Desktop Services Session Virtualization 13](#_Toc242081892)

[Microsoft Virtual Desktop Infrastructure (VDI) 15](#_Toc242081893)

[Another Approach: Microsoft Enterprise Desktop Virtualization (MED-V) 17](#_Toc242081894)

[Application Virtualization: Microsoft Application Virtualization (App-V) 18](#_Toc242081895)

[Managing a Virtualized Windows Environment 20](#_Toc242081896)

[System Center Operations Manager 2007 R2 20](#_Toc242081897)

[System Center Configuration Manager 2007 R2 22](#_Toc242081898)

[System Center Virtual Machine Manager 2008 R2 24](#_Toc242081899)

[Combining Virtualization Technologies 26](#_Toc242081900)

[Conclusion 28](#_Toc242081901)

[About the Author 28](#_Toc242081902)

# Understanding Virtualization

Virtualization is one of the hottest trends in information technology today. This is no accident. A variety of technologies fall under the virtualization umbrella, and all of them are changing the IT world in significant ways.

This overview introduces Microsoft’s virtualization technologies, focusing on three areas: hardware virtualization, desktop virtualization, and application virtualization. Since every technology, virtual or otherwise, must be effectively managed, this discussion also looks at Microsoft’s management products for a virtual world. The goal is to make clear what these offerings do, describe a bit about how they do it, and show how they work together.

## Virtualization Technologies

To understand modern virtualization technologies, think first about a system without them. Imagine, for example, an application such as Microsoft Word running on a standalone desktop computer. Figure 1 shows how this looks.

Figure 1: A system without virtualization

The application is installed and runs directly on the operating system, which in turn runs directly on the computer’s hardware. The application’s user interface is presented via a display that’s directly attached to this machine. This simple scenario is familiar to anybody who’s ever used Windows.

But it’s not the only choice. In fact, it’s often not the best choice. Rather than locking these various parts together—the operating system to the hardware, the application to the operating system, and the user interface to the local machine—it’s possible to loosen the direct reliance these parts have on each other.

Doing this means virtualizing aspects of this environment, something that can be done in various ways. The operating system can be decoupled from the physical hardware it runs on using *hardware virtualization*, for example, while *application virtualization* allows an analogous decoupling between the operating system and the applications that use it. Similarly, *desktop virtualization* allows separating an application’s user interface from the physical machine the application runs on. All of these approaches to virtualization help make the links between components less rigid. This lets hardware and software be used in more diverse ways, and it also makes both easier to change. Given that IT professionals spend most of their time working with what’s already installed rather than rolling out new deployments, making their world more malleable is a good thing.

Each type of virtualization also brings other benefits specific to the problem it addresses. Understanding what these are requires knowing more about the technologies themselves. Accordingly, the next sections take a closer look at each one.

### Hardware Virtualization

For most IT people today, the word “virtualization” conjures up thoughts of running multiple operating systems on a single physical machine. This is hardware virtualization, and while it’s not the only important kind of virtualization, it is unquestionably the most visible today.

The core idea of hardware virtualization is simple: Use software to create a *virtual machine (VM)* that emulates a physical computer. By providing multiple VMs at once, this approach allows running several operating systems simultaneously on a single physical machine. Figure 2 shows how this looks.

Figure 2: Illustrating hardware virtualization

Hardware virtualization is useful on both client and server machines. On clients, hardware virtualization can be used to deal with incompatibility between applications and desktop operating systems. For example, suppose a user running Windows Vista needs to use an application that runs only on Windows XP with Service Pack 2. By creating a VM that runs this older operating system, then installing the application in that VM, this problem can be solved.

Yet while hardware virtualization is useful on clients, the real excitement around this technology is focused on servers. The primary reason for this is economic: Rather than paying for many under-utilized server machines, each dedicated to a specific workload, hardware virtualization allows consolidating those workloads onto a smaller number of more fully used server machines. This implies fewer people to manage those computers, less space to house them, and fewer kilowatt hours of power to run them, all of which saves money.

Hardware virtualization also makes restoring failed server systems easier. VMs are stored as files, and so restoring a failed system can be as simple as copying its file onto a new machine. Since VMs can have different hardware configurations from the physical machine on which they’re running, this approach also allows restoring a failed system onto any available machine. There’s no requirement to use a physically identical system.

Since hardware virtualization is useful on both servers and clients, Microsoft offers a technology for each:

* Hyper-V: Part of Windows Server 2008 R2 (and its predecessor Windows Server 2008), Hyper-V provides hardware virtualization for servers.
* Windows Virtual PC: Part of Windows 7, this technology provides hardware virtualization for client machines. It’s the successor to Virtual PC 2007, which performs a similar function for Windows Vista and Windows XP.

Both of these technologies are useful in different situations, and both are described in more detail later in this overview.

