

Realtime
publishers

"Leading the Conversation"

The Definitive Guide[™] To

Vista Migration

sponsored by



altiris®

*Danielle Ruest
and Nelson Ruest*

Chapter 6: Preparing Applications.....	140
Application Support in Vista	143
Vista Application Compatibility Features	146
Vista File and Registry Virtualization	148
64-bit Support for 32-bit Applications.....	148
Program Compatibility Assistant.....	149
Windows Installer version 4.0	151
Microsoft ACT version 5.0	152
In-house Development Compatibility Tools.....	154
Develop a Structured Application Management Strategy	155
The Application Management Lifecycle	155
Manage Commercial Software Licenses	157
Develop a Structured System Management Process.....	157
Work with a System Stack.....	158
Maintain Constant Inventories	160
Package Applications for Windows Installer.....	162
Explore Application Virtualization Options	165
Do away with Application Conflicts.....	167
Review your System Construction Strategy	169
Integrate Application Virtualization to the Migration Process	171

Copyright Statement

© 2007 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 6: Preparing Applications

Application or software management is one of the most challenging activities related to PC management and migration. Application incompatibilities, application conflicts, application installations, application delivery, application license management, application retirement are only a few of the issues you must master if you want to be in complete control of your desktop and mobile network. In fact, an entire science has been built around the management of software and applications with both manufacturers and experts weighing in to add their grain of salt.

As you might expect, the focus of this chapter is to help you make sense once and for all of how to prepare, distribute, manage and control applications in your network. First, some definitions:


- The term *program* refers to compiled code that executes a function.
- The term *application software* refers to programs designed to operate on top of *system software* such as an operating system. This differentiates between the OS and the programs that run on top of it.
- The term *software* usually refers to an off the shelf commercial program. This category includes items such as Microsoft Office, Adobe Acrobat, Corel Draw, Symantec Corporate Antivirus, and so on—all commercial products you can buy discretely for your organization.
- The term *application* usually refers to custom in-house development. This category includes anything that you develop in-house or that you have developed through outsourcing, but is a custom version of a tool that will only be used by your organization. It includes items such Web applications, line of business systems, or anything that is generated by tools such as Microsoft Visual Studio. It also includes user-developed programs such as those created with Microsoft Access or even just macros and templates created in Microsoft Office.

Many sources use the terms software and application interchangeably, but in an effort to avoid confusion as much as possible, this guide will use the term *applications* to refer to both applications and software. If a specific reference is required to either commercial software or in-house applications, they will be addressed as such.



You might wonder why applications are discussed before operating system images in this guide. In any migration project, the bulk of the engineering work is focused on application preparation and the larger the organization, the greater the effort required. This is why it is so important to stringently apply the rationalization principles outlined in Chapter 5—they help reduce this massive level of effort and keep it as minimized as possible. Since applications take considerable time to prepare, it is a good idea to begin this preparation process as soon as possible in the project. This way, applications will be ready if and when they are needed by the OS team as they build the standard operating environment (SOE) that will be deployed in your network.

There is no better time to build and implement a structured application management strategy than during a migration project. Each and every application in your network must be examined for compatibility, it must then be packaged, and then delivered to endpoints along with the new OS. Since you're making all this effort, why not take the time to revise your application management strategy and reconsider your approaches to application deployment?

 If you will be doing in-place upgrades, you will not have to redeploy every application to your desktops. Of course, few people opt for the in-place upgrade as it has a very poor reputation from previous versions of Windows. While the in-place upgrade actually works in Vista, you still need to remove and replace key applications such as anti-virus or other system utilities and you may need to provide corrections for some of the applications that exist on the upgraded systems. Whichever OS deployment method you choose, you need to review your application strategy during this deployment as you should in every deployment.

This is the goal of this chapter: to help you develop a structured application management strategy that will help ensure you are always compliant in terms of usage and licensing as well as responding to business needs. To do so, this chapter provides a structured look at each activity related to application management. These include:

- Windows Vista features in support of application operation
- Tools which assist in application compatibility testing
- The components of a structured application management strategy
- The components of a structured system management strategy
- Application packaging activities
- Application virtualization and application streaming
- Application distribution mechanisms
- Preparation for system and application deployment

Each of these items forms a cornerstone of a structured application management strategy.

You are now at application preparation in the deployment process (see Figure 1). Activities in this part of the project are performed by the PC team (see Figure 6.2). In fact, the PC team now begins several engineering activities that will continue throughout the Organize phase of the QUOTE system until every element of the solution is brought together for integration.

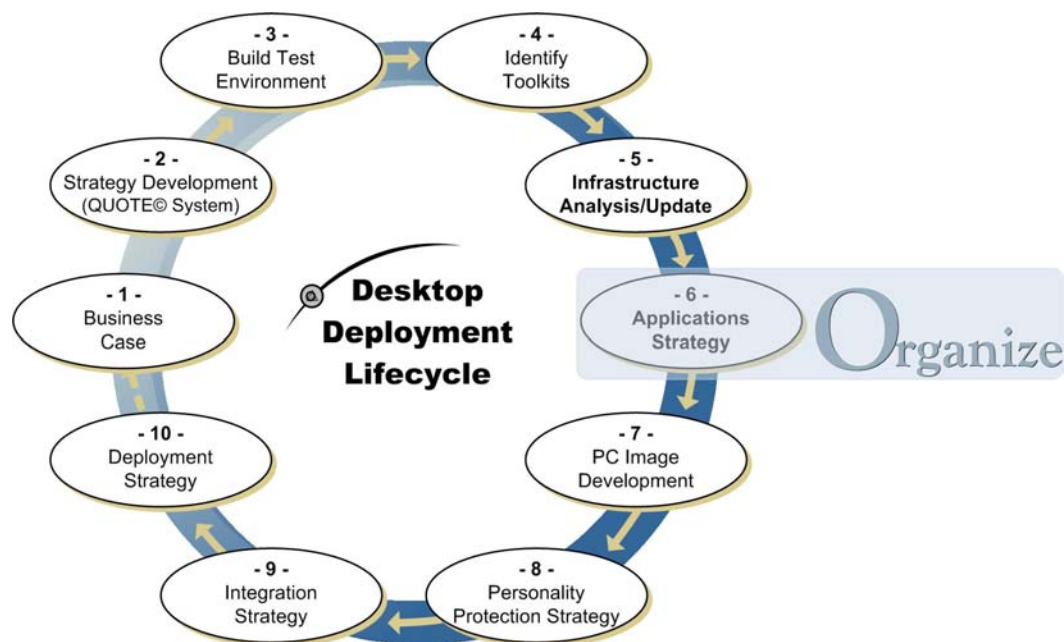


Figure 6.1. Moving through Step 6 in the DDL

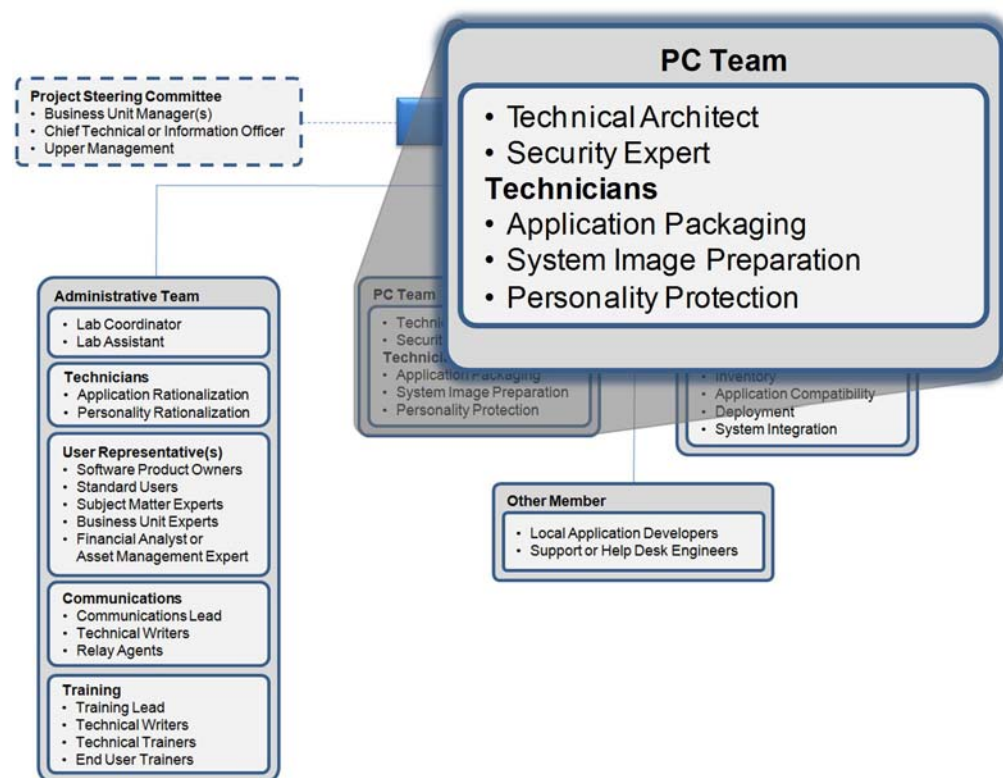



Figure 6.2. Activities in this step focus on the PC team

Application Support in Vista

By now, you have had a chance to play with and examine Windows Vista and are becoming familiar with its core feature set. You should also be aware that many, many things have changed in Vista and no longer work as they did in previous versions of Windows. This is the case for application support. Microsoft has changed the very structure that Windows uses in support of applications. Of course, Windows Vista still offers a central control environment that exposes all hardware and system resources to applications, but the way applications interact with the system is once again, different from previous Windows structures.

 For an overview of application compatibility in Vista, see **Application Compatibility in Vista, Working as a Standard User** available at: <http://www.reso-net.com/presentation.asp?m=7>.

In addition, like every version of Windows since Windows 2000, Vista includes the Windows Installer service (WIS). In Vista, WIS has been upgraded to version 4.0—a version that is only available for Vista or Longhorn Server. Ideally, each application you install on Vista will be able to take advantage of this service to integrate properly to the operating system.

 For more information on Windows Installer in general and Windows Installer 4.0, download **Working with Windows Installer**, a free white paper from Resolutions Enterprises Ltd. and AppDeploy.com from either <http://www.reso-net.com/articles.asp?m=8> or www.appdeploy.com/articles.

Several other items have changed in Vista and each of them affects applications in some way.

- Release-related changes:
 1. Version numbers have changed for Windows Vista. It is now version 6.0. Applications that look for specific versions and have not been updated to include version 6 of Windows will fail.
 2. Vista also includes a 64-bit version which is highly compatible to 32-bit, but has some key changes that affect application operation. Microsoft has not made any official statement to this effect, but Vista may well be Microsoft's last 32-bit desktop OS.
- Changes focused on security:
 3. User Account Control (UAC) now runs all processes with a standard user token. Applications that require elevated rights will fail unless they are Run as Administrator.
 4. Windows Resource Protection (WRP) has been enhanced to include registry keys as well as critical files and folders. Applications that write to protected areas of the registry or the Windows folder structure will fail.
 5. Session 0 is now completely restricted. Session 0 includes only kernel mode processes. Applications now run exclusively in user mode. Applications that need to operate in kernel mode will fail.
 6. The Windows Firewall has been enhanced. It now relies on the Windows Filtering Platform, a new system that filters at several layers in the networking stack and throughout the operating system providing better system protection. Applications that cannot take into account firewall restrictions will fail.

7. The Vista Web platform has been upgraded to Internet Information Services (IIS) version 7. IIS 7 now includes a completely componentized structure letting administrators install only those components that are required to deliver Web services. The logic is that a component that is not installed does not need patching and cannot become a security risk. Applications that have not been updated to operate with IIS 7 will fail.
8. The .NET Framework for Windows Vista has been upgraded to version 3.0. Managed code that is not compatible with .NET Framework 3.0 will fail though older versions of the .NET Framework can be installed on Vista.
9. The assembly identity grammar for managed code has been modified in Vista. The best example of this is the new logon screen and logon structure. Vista no longer relies on the Graphical Identification and Authentication (GINA), but uses the Credential Manager for logons. Applications that do not take this into account will fail.
 - Changes that may hinder the user experience:
10. Vista now supports Fast User Switching (FAS) even when joined to an Active Directory. Applications that do not support FAS may cause compatibility issues for end users.
11. Windows Vista sports a new Desktop Window Manager (DWM). DWM is designed to support several new features—Flip3D, a new structure for switching windows which includes content previews and content previews of open applications on the task bar—all of which will cause incompatibilities or at the very least inconveniences to users when applications do not support these new features.
12. Vista includes some shell or Windows Explorer changes that may affect applications. For example, the Documents and Settings folder has been replaced and split into two. The Users folder now includes all user data and the Program Data folder includes application settings. Applications that do not rely on variables and use hard-coded values will fail. Vista also includes new user interface themes. Applications that do not take advantage of these themes will cause operational issues.
 - Other changes that affect users:
13. Vista no longer supports kernel mode printer drivers, only user mode drivers are allowed.
14. Some components are deprecated and no longer available. These include FrontPage Server Extensions, Point-to-Point (POP3) services and Services for Macintosh. Other strategies are in place for these features. Windows SharePoint Services replace the FrontPage extensions and Services for UNIX replace the previous Macintosh services.
15. Two older Help File formats are also being deprecated. Vista will no longer use the CHM or HLP Help File formats. Help is now all based on XML data structures.

These fifteen system changes all affect application compatibility to some degree, but in many cases, there are workarounds. Here are the most common:

- Version checking can be corrected in a number of ways. It may be possible to edit the installation file to include the new Windows version, but in some cases this may not be enough especially if the application includes internal version checking that it relies on before operation. If you cannot change the internal version for the application, you might be able to run it in a Windows OS compatibility mode. Vista includes support for Windows versions from 95 to XP SP2.
- 64-bit versions of Vista will require applications that are at the very least 32-bit. That's because 64-bit versions no longer include any support for 16-bit applications. In some cases, Vista will automatically replace 16-bit installers with their 32-bit equivalents, but the ideal mitigation strategy is to run 64-bit applications on x64 versions of Vista.
- User Account Control will generate most of the compatibility problems you will face, especially during application installation. Applications may not install or uninstall properly because they cannot use elevated rights to do so. You can modify the installation logic of the application to make sure it requests elevation properly. This is done in the Windows Installer file for the application. Another option is to run the installation interactively with administrative credentials, but this would obviously only work in very small shops. Finally, if you cannot change the code, then you may have to use one of two options. The first is to give elevated rights to your users which is not recommended. Why bother having UAC if you don't use it? The second is to look to commercial application compatibility mitigation tools.
- Windows Resource Protection will also generate a fair share of compatibility issues. Applications that persist data in either the Program Files or the Windows folders fail because they do not have write access to these locations. The same goes for applications that try to replace system dynamic link libraries (DLL). WRP does not allow any changes to these key components. Mitigation strategies include running the application in compatibility mode; modifying the application so that it will write to the new C:\Users and C:\Program Data folder structures as well as in the HKEY_Current_User section of the registry; or once again, look to commercial application compatibility mitigation tools.



Note: Several products are designed specifically to overcome application access rights limitations. These products provide appropriate access rights on a per user basis when otherwise the application would fail due to WRP or UAC. They either use a local agent to do so or use Group Policy extensions to apply rights through Active Directory. These tools are better than modifying access rights directly for an application to run because of several reasons. First, the modifications they make are not permanent and do not affect the system for the long term. Second, the modifications are on a per user basis and therefore are not available to every user of the system. And third, these tools are policy-driven and centrally-controlled. This makes them good stop-gap measures when modifying the structure of an application is not possible.

Vendors of application compatibility mitigation tools include:


Altiris Application Control Solution: <http://www.altiris.com/Products/AppControlSol.aspx>

BeyondTrust Privilege Manager: <http://www.beyondtrust.com/products/PrivilegeManager.aspx>

- Session 0 issues or issues related to user versus kernel mode operation are also a cause of incompatibility. These issues are rare, but when they do occur, they break applications completely. In Vista, only services can run in session 0. In addition, session 0 does not support any user interface. Applications fail or worse crash when they try to display a user interface in session 0. To mitigate this issue, Vista includes the ability to redirect user interfaces from session 0 to user sessions. But the ideal mitigation is to update the application to use global objects instead of local objects and have all user interfaces displayed in user mode.





Other issues will be discussed as we address them, but you can see that there are a number of potential issues with applications running on Vista. Of course, if you have the means to upgrade every commercial application in your network and run at least proper 32-bit versions, then your issues will be limited, but few organizations have the ability to do this. In our experience, organizations running deployment projects will have budget for the upgrade of some, but not all applications. Applications such as productivity suites, antivirus applications and perhaps backup applications will be upgraded as part of the OS deployment project, but every other application will need to have a separate budget to pay for their upgrade. In most cases, they are not upgraded and are transferred over in their current state. This is another reason for rationalization.

It will often be custom in-house applications that will cause the most grief. Of course, if you use proper development strategies and program applications according to best practices and Windows Logo—the recommended structure Microsoft provides for applications running on Windows—guidelines, then you will have few issues, but this is not always the case. If you don't want to end up running a parallel redevelopment project for mission critical corporate systems, then you'll turn to the application compatibility mitigation strategies outlined in this chapter.

 For more information and access to a list of tools for application compatibility, go to the Vista Application Compatibility Resource Guide at <http://technet.microsoft.com/en-us/windowsvista/aa937621.aspx>.