### Desktop Virtualization

Much of the software people use most is designed to both run directly on a desktop machine. The applications in Microsoft Office are one common example, but there are plenty of others. While accepting this default is fine much of the time, it’s not without some downside. For example, organizations that manage many desktop machines must make sure that any sensitive data on those desktops is kept secure. They’re also obliged to spend significant amounts of time and money managing the applications resident on those machines. Freeing an application from a tight connection with a particular desktop machine—desktop virtualization—can help. Figure 3 illustrates one approach to doing this, an option called *session* *virtualization*.

Figure 3: Illustrating session virtualization, one style of desktop virtualization

As the figure shows, this approach to desktop virtualization allows creating *virtual sessions*, each interacting with a remote client machine. The applications executing in those sessions project their user interfaces remotely. Each session might run only a single application, or it might present its user with a complete desktop offering multiple applications. In either case, several virtual sessions can use the same installed copy of an application.

Session virtualization offers several benefits, including the following:

* Data can be centralized, storing it safely on a central server rather than on multiple desktop machines. This improves security, since information isn’t spread across many different systems.
* The cost of managing applications can be significantly reduced. Instead of updating each application on each individual desktop, for example, only the single shared copy on the server needs to be changed. This style of desktop virtualization also allows using simpler client operating system images or specialized client devices, commonly called *thin clients*, both of which can lower management costs.
* Organizations need no longer worry about incompatibilities between an application and a desktop operating system. While application virtualization can also solve this problem, it’s sometimes simpler to run the application on a central server, then use session virtualization to make the application accessible to clients running any operating system.

Microsoft provides several technologies for desktop virtualization. They include:

* Remote Desktop Services Session Virtualization: Formerly known as Windows Terminal Services, this approach to desktop virtualization has been part of Windows since Windows NT 4.
* Virtual Desktop Infrastructure (VDI): Based on Hyper-V, VDI provides an alternative way to create virtual desktops with operating systems such as Windows 7 and Windows Vista.
* Microsoft Enterprise Desktop Virtualization (MED-V): Using MED-V, an administrator can create Virtual PC-based VMs that include one or more applications, then distribute them to client machines.

Desktop virtualization covers a range of approaches. All are focused on the same thing, however: breaking the tight connection between a desktop application and a particular client machine.

### Application Virtualization

Virtualization provides an abstracted view of some computing resource. Rather than run directly on a physical computer, for example, hardware virtualization lets an operating system run on a software abstraction of a machine. Similarly, desktop virtualization can let an application’s user interface be abstracted to a remote device. In both cases, virtualization loosens an otherwise tight bond between components.

Another bond that can benefit from more abstraction is the connection between an application and the operating system it runs on. Every application depends on its operating system for a range of services, including memory allocation, device drivers, and much more. Incompatibilities between an application and its operating system can be addressed by either hardware virtualization or desktop virtualization. But what about incompatibilities between two applications installed on the same instance of an operating system? Applications commonly share various things with other applications on their system, yet this sharing can be problematic. For example, one application might require a specific version of a library to function, while another application on that system might require a different version of the same library. Installing both applications leads to problems, since one of them overwrites the version required by the other. To avoid this, organizations often perform extensive testing before installing a new application, an approach that’s time-consuming and expensive.

Application virtualization solves this problem by creating application-specific copies of all shared resources, as Figure 4 illustrates. The problematic things an application might share with other applications on its system—registry entries, specific DLLs, and more—are instead packaged with it, creating a *virtual application*. When a virtual application is deployed, it uses its own copy of these shared resources.

Figure 4: Illustrating application virtualization

Application virtualization makes deployment significantly easier. Since applications no longer compete for DLL versions or other shared aspects of their environment, there’s no need to test new applications for conflicts with existing applications before they’re rolled out. And as Figure 4 suggests, these virtual applications can run alongside ordinary applications—not everything needs to be virtualized.

Microsoft Application Virtualization, called App-V for short, is Microsoft’s technology for this area. An App-V administrator can create virtual applications, then deploy those applications as needed. By providing an abstracted view of key parts of the system, application virtualization reduces the time and expense required to deploy and update applications.

### Other Virtualization Technologies

This overview looks at three kinds of virtualization: hardware, desktop, and application. Similar kinds of abstraction are also used in other contexts, however. Among the most important are network virtualization and storage virtualization.

The term *network virtualization* is used to describe a number of different things. Perhaps the most common is the idea of a virtual private network (VPN). VPNs abstract the notion of a network connection, allowing a remote user to access an organization’s internal network just as if she were physically attached to that network. VPNs are a widely implemented idea, and they can use various technologies. In the Microsoft world, the primary VPN technologies today are Internet Security and Acceleration (ISA) Server 2006 and Internet Application Gateway 2007.

The term *storage virtualization* is also used quite broadly. In a general sense, it means providing a logical, abstracted view of physical storage devices, and so anything other than a locally attached disk drive might be viewed in this light. A simple example is folder redirection in Windows, which lets the information in a folder be stored on any network-accessible drive. Much more powerful (and more complex) approaches also fit into this category, including storage area networks (SANs) and others. However it’s done, the benefits of storage virtualization are analogous to those of every other kind of virtualization: more abstraction and less direct coupling between components.