Vista Application Compatibility Features

From the previous list of the fifteen system changes in Vista, you'd think that all you'll run into are application compatibility issues. Surprisingly, or perhaps not as Microsoft planned for it, very few applications have issues with Vista. Of all of the applications we run here at Resolutions, none of them had any issues. Of course, we had to upgrade anything that interacted at a low level in the operating system such as antivirus, disk defragmentation, or other utilities, but once that was done, most other applications worked just fine. In most cases, this is exactly what you'll find in your own networks.

-  Microsoft is keeping track of each application that passes either the designed for Windows Vista or the compatible with Windows Vista bar and is providing weekly updates through Knowledge Base article number 933305. See <http://support.microsoft.com/kb/933305> for more information. At the time of this writing this article listed 129 designed for Vista and 922 compatible with Vista applications.
-  Microsoft recently released an application compatibility update for Windows Vista. It is documented in Knowledge Base article number 929427 and can be found at <http://support.microsoft.com/kb/929427>. Make sure your systems include this update before you deploy them.
-  Also, the ieXBeta site includes a list of applications that work, that have issues and that don't work with Windows Vista at http://www.ieXBeta.com/wiki/index.php/Windows_Vista_Software_Compatibility_List.
-  The University of Wisconsin also publishes a list of compatible applications for Vista at <http://kb.wisc.edu/helpdesk/page.php?id=5175>.

Microsoft has documented known issues with applications in several tools they have made available for application and system compatibility checking. Chapter 1 introduced the Vista Upgrade Advisor (VUA). This tool installs and scans one machine at a time so it's not an enterprise tool by any means, but if you run it on some of the most typical and even the most challenging PC configurations in your network, it will give you a very good idea of the issues you're likely to run into. VUA requires administrative rights for both the installation and the scan you run—very weird things happen with VUA if you run it as a standard user. If you want a quick peek at how challenging your application picture will be, then run the scan on a few systems and print out the report.

For a more advanced application compatibility analysis, you'll want to use the Application Compatibility Toolkit (ACT) which is discussed below. Among other things, ACT can provide a systems-wide scan of all PCs and report on the state of their applications. But, it is also important to know which features Vista itself includes for application compatibility. There are several:

- File and Registry Virtualization
- 64-bit Support for 32-bit Applications
- Program Compatibility Assistant
- Windows Installer version 4.0

In addition, Microsoft has provided additional tools to assist in-house development projects in analyzing potential compatibility issues.

Vista File and Registry Virtualization

Because of some of the changes in the operating system and because of the implementation of Windows Resource Protection, Microsoft has instituted a small form of file and registry virtualization in Windows Vista. This is a small form of virtualization because Microsoft elected to provide only basic support for virtualization. For full application virtualization, organizations must look to commercial tools.

Vista's file virtualization is designed to address situations where an application relies on the ability to store files in system-wide locations such as Windows or Program Files. In these cases, Vista will automatically redirect the file to a folder structure called `C:\Virtual Store\SID\Program Files\...` where the SID is the security identifier of the user running the application.

Similarly, Vista's registry virtualization redirects system-wide registry keys. Keys that would normally be stored within the `HKEY_Local_Machine\Software` structure are redirected to `HKEY_Classes_Root\VirtualStore\Machine\Software`.

Vista-based virtualization does not work with every application but it does work. You'll have to test each suspect application to determine whether it interacts properly with this level of virtualization or if it needs repairs.

64-bit Support for 32-bit Applications


As mentioned earlier, x64 versions of Windows Vista do provide support for 32-bit applications but no longer run 16-bit applications. Each x64 system includes a Windows on Windows or WOW64 emulator which runs 32-bit applications. This WOW64 emulator will also be able to convert some well-known 16-bit installers into their 32-bit equivalents. This is done through the inclusion of some installer mappings within the registry.

But x64 systems do not support the installation of 32-bit kernel mode drivers. If kernel mode drivers are required, they must be 64-bit and they must be digitally signed. Digital signature of drivers ensures they are the actual drivers provided by the manufacturer and they have not been tampered with.

In addition, x64 systems make it easy to identify x86 processes—each process includes a *32 beside the executable name in Task Manager. They also include a special Program Files (x86) folder along with the standard Program Files folder to differentiate between installed 32 and 64-bit applications. For files installed in the Windows folder, a special SysWOW64 folder is used instead of the System32 folder. And the registry includes a special key called `HKEY_Local_Machine\Software\Wow6432Node` to store 32-bit information. The WOW64 emulator automatically redirects all 32-bit application requests to the appropriate supporting structure.

x64 systems offer better memory support than their x86 counterparts accessing up to 32 GB of RAM and 16 TB of virtual memory and because they do not have the operational limitations of 32-bit processors, 64-bit processors can grant a full 4 GB of memory to running 32-bit applications, something they will never be able to obtain on a x86 system.

Because of these support features in x64 systems, moving to an x64 platform does not have to occur all at once. You can begin with the move to the OS itself along with low level utilities and continue to run 32-bit applications. Then you can migrate your applications one at a time or on an as needed basis until your entire infrastructure is migrated to 64-bits. You should normally experience noticeable performance improvements as soon as you begin the process.

 For an overview of a migration to x64 systems, see **Move to the Power of x64**, a session presented at Microsoft Management Summit 2007 available at: <http://www.reso-net.com/presentation.asp?m=7>.

Program Compatibility Assistant

Microsoft also introduced the Program Compatibility Assistant (PCA) in Vista. PCA replaces the Program Compatibility wizard in the Help and Support as well as in the Compatibility tab of an executable's file properties in Windows XP. PCA is designed to automate the assignment of compatibility fixes to applications when they are required. PCA runs in the background and monitors applications for known issues. If an issue is detected, it notifies the user and offers solutions. Note that PCA is a client-only feature and is not available for servers and as such will not be in the Longhorn Server OS.

PCA can detect and help resolve several issues:

- Failures in setup programs
- Programs failures while trying to launch installers
- Installers that need to be run as administrator
- Legacy control panels that may need to run as administrator
- Program failures due to deprecated Windows Components
- Unsigned drivers on 64-bit platforms

In addition, PCA manages its own settings in the registry and will automatically inform users about compatibility issues with known programs at startup. PCA can also manage application Help messages. Programs can be excluded from PCA through Group Policy. Group Policy can also be used to control the behavior of PCA on PCs. After all, you don't want users being pestered by PCA messages once applications are deployed.

PCA automatically pops up when issues arise (see Figure 6.3) and will automatically reconfigure application compatibility settings based on known issues. These settings appear in the Compatibility tab of a program executable's file properties (see Figure 6.4).

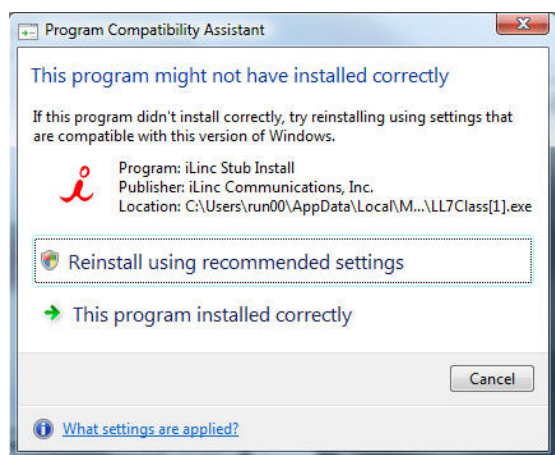


Figure 6.3. A Program Compatibility Assistant Message

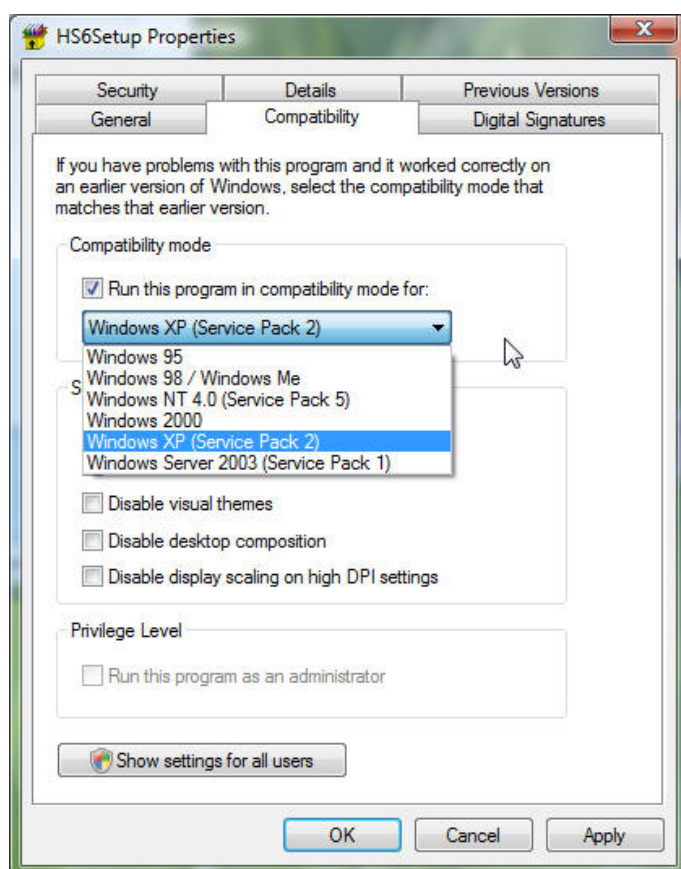


Figure 6.4. The Compatibility Tab demonstrates that Vista supports several compatibility modes

PCA modifications are applied at the individual user level in keeping with the user mode approach used by Vista. Settings can be modified for all users, but elevated privileges are required to do so. This lets PCA work even with standard user accounts.