## Managing a Virtualized World

Virtualization technologies provide a range of benefits. Yet as an organization’s computing environment gets more virtualized, it also gets more abstract. Increasing abstraction can increase complexity, making it harder for IT staff to control their world. The corollary is clear: If a virtualized world isn’t managed well, its benefits can be elusive.

For example, think about what happens when the workloads of several existing server machines are moved into virtual machines running on a single server. That one physical computer is now as important to the organization as were all of the machines it replaced. If it fails, havoc will ensue. A virtualized world that isn’t well-managed can be less reliable and perhaps even more expensive than its non-virtualized counterpart.

To address this, Microsoft provides a family of tools for systems management. To a large degree, the specifics of managing a virtualized world are the same as those of managing a physical world, and so the same tools can be used. This is a good thing, since it lets the people who manage the environment use the same skills and knowledge for both. Still, there are cases where a tool focused explicitly on virtualization makes sense. With System Center Operations Manager 2007 R2, System Center Configuration Manager 2007 R2, and System Center Virtual Machine Manager 2008 R2, Microsoft provides products addressing both situations.

A fundamental challenge in systems management is monitoring and managing the hardware and software in a distributed environment. System Center Operations Manager 2007 R2 is Microsoft’s flagship product for addressing this concern. By allowing operations staff to monitor both the software running on physical machines and the physical machines themselves, Operations Manager lets them know what’s happening in their environment. It also lets these people respond appropriately, running tasks and taking other actions to fix problems that occur. Given the strong similarities between physical and virtual environments, Operations Manager can also be used to monitor and manage virtual machines and other aspects of a virtualized world.

Another unavoidable concern for people who manage a computing environment is installing software and managing how that software is configured. While it’s possible to perform these tasks by hand, automated solutions are a much better approach in all but the smallest environments. To allow this, Microsoft provides System Center Configuration Manager 2007 R2. Like Operations Manager, Configuration Manager handles virtual environments in much the same way as physical environments. Once again, the same tool can be used for both situations.

Tools that work in both the physical and virtual worlds are attractive. Yet think about an environment that has dozens or even hundreds of VMs installed. How are these machines created? How are they destroyed? And how are other VM-specific management functions performed? Addressing these questions requires a tool that’s focused on managing hardware virtualization. Microsoft’s answer for Windows VMs is System Center Virtual Machine Manager 2008 R2. Among other things, this tool helps operations staff choose workloads for virtualization, create the VMs that will run those workloads, and move VMs from one physical machine to another.

# Microsoft Virtualization Technologies

Every virtualization technology abstracts a computing resource in some way to make it more useful. Whether the thing being abstracted is a computer, a desktop, or the environment that application runs in, virtualization boils down to this core idea. And while all of these technologies are important, it’s fair to say that hardware virtualization gets the most attention today. Accordingly, it’s the place to begin this technology tour.

## Hardware Virtualization

Many trends in computing depend on an underlying megatrend: the exponential growth in processing power described by Moore’s Law. One way to think of this growth is to realize that in the next two years, processor capability will increase by as much as it has since the dawn of computing. Given this rate of increase, keeping machines busy gets harder and harder. Combine this with the difficulty of running different workloads provided by different applications on a single operating system, and the result is lots of under-utilized servers. Each one of these server machines costs money to buy, house, run, and manage, and so a technology for increasing server utilization would be very attractive.

Hardware virtualization is that technology, and it is unquestionably very attractive. While hardware virtualization is a 40-year-old idea, it’s just now becoming a major part of mainstream computing environments. In the not-too-distant future, expect to see the majority of applications deployed on virtualized servers rather than dedicated physical machines. The benefits are too great to ignore.

To let Windows customers reap these benefits, Microsoft today provides two fundamental hardware virtualization technologies: Hyper-V for servers and Windows Virtual PC for desktops. The following sections describe both.

### Hyper-V

The fundamental problem in hardware virtualization is to create virtual machines in software. The most efficient way to do this is to rely on a thin layer of software known as a *hypervisor* running directly on the hardware. Hyper-V, part of Windows Server 2008 R2 (and its predecessor Windows Server 2008), is Microsoft’s hypervisor. Each VM Hyper-V provides is completely isolated from its fellows, running its own guest operating system. This lets the workload on each one execute as if it were running on its own physical server. Figure 5 shows how this looks.

Figure : Illustrating Hyper-V in Windows Server 2008 R2

As the figure shows, VMs are referred to as *partitions* in the Hyper-V world. One of these, the *parent* partition, must run Windows Server 2008 or Windows Server 2008 R2. *Child* partitions can run any other supported operating system, including Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, Windows 2000 Server, and Linux distributions such as SUSE Linux. To create and manage new partitions, an administrator can use an MMC snap-in running in the parent partition.

This approach is fundamentally different from Microsoft’s earlier server technology for hardware virtualization. Virtual Server 2005 R2, the virtualization technology used with Windows Server 2003, ran largely on top of the operating system rather than as a hypervisor. One important difference between these two approaches is that the low-level support provided by the Windows hypervisor lets virtualization be done in a more efficient way, providing better performance.