Of course, PCA is not the solution for all applications, but it is a start and will make sure many applications that would normally fail will work properly in Vista without modification.

Windows Installer version 4.0

As mentioned earlier, Vista also includes a new version of Windows Installer, version 4.0. There are strong arguments made for the inclusion of all software installations to this service, mostly because of the powerful feature set it offers. WIS offers many features, but the most important are:

- Application self-healing—if untoward events occur in the structure of an installed application, WIS repairs it before it starts.
- Clean uninstallation and installation rollback—Because WIS relies on an installation database, it can undo any changes it makes to a system and do so cleanly in the event of a failed installation or in the event of a complete removal of an installed application.
- Elevated installation rights—WIS is designed to work with Group Policy Software Delivery to provide automatic rights elevation when applications are installed.
- Integrated patching capabilities—WIS will support in-line patching of installed applications as well as the update of installation logic to include patches prior to application installation.

There are many more features and readers interested in knowing more should look up the white paper mentioned at the beginning of this section ([Application Support in Vista](#)). Anyone using a standard application management strategy in Windows should endeavor to integrate all applications to WIS and many organizations have already done so. But, application installation packages that are designed as MSIs—the WIS file format—will not necessarily work with Vista because of the enhancements to WIS 4.0.

Many commercial application packaging tools support the evaluation of existing MSIs against the requirements of WIS 4.0 through the application of internal consistency evaluator (ICE) rules provided by the WIS 4.0 software development kit and report on required changes. WIS now supports several Vista features:

- The Restart Manager, a tool designed to stop and restart applications while patching occurs instead of causing a system restart.
- User Account Control to properly elevate rights during silent installations.
- User Account Control patching to ensure patches use proper rights elevation when applied.
- Windows Resource Protection to properly redirect files and registry settings when applications are installed.

The support of these new features should be carried through to all MSI packages. And, to make sure packages work properly, each MSI should include a manifest that describes the application components to the OS. It also might be easiest for organizations to digitally sign all software packages if they haven't already been signed by manufacturers to avoid future issues related to UAC once the application is deployed.



For example, software packaging products such as Altiris Wise Package Studio (<http://www.altiris.com/Products/WisePackageStudio.aspx>) allow organizations to not only upgrade existing packages to function with WIS 4.0, but also allow them to capture legacy installations and transform them into WIS 4.0 compatible installations.

Microsoft ACT version 5.0

Since applications form such a big part of every OS deployment project, Microsoft has endeavored to build tools that help mitigate migration efforts. The Application Compatibility Toolkit is one such tool. ACT has been around for several generations, but Microsoft took special care to provide as many features as possible in ACT version 5. As has been mentioned before, ACT requires a SQL Server database to store data. The server team should already have been preparing this database service in support of the migration effort. So, if you decide to use ACT, you can simply make use of the existing database service at installation.

ACT uses a simple three step process for compatibility evaluation:

1. First, it inventories applications and gathers application compatibility data through its built-in evaluators.
2. Then it lets you analyze applications, letting you prioritize, categorize, rationalize and otherwise consolidate applications. This is done partly through the tracking data that ACT can collect from end user systems. Additional data can be collected by synchronizing your local ACT database with the online Application Exchange.
3. Finally, it lets you test and mitigate compatibility issues if they arise by packaging fixes and corrections for the applications.

ACT includes several compatibility evaluators:

- User Account Control Compatibility Evaluator (UACCE) which runs on XP and detects when applications attempt to modify components a standard user does not have access to. It can also tell you whether the file and registry virtualization included in Vista would correct this behavior.
- Internet Explorer Compatibility Evaluator (IECE) which runs on both XP service pack 2 and Vista to detect issues related to IE version 6 and IE 7, especially the latter's execution in protected mode.
- Update Compatibility Evaluator (UCE) which runs on Windows 2000, XP and WS03 to detect applications that depend on files or registry entries that are affected by Windows Update.
- Windows Vista Compatibility Evaluator (WVCE) which enables you to evaluate issues related to the GINA deprecation, session 0 and other Vista deprecations.

ACT includes an Inventory Collector which is in the form of an executable package that must be installed on each local machine. Installation of the collector requires elevated rights. In managed environments where everyone is running as a standard user, you will need to use a deployment method to install this package. Once installed, the package runs for several days or weeks, inventorying the applications on a system, analyzing potential compatibility issues, and analyzing application use. It then reports back to the central ACT database, creating an entry for each system (see Figure 6.5).

Several tools are available for the remote installation of the collector package on a system. These tools are required because administrative rights are required for installation. If you already have a systems management system in place such as Altiris Deployment Solution or any other product, then you can rely on this tool for the deployment of the ACT package. If not, then you can either look to the recommendations in Chapter 4 to select one. But, deploying a systems management infrastructure may be overkill at this point. After all, you only want to install this one component and a deployment of a systems management infrastructure, if it is part of your project, will come as you deploy new OSes. Several more discreet tools can support this installation. For example, iTripoli (www.iTripoli.com) offers AdminScriptEditor (ASE). ASE allows you to generate scripts that can run in different administrative contexts, encrypting the credentials to protect them. You can therefore create a logon script that will run the installation with elevated privileges in complete confidence, letting you make use of ACT without having to relax any security measures.

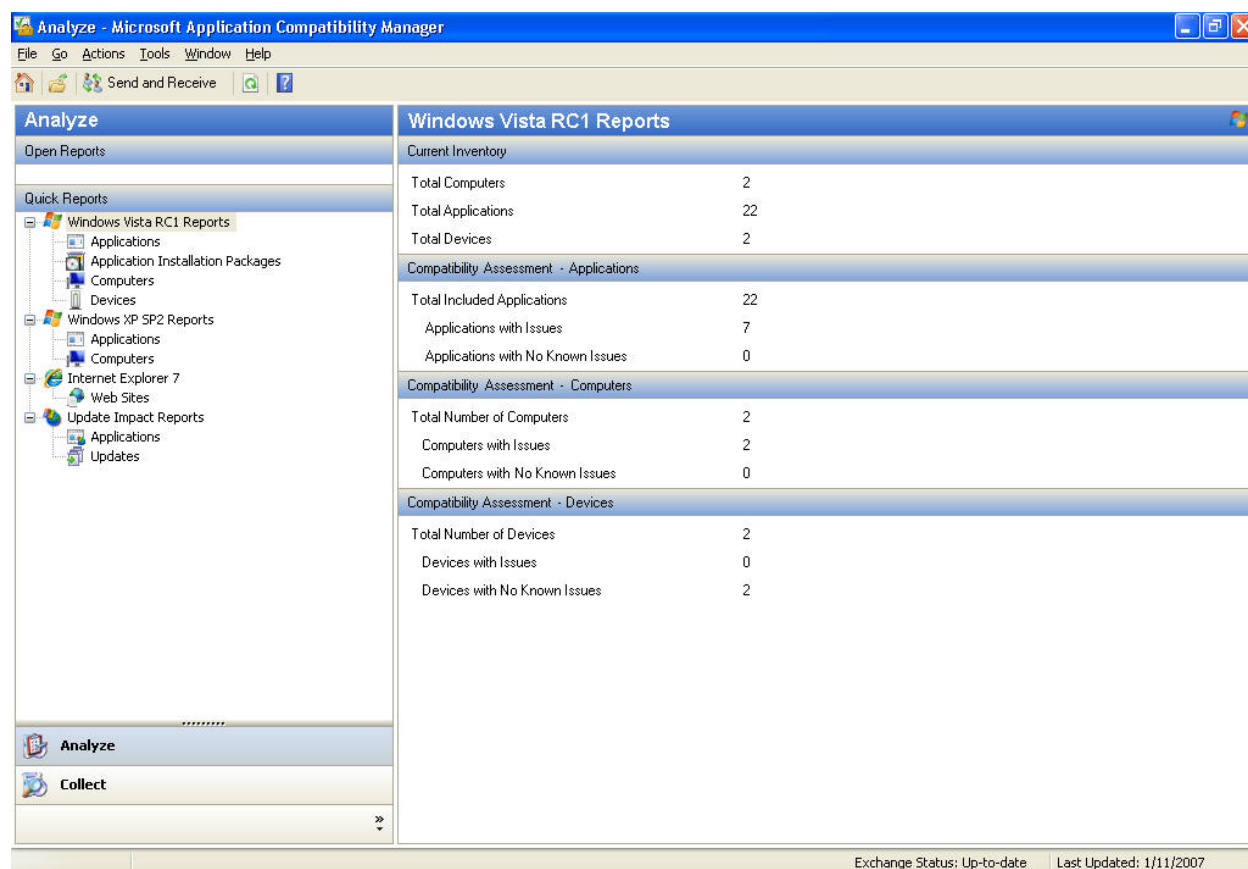


Figure 6.5. ACT provides a central repository of application compatibility data

The usage analysis is especially useful as you wouldn't want to spend any time with applications that are installed but not used. And, if you're so inclined, you can share your application data with the Microsoft Compatibility Exchange (MCE). MCE is a system that collects anonymous data on application compatibility. It relies on administrators like you to provide information to others. In return, you receive information others have shared on the applications that are found in your network. Microsoft also collects information from commercial application testing and the Microsoft Logo Certification program.