Other aspects of Hyper-V are also designed for high performance. Hyper-V allows assigning multiple CPUs to a single VM, for example, and it’s a native 64-bit technology. The large physical memory space this allows is useful when many virtual machines must run on a single physical server. Hyper-V also allows the VMs it supports to have up to 64 gigabytes of memory per virtual machine. And while Hyper-V itself is a 64-bit technology, it supports both 32-bit and 64- bit VMs. VMs of both types can run simultaneously on a single Windows Server 2008 machine.

Whatever operating system it’s running, every VM requires storage. To allow this, Microsoft has defined a virtual hard disk (VHD) format. A VHD is really just a file, but to a virtual machine, it appears to be an attached disk drive. Guest operating systems and their applications rely on one or more VHDs for storage. To encourage industry adoption, Microsoft has included the VHD specification under its Open Specification Promise (OSP), making this format freely available for others to implement. And because Hyper-V uses the same VHD format as Virtual Server 2005 R2, migrating workloads from this earlier technology is relatively straightforward.

Windows Server 2008 has an installation option called *Server Core*, in which only a limited subset of the system’s functions is installed. This reduces both the management effort and the possible security threats for this system, and it’s the recommended choice for servers that deploy Hyper-V. Systems that use this option have no graphical user interface support, however, and so they can’t run the Windows Server virtualization management snap-in locally. Instead, VM management can be done remotely using Virtual Machine Manager. It’s also possible to deploy Windows Server 2008 in a traditional non-virtualized configuration. If this is done, Hyper-V isn’t installed, and the operating system runs directly on the hardware.

Starting with Windows Server 2008 R2, Hyper-V provides Live Migration, the ability to move a running VM from one physical machine to another without interruption. Users of applications running on this VM don’t notice the move—everything keeps running as usual. This ability to move VMs across machines on the fly makes it easier to rebalance workloads across physical servers, allowing higher loads on those servers. It can also improve uptime, because the physical server a VM is running on can be shut down for maintenance or upgrades without stopping that VM—just move it to another server.

Hardware virtualization is a mainstream technology today. Microsoft’s decision to make it a fundamental part of Windows only underscores its importance. After perhaps the longest adolescence in computing history, this useful idea has at last reached maturity.

### Windows Virtual PC

The most commercially important aspect of hardware virtualization today is the ability to consolidate workloads from multiple physical servers onto one machine. Yet it can also be useful to run guest operating systems on a desktop machine. Windows Virtual PC, shown in Figure 6, is designed for this situation.

Figure 6: Illustrating Windows Virtual PC

While its predecessor Virtual PC 2007 runs on Windows Vista and Windows XP, Windows Virtual PC runs only on Windows 7. Like those earlier incarnations, Windows Virtual PC can run a variety of x86-based guest operating systems. The supported guests include Windows 7, Windows Vista, Windows XP, Windows 2000, and more. Applications running in these guest VMs can even appear in the Start menu, letting a user treat them like ordinary applications. Windows Virtual PC also uses the same VHD format for storage as Hyper-V, Virtual PC 2007, and Virtual Server 2005 R2.

As Figure 6 shows, however, Windows Virtual PC takes a different approach from Hyper-V. Unlike this server technology, the virtualization software runs largely on top of the client machine’s operating system. While this approach is typically less efficient than the Hyper-V approach, it’s fast enough for many, probably even most, desktop applications. Native applications can also run side-by-side with those running inside VMs, so the performance penalty is paid only when necessary.

## Desktop Virtualization

While hardware virtualization is certainly important, virtualizing desktops is also a big deal. Managing desktop systems can be expensive, and so providing virtualized desktops can potentially save lots of money. Given this, Microsoft provides several desktop virtualization technologies, each approaching this problem in a somewhat different way. This section looks at three of them.

### Remote Desktop Services Session Virtualization

Software today typically interacts with people through a screen, keyboard, and mouse. To accomplish this, an application can provide a graphical user interface for a local user. Yet there are plenty of situations where letting the user access a remote application as if it were local is a better approach. Making the application’s user interface available remotely through session virtualization is an effective way to do this.

The idea of session virtualization (formerly known as *presentation* virtualization) has been around for many years in the form of Windows Terminal Services. Beginning with Windows Server 2008 R2, this useful technology now sits under the umbrella of Remote Desktop Services (RDS). Figure 7 shows how it works.

Figure 7: Illustrating Remote Desktop Services Session Virtualization

Remote Desktop Services Session Virtualization is provided by a specific Windows Server 2008 R2 role: Remote Desktop Session Host. This role works with standard Windows applications—no changes are required. Using this technology, an entire desktop, complete with all application user interfaces, can be presented across a network. Alternatively, as Figure 7 shows, just a single application’s interface can be displayed on a user’s local desktop. This option relies on *RemoteApp*, a capability that first appeared in Windows Server 2008. With RemoteApp, an application’s user interface appears on the desktop just as if that application were running locally. In fact, an application accessed via RemoteApp appears in the Task Bar like a local application, and it can also be launched like one: from the Start menu, through a shortcut, or in some other way.