While the MCE is a great idea, its value is only as good as the data it stores. There are no real standards in regard to the structure of this data or the evaluation ratings organizations will submit to this program and it is unlikely organizations will submit data on their own internally-developed systems. But, in regards to commercial applications and the potential issues you may encounter in relation to them, it is a valid resource.


 More information on ACT can be found at <http://technet.microsoft.com/en-us/windowsvista/aa905072.aspx>. You can also rely on the BDD 2007 guide for Application Compatibility: <http://www.microsoft.com/technet/desktopdeployment/bdd/2007/AppCompact.mspx>.

In-house Development Compatibility Tools


ACT doesn't only include tools for IT professionals. It also includes tools for developers. ACT includes three tools for developers—tools for testing application setups, tools for testing Web sites with IE7 and tools for testing applications with UAC. Specifically, these tools are:

- The Setup Analysis Tool (SAT) which will verify that installers will work correctly and avoid the issues that make them fail such as kernel mode driver installation, 16-bit components, GINA components or modification of components protected by WRP.
- IE Test Tool which collects any issues related to Web sites and will upload the data to the ACT database.
- Standard User Analyzer (SUA) which identifies any issues related to applications running under UAC.

But ACT isn't the only source of information for developers. Microsoft has put together the Application Compatibility Cookbook which was mentioned in Chapter 1 as well as the Windows Vista Application Development Requirements for UAC Compatibility guide. The first details application compatibility changes in Vista while the second identifies how to design and develop UAC compliant applications. A third guide outlines Best Practices and Guidelines for Applications in a Least Privileged Environment.

 The Application Compatibility Cookbook can be found at <http://technet.microsoft.com/en-us/windowsvista/aa905072.aspx>, the UAC Compatibility guide can be found at <http://msdn2.microsoft.com/en-us/library/aa905330.aspx>, and the Best Practices and Guidelines for Applications in a Least Privileged Environment guide can be found at <http://msdn2.microsoft.com/en-us/library/aa480150.aspx>.

Microsoft also produced a Vista Readiness Hands on Lab which works on a Vista virtual machine and runs you through the most common issues you'll face with Vista. This lab helps you learn how these potential issues can affect your own applications.

 The Vista Readiness Hands on Lab can be found at http://www.microsoft.com/downloads/details.aspx?FamilyID=5F0C8D20-A8D5-4DA3-A977-E039A40D67D5&displaylang=en&mg_id=10049.

Finally, Aaron Margosis, a senior consultant with Microsoft Consulting Services, has developed a nifty little tool to test application compatibility with User Account Control. The Limited User Access (LUA) Buglight is a free tool that scans an application as it runs to identify any activity that requires administrative rights. Once these activities are identified, it is easier to either correct the code, correct the application's configuration or try running it in a compatibility mode because you know what specifically needs to be fixed. Aaron's blog also provides a lot of information on potential solutions for running applications in 'non-admin' mode.

 LUA Buglight and Aaron's blog can be found at
http://blogs.msdn.com/aaron_margosis/archive/2006/08/07/LuaBuglight.aspx.

Relying on these resources should greatly reduce the likelihood of running into issues when you try to run your custom applications on Vista.

Develop a Structured Application Management Strategy

With all of these resources to assist your assessment of compatibility for your applications, the process should be relatively smooth. Next, you'll need to prepare the applications themselves. This is a fairly complex process as you run through each application, evaluate its compatibility, prepare mitigations if required, package the application, test deployment and uninstallation and then have the application validated by a subject matter expert prior to deployment. Considering that organizations often have a large ratio of applications per users—a ratio that tends to increase with the number of users in an organization—it is easy to understand why this process is the process that will require the largest amount of effort and resources in the project. Fortunately, there are ways to reduce this level of effort. One of the best ways to do this is to implement a lifecycle approach to application management.

The Application Management Lifecycle

Few organizations know offhand what software can be found in their network. Fewer still can guarantee that there are no unused software products on their users' PCs. This issue stems from the very nature of distributed systems, but it can be circumvented with the introduction of an application lifecycle management process.

This lifecycle is based on four major phases and can be applied to both commercial software products and corporate applications, though there are slight variations in the initial phases. The four phases include:

- **Commercial Software Evaluation or Application Preparation:** This involves the identification of the requirement followed by the selection of a commercial software product and/or the design of a corporate application.
- **Software Implementation:** This phase focuses on software packaging, quality assurance testing and deployment.
- **Maintenance:** This phase focuses on ongoing support activities for the product. It will involve the preparation, testing and distribution of scheduled updates.
- **Retirement:** The final phase is focused on removal of the product from the network due to obsolescence or on the reallocation of the product to someone else. A removal may be followed by a replacement which would initiate the lifecycle process once again.

Every application has a lifecycle. It begins the moment the software development project is initiated by a manufacturer until the moment the software is retired from the marketplace. For user organizations, the lifecycle focuses more on when it is acquired, when it is deployed, how it is maintained and supported and when it is retired from the network. In the case of custom corporate applications, it begins the moment corporate developers begin to work on the project until the product is retired from use (see Figure 6.6).

Every application also requires patching during its lifecycle in the network. If you adopt an application early in its lifecycle, you will need to patch it once it is deployed. If you adopt it later in its lifecycle, you will most likely be able to patch it before it is deployed. Whichever method you use, you will need to make sure your application management processes take both pre-deployment and post-deployment patching into account.