Both options—displaying a complete desktop or just a single application—rely on the Remote Desktop Connection, as the figure shows. Running on a client machine, this software communicates with Remote Desktop Services using the Remote Desktop Protocol (RDP), sending only key presses, mouse movements, and screen data. This minimalist approach lets RDP work over low-bandwidth connections such as dial-up lines. RDP also encrypts traffic, allowing more secure access to applications.

The Remote Desktop Connection runs on Windows XP, Windows Vista, and Windows 7. Clients are also available for earlier versions of Windows as well as Pocket PCs and the Apple Macintosh. And for browser access, a client supporting RDP is available as an ActiveX control, allowing Web-based access to applications.

Remote Desktop Services Session Virtualization also provides other support for accessing applications over the Web. Rather than requiring the full Remote Desktop Connection client, for example, the Remote Desktop Web Access capability allows single applications (via RemoteApp) and complete desktops to be accessed from a Web browser. The technology also provides a gateway that encapsulates RDP traffic in HTTPS. This gives users outside an organization’s firewall more secure access to internal applications without using a VPN.

Session Virtualization moves most of the work an application does from a user’s desktop to a shared server. Giving users the responsiveness they expect can require significant processing resources, especially in a large environment. To help make this possible, Remote Desktop Services Session Virtualization allows creating server farms that spread the processing load across multiple machines. Remote Desktop Services can also keep track of where a user is connected, then let him reconnect to that same system if the user disconnects or the connection is unexpectedly lost. While it’s not right for every situation, session virtualization can be the right choice for quite a few scenarios.

### Microsoft Virtual Desktop Infrastructure (VDI)

Another approach to desktop virtualization relies on the VMs that Hyper-V provides. This option, called Microsoft Virtual Desktop Infrastructure, is shown in Figure 8.

Figure 8: Illustrating Microsoft Virtual Desktop Infrastructure

As the figure shows, VDI runs on a Windows Server 2008 R2 machine configured as a Remote Desktop Virtualization Host. This server runs an instance of Windows 7 or Windows Vista in each of Hyper-V’s child partitions (i.e., its VMs). Both Windows 7 and Vista have built-in support for RDP, which allows their user interfaces to be accessed remotely. The client machine can be anything that supports RDP, such as a thin client, a Macintosh, or a Windows system. The result is similar to session virtualization: desktop applications run on a server with only their user interface projected across the network.

This similarity raises an obvious question: How should an organization choose between Session Virtualization and VDI? Here are some simple guidelines:

* Session Virtualization is a mature, proven technology, while VDI is newer and less commonly used. Session Virtualization is also less hardware-intensive than VDI; one server can support many more users, since it’s not required to run multiple VMs. Put simply, Session Virtualization is probably the best option if the primary goal is lowering the cost of providing user desktops.
* VDI gives users much greater control over their desktops, since everybody has his own VM. The user can set his own wallpaper and even reboot the VM without asking an administrator. While creating a VDI environment will probably cost more than relying on Session Virtualization, it can provide significantly more flexibility.

It’s also possible to use Session Virtualization and VDI together. To establish a connection in this scenario, an RDP client talks first to a connection broker, which then sets up an RDP connection to either a Remote Desktop Session Host or a Remote Desktop Virtualization Host.

### Another Approach: Microsoft Enterprise Desktop Virtualization (MED-V)

So far, desktop virtualization has meant providing remote access to applications running on a server. There’s another approach, however, exemplified by Microsoft Enterprise Desktop Virtualization. With MED-V, clients with Virtual PC 2007 installed can have pre-configured VM images running Windows XP or Windows 2000 delivered to them from a MED-V server. Figure 9 shows how this looks.

Figure 9: Illustrating Microsoft Enterprise Desktop Virtualization (MED-V)

A client machine might run some applications natively and some in VMs, as shown on the left in the figure, or it might run all of its applications in one or more VMs, as shown on the right. In either case, a central administrator can create and deliver fully functional VM images to clients. Those images can contain a single application or multiple applications, allowing all or part of a user’s desktop to be delivered on demand.

For example, suppose an organization has installed Windows Vista on its clients, but still needs to use an application that requires Windows XP. An administrator can create a VM running Windows XP and only this application, then rely on the MED-V Server to deliver that VM to client machines that need it. An application packaged in this way can look just like any other application—the user launches it from the Start menu and sees just its interface—while it actually runs safely within its own virtual machine.

## Application Virtualization: Microsoft Application Virtualization (App-V)

Both hardware virtualization and desktop virtualization are familiar ideas to many people. Application virtualization is a more recent notion, but it’s not hard to understand. As described earlier, the primary goal of this technology is to avoid conflicts between applications running on the same machine. To do this, application-specific copies of potentially shared resources are included in each virtual application. Figure 10 illustrates how Microsoft Application Virtualization (originally known as *SoftGrid*) does this.

Figure 10: Illustrating Microsoft Application Virtualization

As Figure 10 shows, virtual applications can be stored on a central machine running System Center Application Virtualization Management Server. (As described later, System Center Configuration Manager 2007 R2 can also fill this role—using this specialized server isn’t required.) The first time a user starts a virtual application, this server sends the application’s code to the user’s system via a process called *streaming*. The virtual application then begins executing, perhaps running alongside other non-virtual applications on the same machine. After this initial download, applications are stored in a local App-V cache on the machine, Future uses of the application rely on this cached code, and so streaming is required only for the first access to an application.

From the user’s perspective, a virtual application looks just like any other application. It may have been started from the Windows Start menu, from an icon on the desktop, or in some other way. The application appears in Task Manager, and it can use printers, network connections, and other resources on the machine. This makes sense, since the application really is running locally on the machine. Yet all of the resources it uses that might conflict with other applications on this system have been made part of the virtual application itself. If the application writes a registry entry, for example, that change is actually made to an entry stored within the virtual application; the machine’s registry isn’t affected.

For this to work, applications must be packaged using a process called *sequencing* before they are downloaded. Using App-V’s wizard-based Sequencer tool, an administrator creates a virtual application from its ordinary counterpart. The Sequencer doesn’t modify an application’s source code, but instead looks at how the application functions to see what shared configuration information it uses. It then packages the application into the App-V format, including application-specific copies of this information.

Storing virtual applications centrally, then downloading them to a user’s system on demand makes management easier. Yet if a user were required to wait for the entire virtual application to be downloaded before it started, her first access to this application might be very slow. To avoid this, App-V’s streaming process brings down only the code required to get the application up and running. (Determining exactly which parts those are is part of the sequencing process.) The rest of the application can then be downloaded in the background as needed.

Because downloaded virtual applications are stored in a cache provided by App-V, they can be executed multiple times without being downloaded again. When a user starts a cached virtual application, App-V automatically checks this application with the version currently stored on the central server. If a new version is available on the server, any changed parts of that application are streamed to the user’s machine. This lets patches and other updates be applied to the copy of the virtual application stored on the central server, then be automatically distributed to all cached copies of the application.

App-V also allows disconnected use of virtual applications. Suppose, for example, that the client is a laptop machine. The user can access the applications he’ll need, causing them to be downloaded into the App-V cache. Once this is done, the laptop can be disconnected from the network and used as usual. Virtual applications will be run from the machine’s cache.

Whether the system they’re copied to is a desktop machine or a laptop, virtual applications have a license attached to them. The server keeps track of which applications are used by which machines, providing a central point for license management. Each application will eventually time out, so a user with applications downloaded onto his laptop will eventually need to contact the central App-V server to reacquire the licenses for those applications.

Another challenge faced by App-V’s creators was determining which virtual applications should be visible to each user. To address this, virtual applications are assigned to users based on the Active Directory groups those users belong to. If a new user is added to a group, he can access his App-V virtual applications from any machine in this domain.

The benefits of using virtual applications with desktop and laptop computers are obvious. There’s also another important use of this technology, however, that might be less obvious. Just as applications conflict with one another on a single-user machine, applications used with Windows Remote Desktop Session Virtualization can also conflict. Suppose, for example, that an organization installs two applications on the same RDS server machine (commonly called just a *terminal server*) that require different versions of the same DLL. This conflict will be even more problematic then it would be on a user’s desktop, since it now affects all of the RDS clients that rely on this server. If both applications must be made available, the typical solution has been to deploy them on separate terminal servers. While this works, it also tends to leave those servers under-utilized.

Application virtualization can help. If the applications are virtualized before they’re loaded onto a terminal server, they can avoid the typical conflicts that require using different servers. Rather than creating separate server silos, then seeing those servers underutilized, virtual applications can be run on any terminal server. This lets organizations use fewer server machines, reducing hardware and administrative costs.

In effect, an App-V virtual application is managed less like ordinary installed software and more like a Web page. A virtual application can be brought down from a server on demand, like a Web page, and just as there’s no need to test Web pages for potential conflicts before they’re accessed, there’s no need to test virtual applications before they’re deployed. Once again, the underlying idea is abstraction: providing a virtual view of an application’s configuration information. As with other kinds of virtualization, the benefits stem from increasing the separation between different elements of the computing environment.

# Managing a Virtualized Windows Environment

The biggest cost in many IT organizations is salaries. If virtualization reduced hardware costs but led to increased management effort, it would likely be a net loss—people cost more than machines. Given this fact, managing virtualization technologies effectively is essential. This section describes how Microsoft’s System Center tools—Operations Manager, Configuration Manager, and Virtual Machine Manager—can be used to manage a virtualized Windows environment.

## System Center Operations Manager 2007 R2

For all but the smallest organizations, tools for monitoring and managing the systems in a distributed world are an inescapable requirement. Microsoft provides Operations Manager to address this challenge for Windows-oriented environments. Focused on managing hardware and software on desktops, servers, and other devices, the product supports a broad approach to systems management.