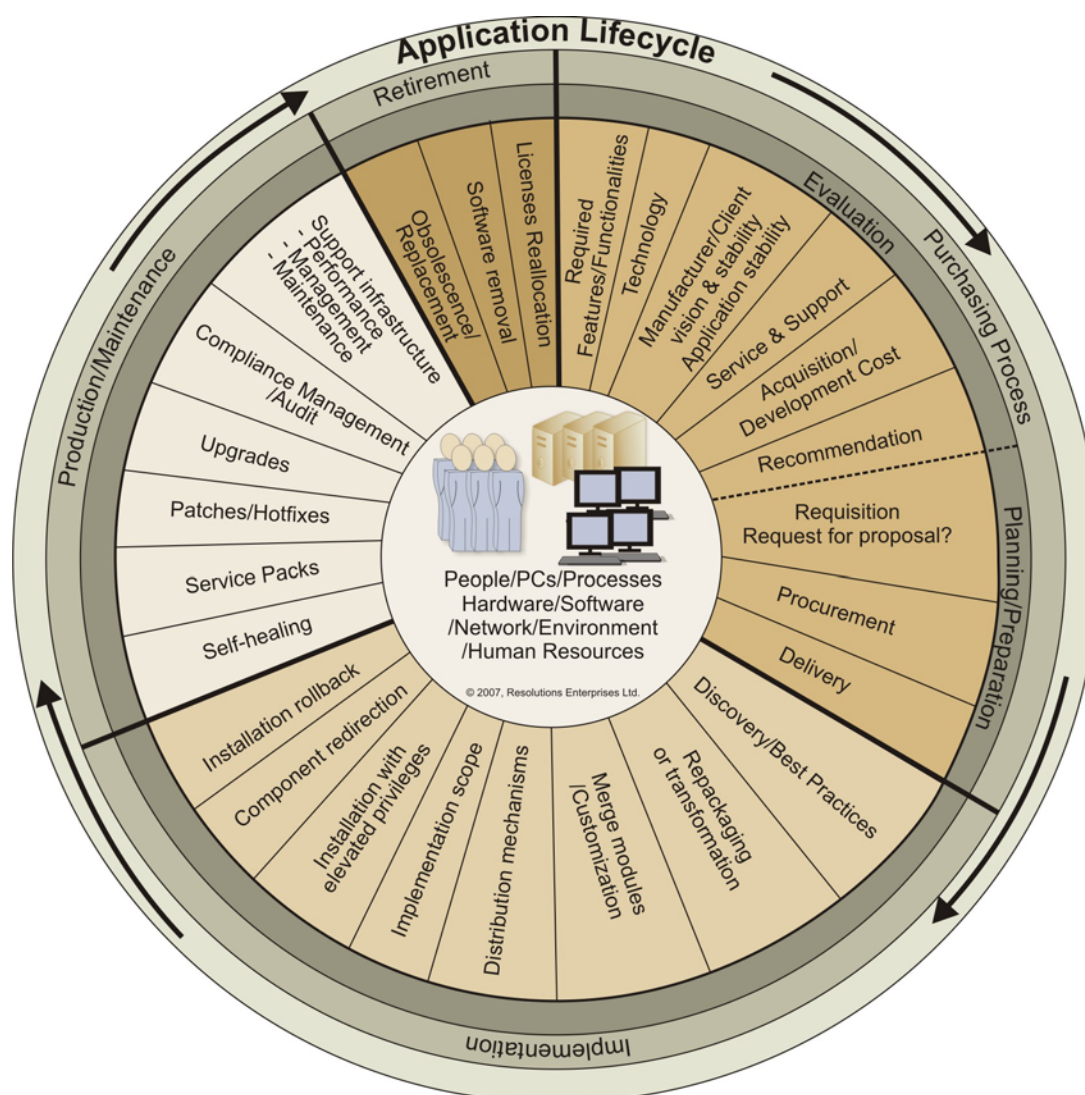


Figure 6.6. The Application Management Lifecycle

Manage Commercial Software Licenses

You also need to include licence management as well as performance assessments in your maintenance task list. With the advent of compliance regulations for organizations of all sizes, licence management takes on a much more important role in application management today. The easiest way to perform license management is to use a proper application allocation process and make sure that it runs through the entire lifecycle, especially all the way to removal when the product is no longer needed by the user it was deployed to. There is no justification today for an incomplete or inadequate application lifecycle management strategy.

Application removal is probably the most unused portion of the entire software lifecycle, yet it is the most important if organizations want to maintain a legal and compliant network. The issue is simple. When users move from position A to position B, they change their role within the organization. With new roles come new tasks, IT groups are quite adept at making sure users' PCs are updated with the applications required to meet their new requirements because if they don't users will be quick to log a support call. However, the same IT groups are not quite so adept when it comes to removing unused applications from the same PC.

That's because application removal is viewed as complex and cumbersome. This myth needs to be dispelled. Many IT professionals still think that the problem with application removal is that it is seldom effective. Modern applications are made up of a series of private and shared components. Removing any of these components can affect the stability of the system. Because of this, IT often makes the decision to opt for stability at the expense of legal compliance. After all, systems undergo regular hardware maintenance at which time they are reinstalled anyway. If the system isn't compliant, it is only for a short period of time.

This is one more justification for packaging applications to work with the Windows Installer service. WIS is just one feature that Microsoft has embedded into Windows operating system in support of application stability, but it is the one with the most impact because you can control the interaction of your applications with this service. WIS fully supports the complete and effective removal of every application component from a system so long, of course, as the application was installed using WIS in the first place. Proper application packaging methods will involve complete package testing and quality assurance including proper and non-damaging removal of packaged applications. This is why packaging is such an important part of application preparation in migration projects.

Develop a Structured System Management Process

Migration projects could not work without some form of automated application delivery to go along with the automated OS delivery the project will support. Much has been said to date about the various tools or suites you can use to do this. The fact is, medium to large organizations often run from 200 to 500 applications both in-house and commercial within their network. Managing application installations and maintenance for several hundred products can be complex. While the application lifecycle will help because it lets you understand all of the activities required to manage the application in your network, you'll find that it also requires another model to help simplify management issues: the system stack.

Work with a System Stack

Using a system stack simplifies the application management process because it structures how computer systems are built. Few organizations will ever need to install 200 to 500 products on the same computer. Using a model for system design will ensure that applications are categorized properly and regrouped into families that work together to provide the functionality required to fulfill specific job roles within your organization. Resolutions has been promoting the Point of Access for Secure Services (PASS) model for more than 10 years (see Figure 6.7). Organizations relying on this system stack have a proven track record of proper system and application management.

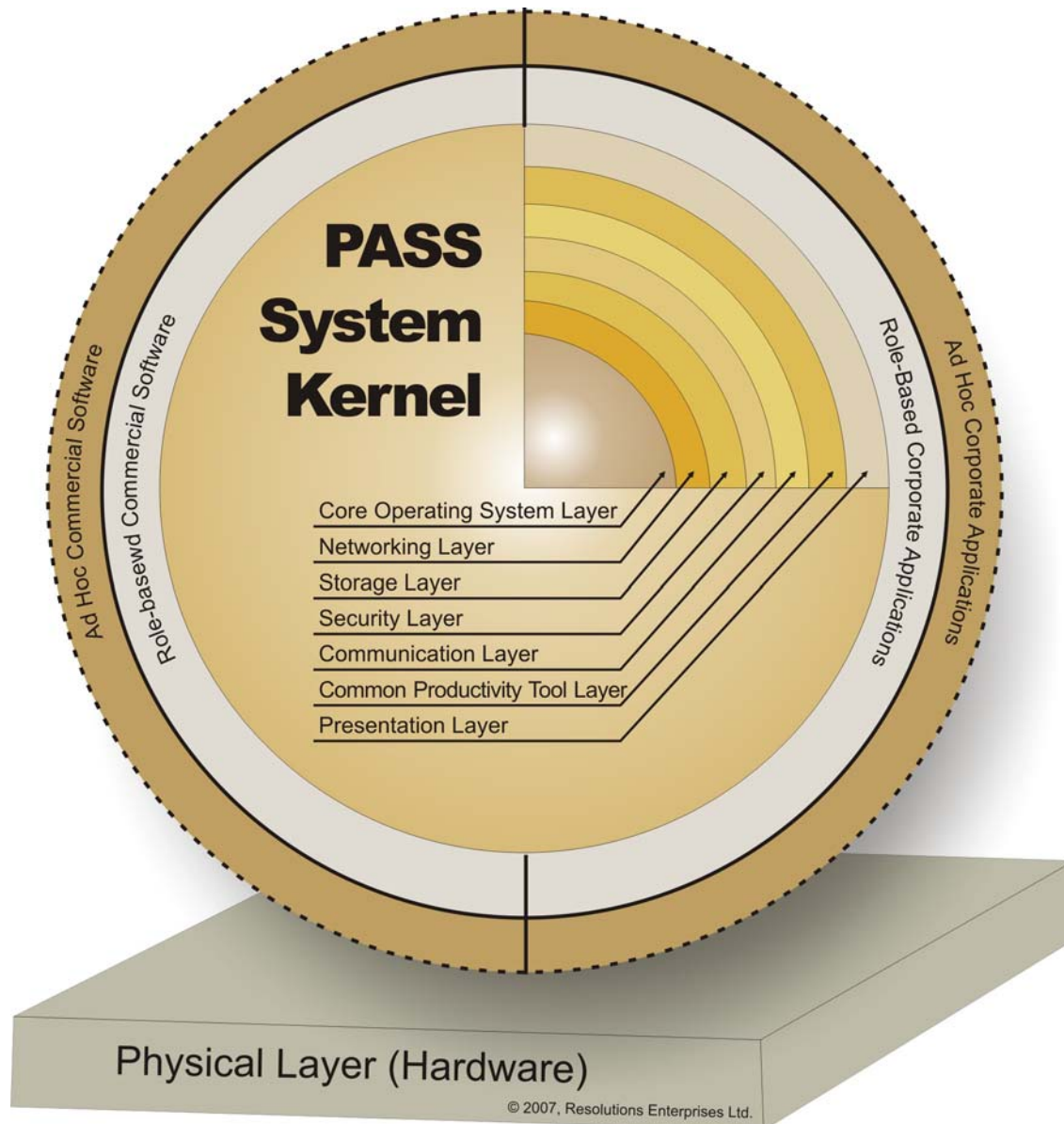



Figure 6.7. The PASS System Stack

This system stack is based on the construction of a computer system that responds to corporate needs in three ways:

- The **PASS system “kernel”** is designed to meet the needs of the average or generic user. It contains all of the software components required to perform basic office automation and collaboration tasks. In addition, it is divided into a series of layers similar to the OSI Networking Model. Like the OSI Model, it uses seven layers to provide core corporate services. Because its functionalities are required by all personnel, this kernel is installed on all computer systems. More on the kernel will be discussed in Chapter 7 as we prepare the OS system image for deployment.
- **Role-based applications and commercial software** are added on top of the kernel to meet the requirements of the special Information Technology roles every person plays within the organization.
- Finally, an **ad hoc layer** responds to highly specialized IT requirements that are often expressed on an individual basis. This ad-hoc layer can be applied at any time and traverses traditional vertical IT roles.