Computing environments contain many different components: client and server machines, operating systems, databases, mail servers, and much more. To deal with this diversity, Operations Manager relies on *management packs (MPs)*. Each MP encapsulates knowledge and more about how to manage a particular component, and each one is created by people with extensive experience in that area. For example, Microsoft provides MPs for managing Windows, SQL Server, Exchange Server, and nearly all of its other enterprise products. HP and Dell each provide MPs for managing their server machines, while several other vendors also provide MPs for their products. By installing the appropriate MPs, an organization can exploit the knowledge of a product’s creators to manage it more effectively. This includes managing an environment using virtualization, as Figure 11 shows.

Figure 11: Operations Manager 2007 R2 in a virtualized environment

As the system on the left shows, Operations Manager can manage virtual as well as physical machines. In fact, the product works the same way in both cases. Operations Manager relies on an agent that runs on each machine it manages, and so every machine—physical or virtual—has its own agent. In Figure 11, for example, the system on the far left would have two agents: one for the physical machine and one for the VM provided by Hyper-V. From the perspective of an operator at the Operations Manager console, both look like ordinary Windows machines, and both are managed in the same way. Rather than deploying different tools for managing physical and virtual environments, Operations Manager applies the same user interface and the same MPs to both worlds. And as the figure shows, Operations Manager 2007 R2 adds the ability to manage machines running Linux and Solaris. This includes management of virtual machines on those systems.

While managing physical and virtual machines is done with the same MPs, there are also specific MPs for managing virtualization technologies. The MP for Hyper-V, for example, allows an operator to enumerate the VMs that are running on a particular physical machine, monitor the state of those VMs, and more. The MP for Remote Desktop Services lets an operator track the performance and availability of this desktop virtualization technology, while the MP for App-V supports similar types of management operations. By applying the same technology to physical and virtual environments, Operations Manager helps provide a consistent approach to managing these two worlds.

## System Center Configuration Manager 2007 R2

Deploying the right software onto the right machines, then keeping that software up to date can be a herculean task. Add the challenge of maintaining a current record of software assets, and the value of an automated tool becomes clear. To address these challenges, Microsoft provides Configuration Manager, another member of the System Center family.

Challenging as it is in the physical world, managing software configurations can become even more challenging once virtualization is on the scene. Creating more virtual machines, for example, means more machines whose software must be updated. Effective configuration management is essential in this environment.

Like Operations Manager, Configuration Manager approaches the physical and virtual worlds in the same way. Rather than requiring separate tools for managing software configuration in these separate environments, Configuration Manager applies the same technology to both. Figure 12 shows how this looks.

Figure : Configuration Manager 2007 R2 in a virtualized environment

As the leftmost system in this figure illustrates, Configuration Manager treats a VM provided by Hyper-V as if it were a physical machine. Software can be installed on this machine, updated as needed, and appear as part of the asset inventory maintained by Configuration Manager. Similarly, as the middle system shows, this tool works with applications running on a terminal server just like any others.

Configuration Manager also works with App-V, as the system on the right in Figure 12 illustrates. As described earlier, an organization using App-V has a choice: It can distribute virtual applications using System Center Application Virtualization Management Server, part of App-V, or it can use System Center Configuration Manager 2007 R2. Using Configuration Manager doesn’t let virtual applications be streamed on demand to the system on which they’ll run, but it does allow using the same server to deploy both virtual applications and their non-virtual counterparts.

Managing software configurations is important in every organization. As the virtualization wave continues to roll across the IT world, managing virtualized software matters more and more. The goal of Configuration Manager is to provide a common solution to this problem for both physical and virtual environments.

## System Center Virtual Machine Manager 2008 R2

Many of the requirements for managing a virtualized environment are identical to those of a purely physical world. Operations Manager and Configuration Manager exploit this fact, viewing both in much the same way. But virtualization also brings its own unique management challenges. The most important example of this stems from hardware virtualization and the plethora of virtual machines it allows. As more virtual machines are created and used, the need for a tool focused solely on managing them also grows.

For example, while Hyper-V provides a tool for managing its VMs, this tool works on only a single physical machine. Once an organization has more than a handful of VMs spread across different physical machines, a centralized approach to managing them makes more sense. Virtual Machine Manager is Microsoft’s response to this need. As its name suggests, the tool is designed entirely for managing VMs. Figure 13 gives a simple illustration of how Virtual Machine Manager 2008 R2, the latest release, can be used.

Figure 13: Illustrating Virtual Machine Manager 2008 R2

As the figure shows, Virtual Machine Manager provides a central console, allowing many VMs to be managed from a single point. An administrator can use this console to check the status of a VM, see exactly what’s running in that virtual machine, move VMs from one physical machine to another, and perform other management tasks. And although the console provides a graphical interface, this interface is built entirely on Microsoft’s PowerShell scripting tool. Anything that can be done graphically can also be done from the command line using this language.

As Figure 13 also shows, Virtual Machine Manager can manage VMs created using three different technologies: Hyper-V, Microsoft Virtual Server 2005 R2 SP1, and VMware’s ESX Server. The management functions available are the same across all three. For example, to help administrators create VMs using any of these technologies, Virtual Machine Manager includes the New Virtual Machine Wizard. This tool provides a number of options for defining a new VM, such as:

* Creating a new VM from scratch, specifying its CPU type, memory size, and more.
* Converting a physical machine’s environment into a new VM, a process known as *P2V*.
* Creating a new VM from an existing VM.
* Converting an existing VM created using VMware into Microsoft’s VHD format.
* Using a *template*. Each template is a virtual machine containing a deployment-ready version of Windows that can be customized by the administrator.

Whatever choice is made, the wizard can examine performance data to help determine which physical machine should host this new VM, a process known as *intelligent placement*. Based on their available capacity and other criteria, the wizard ranks candidate servers from one to five stars. Once the administrator chooses a server, the tool then helps her install the new virtual machine on that system.

To make life easier for administrators, Virtual Machine Manager maintains a library of templates, VHDs, and other information. Along with creating new VMs using the contents of this library, an administrator can take an existing VM offline, store it in the library, then restore it later. Users can also create VMs themselves from the templates in this library through Virtual Machine Manager’s self-service portal. To help administrators remain in control, Virtual Machine Manager allows defining per-user policies, specifying things such as a quota limiting the number of VMs a user can create.

Another challenge in managing a virtualized environment is connecting the tool used to manage VMs with the tool used to monitor systems and applications. For example, suppose a physical machine hosting several VMs is running low on disk space. It can send an alert to a monitoring tool, but solving the problem might require moving some of its VMs onto another physical machine. The monitoring tool can’t do this, but the VM management tool can. Solving this problem requires connecting these two.

To address this, Virtual Machine Manager 2008 includes a facility called Performance and Resource Optimization (PRO) that lets it work together with Operations Manager. If a machine running low on disk space sends an alert to Operations Manager, for instance, Operations Manager can pass this information on to Virtual Machine Manager. This tool can then be used to move one or more VMs off this machine onto others. As mentioned earlier, Hyper-V now provides Live Migration, letting VMs be moved without interruption to their users. This addition can make PRO more useful in some situations. And starting with Virtual Machine Manager 2008 R2, the physical storage used by these VMs can also be moved, a feature that Microsoft calls Quick Storage Migration.

Operations Manager and Virtual Machine Manager can work together to let an administrator see what VMs are running on a machine, what applications each of those VMs is running, and more. All of this works regardless of whether the VMs are implemented using Microsoft technologies or VMware ESX. Microsoft’s goal is to make its management tools attractive to customers using any of these options.

Hardware virtualization, especially on servers, is fast becoming the norm. While single-machine tools for managing VMs are fine in simple scenarios, they’re not sufficient for the kind of widespread virtualization that’s appearing today. By providing a centralized console, a library to draw from, and other tools, Virtual Machine Manager aims at providing a single point for managing Windows VMs across an organization.

# Combining Virtualization Technologies

Looking at each virtualization technology in isolation is useful, since it’s the simplest way to understand each one. Yet using these technologies together is useful, too. Figure 14 shows an example scenario that combines hardware virtualization, desktop virtualization, and application virtualization.

Figure 14: Using different virtualization technologies together

In this example, the system on the left uses hardware virtualization provided by Hyper-V. One VM is running a workload on Linux, while the other is running System Center Configuration Manager 2007 R2 on Windows. This server provides App-V virtual applications to other systems in this organization. The machine at the top of the figure, for example, might be a desktop, laptop, or server machine, and some of its applications are App-V virtual applications streamed on demand. The system at the bottom is providing desktop virtualization using Remote Desktop Services, and all of the applications it runs are packaged as App-V virtual applications. As all kinds of virtualization continue to spread, multi-technology scenarios like this are becoming increasingly common.

One important issue that isn’t described in this paper is the impact of virtualization technologies on licensing. Traditional licenses are often wedded to hardware, a marriage of convenience that breaks down in a virtualized world. A different approach is needed, and so understanding the licensing requirements for these technologies is unavoidable. For example, using VDI with Windows Vista requires the Vista Enterprise Centralized Desktop product license, while other situations also have their own unique licensing requirements.

# Conclusion

The pull of virtualization is strong—the economics are too attractive to resist. And for most organizations, there’s no reason to fight against this pull. Well-managed virtualization technologies can make their world better.

Microsoft takes a broad view of this area, providing hardware virtualization, desktop virtualization, application virtualization, and more. The company also takes a broad view of management, with virtualized technologies given equal weight to their physical counterparts. As the popularity of virtualization continues to grow, these technologies are becoming a bedrock of modern computing.

# About the Author

David Chappell is Principal of Chappell & Associates ([www.davidchappell.com](http://www.davidchappell.com)) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technology.