 In the PASS model, the kernel is considered as a closed component that is reproduced on all systems. Layers that are located beyond the kernel are considered optional for all systems.

Constructing systems based on a system stack such as the PASS model greatly reduces system management efforts because it reduces the number of programs that must coexist on any system. First, a good portion of systems, sometimes up to 50 percent or even more, will only require the system kernel. Remember that the kernel should contain every single program that is either royalty-free and required by the entire organization (for example, Adobe’s Acrobat Reader or the new Microsoft XPS Document Reader), every program that is mandated by internal policies—antivirus tools for example, or every program for which the organization obtains an enterprise-wide license (for example, many organizations obtain an enterprise license of Microsoft Office).

Second, by grouping programs into role-based configurations, organizations are able to reduce the number of applications that must coexist on a system. Role-based configurations include every program that is required by every member of the IT role grouping. For example, Web Editors would require a Web editing tool, a graphics tool, a Web-based animation tool, and other Web-specific utilities. This group of tools can be packaged separately, but should be delivered as a single unit on all systems belonging to the IT role. Role-based configurations often include no more than 10 to 30 individual programs depending on the role. Only these groupings need to be verified with each other and against the contents of the system kernel. There is no requirement to verify or test the cohabitation of programs contained in different configurations because they are not likely to coexist on the same system.

Third, ad hoc programs reduce system management efforts even further because they are only required by very few users in the organization. They are still packaged to enable centralized distribution and automated installation, but once again, they only require testing against both the kernel and the configurations they will coexist with but, because of their ad hoc nature, they may coexist with all possible configurations.

As discussed earlier, each application has a lifecycle of its own that is independent of its location within the system construction model. The difference lies in the rate with which you apply lifecycle activities to the application. Components of the kernel will have an accelerated lifecycle rate—since they are located on all systems, they tend to evolve at a faster pace than other components because they are supported by corporate-wide funding—while products within the outer layers of the model will have slower lifecycle rates which will be funded by the groups that require them. Ideally, the rate of evolution of these products will be monitored by the subject matter experts or application sponsors your organization identifies for each non-kernel application.

- ☞ Application sponsors are responsible for several activities, four of which are:
- Subject matter expertise for the application.
 - Acceptance testing for the application package.
 - Application monitoring or watching for new versions or patches.
 - Rationalization justifications or justifying why the application should be in the overall software portfolio.

Maintain Constant Inventories

Another key aspect of the application management lifecycle process is the maintenance and upkeep of corporate inventories. This is another area where a system stack like the PASS model can play a role because of the way it is designed to work. With a system stack, maintaining an inventory need only focus on identifying role groupings for role-based configurations. Applications that are contained in the kernel already have corporate-wide licenses so they are easy to track. Similarly, ad hoc products are only located on a few machines which also makes them easy to track.

This leaves the vocational groupings as they become the mainstay of the inventory system. Constant application inventories should be directly integrated to application management practices which should include several elements (see Figure 6.8).

- **Package Repository:** A central deposit for all authorized applications for the network. This repository is the source for all application distributions within the organization.
- **System Kernel Inventory:** The system kernel must be completely inventoried. This inventory will be linked to the Package Repository because many of its components will be stored in packaged format to facilitate automated system construction and conflict resolution.
- **Role-based Configuration Deposit:** This deposit identifies each of the packages found within each configuration. It is tied to the Package Repository to support system construction and vocational changes.
- **Vocational Groupings:** This deposit regroups all of the users belonging to a given IT role. It is tied to the Role-based Configuration Deposit because it identifies which configurations systems require on top of the system kernel. Ideally, these groupings will be stored within your organization's directory service (for example, Active Directory) since each grouping contains only user and machine accounts.

- Core Inventory Database:** The core inventory database brings everything together. It includes inventories of all computer systems in the network, all user accounts, all application components, and much more. It is used to validate that systems contain only authorized components as well as assign ad hoc product installations since these are mostly performed on a case-by-case basis. This database is maintained by the organization's systems management tool and forms the core configuration management database (CMDB).
- Web-based Reporting System:** Another function of the organization's systems management tool is to provide detailed information on the inventories that have been collected as well as provide consistency and compliance reports for all systems.

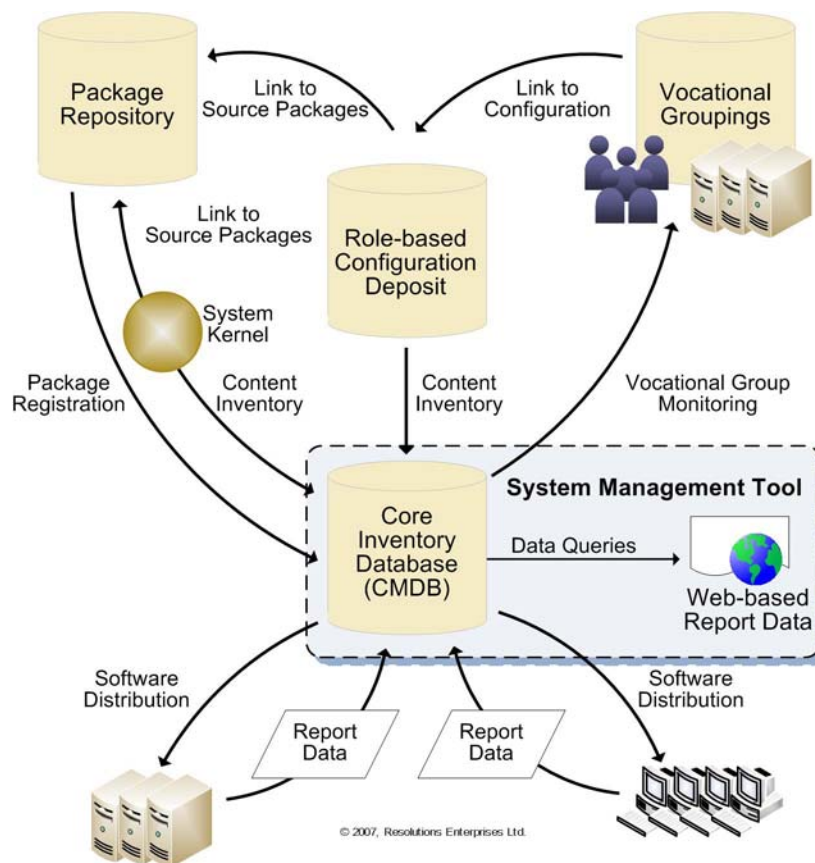


Figure 6.8. A Structured Inventory Management and Systems Management System

System staging and application distribution can be performed on the basis of the inventories your organization maintains. In addition, when systems change vocations (i.e., a user changes IT roles), the inventories are used to remove applications that are no longer required and add applications required by the new role-based configuration. This means that there must be a very tight relationship between the inventory process and the application distribution tools and mechanisms.

Locked Systems and the Standard User

For processes based on system construction and inventory management to work, organizations must ensure that software installations originate only from centralized and authorized sources. When employees work as a Standard User, they cannot install applications on their own.

Package Applications for Windows Installer

Because they need to be installed silently and in the background, especially during system reconstruction, applications must be packaged to automate the installation process. Yet, application packaging is often one of the most overlooked processes in IT. But, because it is the process that controls application installations, it is one of the most important processes in any application management strategy.


Few people in IT haven't performed an interactive software installation. The installation experience can range from good for an advanced user to very bad for a novice. The simplest installation will ask only a few basic questions but some of the more complex installations are enough to stymie even the most sophisticated PC technicians.

The best installation is the automated installation—the one where no user input is required. And the best of these is the customized version of the automated installation—the version that is designed to work within the specifications your organization outlines for its network.

In comes enterprise software packaging (ESP). ESP deals with the preparation of standard, structured automated installations for deployment within a specific organizational environment. These automated installations or packages take into consideration all of the installation requirements for the organization: organizational standards for software usage and desktop design, multiple languages, regional issues, and especially, application-related support issues. In addition, packaging should cover all applications including both commercial software and in-house applications.

There is a difference. Most commercial software products already include the ability to automate their installation. Unfortunately, there are no official standards in the plethora of Windows-based commercial software products for installation automation. Fortunately, this state of affairs is slowly changing with the advent of Windows Installer. With WIS, it is now possible to aim for a standard, consistent installation methodology for all software products within an organization. This standard approach is at the heart of Enterprise Software Packaging.

There are a whole series of different software packaging tools on the market. These tools are designed to assist IT personnel in the preparation of automated, interaction-free application installations. The focus of these tools is simple: allow IT personnel to perform an application installation and configuration, capture this customized installation and ideally, reproduce it successfully on every PC in the enterprise.

 Two of the most famous software packaging tools on the market are:
Altiris Wise Package Studio: <http://www.altiris.com/Products/Packaging.aspx> and
Macrovision AdminStudio:
<http://www.macrovision.com/products/adminstudio/adminstudio/index.shtml>.

But application packaging must be structured if it is to succeed. Standards must be set and maintained. Application packages must be documented and detailed in the same manner no matter who performs the packaging process. Quality assurance must be high; installations must work in every instance even if the configuration of destination computers varies. Package repositories must be maintained and updated according to organizational application acquisition policies. All of these activities are part and parcel of the ESP process.

In addition, there are several secondary reasons to perform packaging. One of the most important is application conflicts. Applications in a Windows environment are composed of several different components which are either private or public. Public components are shared between different applications. In fact, Windows Resource Protection was designed by Microsoft to help alleviate the transport and installation of public or shared components by applications because of its potential for disrupting system stability. This is why one of the strongest features of application packaging tools is conflict management—the ability to inventory all of the components in a system stack as well as all of the components in each and every application package. You then use this conflict management database to identify potential conflicts and circumvent potential issues before they occur. This is one more reason for a standards-based packaging approach (see Figure 6.9).

Packaging tools include support for this standards-based approach in the form of packaging templates and workflow templates. Packaging templates let you create Windows Installer templates that are applied with default settings each time you prepare a package for WIS integration. For example, if you decide to digitally sign every package in your network, you could insert a digital certificate within your template and have it automatically apply to every package you create. The preparation of these templates must be done with care before you begin the packaging process.



There are several reasons why the inclusion of digital certificates into application packages is a great idea. First, if your packages are digitally-signed and you sign any application patches with a similar signature, standard users will be able to install patches without administrative rights in Windows Vista. Second, if you digitally sign your packages, then you can control their use through Group Policy via the Software Restriction Policies Windows Server supports. These policies ensure only approved software is installed and runs in your network. You should look to both practices during application preparation. After all, the only thing you need is one single digital certificate which is quite easy to obtain.

Workflow templates control the process your packaging team uses during the creation of a package. If you design your templates before you begin packaging, then you can guarantee that every package will be created in the same way. For example, you can structure the process to follow the workflow example illustrated in Figure 6.9. This means you can assign junior packagers to most applications and rely on expert advice only for tricky applications which require more knowledge to work.



Packaging tools also include lots of guidance and some sample templates to work with. But, if you feel you need additional packaging expertise, rely on tools such as the **AppDeploy Library** from www.AppDeploy.com. This library includes the use of a tool called **Package Cleaner**. Package Cleaner will automatically scan your WIS packages and provide advice as to its contents, especially identifying items which can and should be removed. What makes this tool great is that it does not remove the contents of the package you select, it just marks them as non-installable. This way if you find that you needed a component you removed, you can simply run it through Package Cleaner to mark it for installation again.

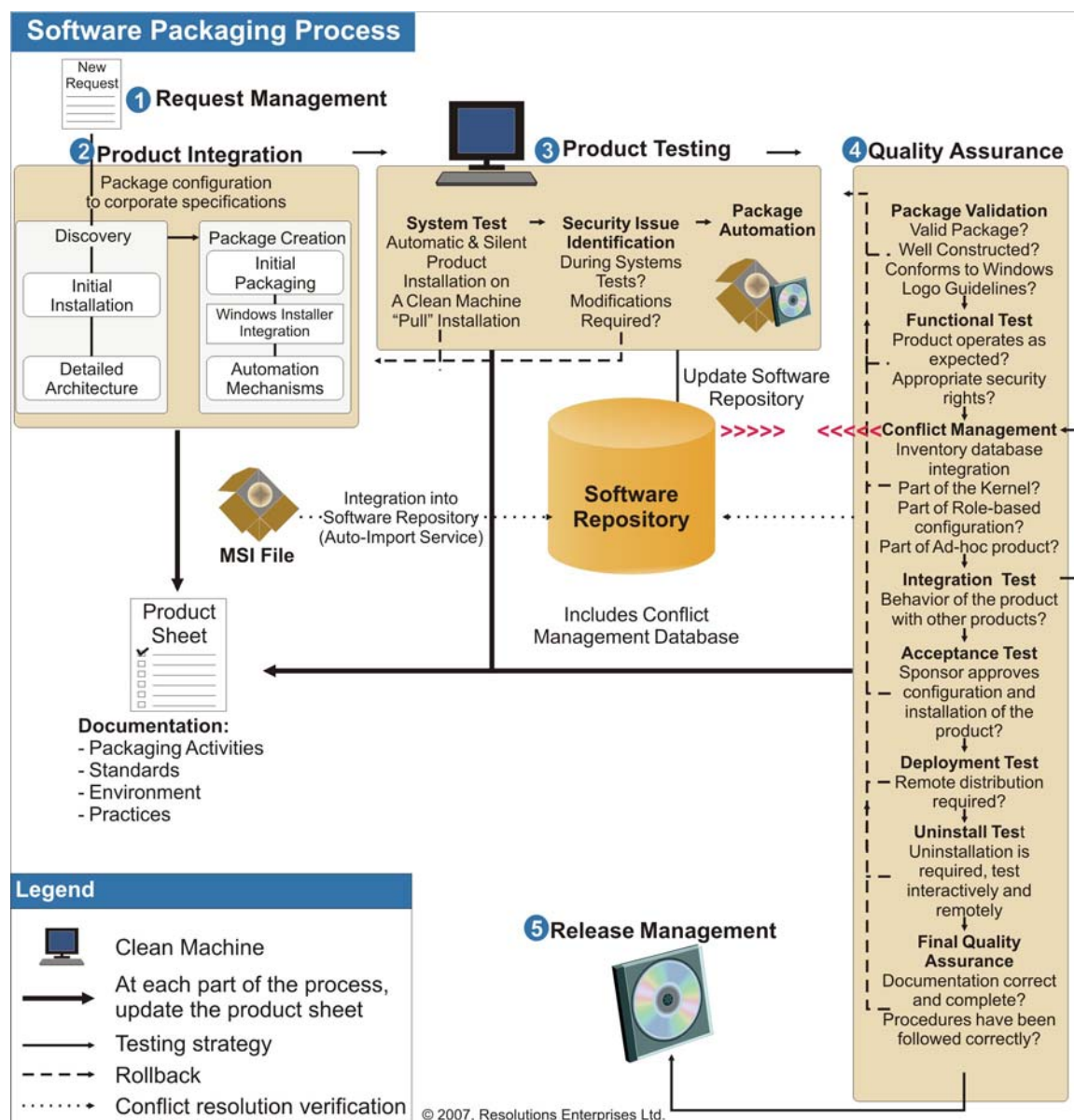


Figure 6.9. A Structured Packaging Approach


Packaging tools will also scale well. If your organization has multiple sites with technical staff in each site, you can easily set the tool up in each site and have its packaging database replicated to a central location. This allows distributed teams to work together to build the package repository.

A good source of information on packaging is the BDD 2007 Application Management guide: http://www.microsoft.com/technet/desktopdeployment/bdd/2007/AppMgmt_2.mspx.

Finally, to perform your packaging, you will need to categorize all applications—identifying which type each application falls into. Usually there will be three or four types in each organization:

- Native Windows Installer commercial applications—applications designed to work with WIS.
- Legacy commercial applications—applications that must be repackaged for WIS.
- Native Windows Installer in-house applications
- Legacy in-house applications.

Some organizations may further subdivide these categories to Win32, Win64 or .NET applications. Categorization will greatly facilitate the packaging activity because each application type requires a different approach. For example, you should never repackage a native Windows Installer application. Instead, you should create transforms which will modify the Windows Installer process to meet organizational standards. You will however, want to repackage all legacy applications to turn them into MSIs.

 For more information on application management and packaging in general, refer to <http://www.reso-net.com/articles.asp?m=8> and look up the **Application Lifecycle Management** section. Of special interest will be the **20 Commandments of Software Packaging**, a guide which has now become the leading authority in the industry, as well as **Enterprise Software Packaging, Practices, Benefits and Strategic Advantages** which offers a complete overview of application packaging.

Explore Application Virtualization Options


As you can see, application packaging and general application preparation is a lot of work and forms the bulk of the PC preparation activities. Now, wouldn't it be nice if you could vastly reduce this amount of work while ensuring that applications always work even when applications that are known to cause conflicts operate together on the same PC? Don't believe it? Well, welcome to application virtualization. Application virtualization tools isolate or abstract application components from the operating system and other applications yet provide full capabilities for application and operating system interaction. Think of it as Vista's file and registry virtualization capabilities on steroids.

In and of itself, application virtualization offers many benefits and may warrant an immediate implementation, but because of its nature, it requires redeployment of all of the applications you run in order to take full advantage of the virtualization capabilities. This is why it is ideal to adopt this technology during a migration project. Otherwise, you would have to replace all of the applications that are already deployed in your network—uninstalling the application and then redeploying it as a virtual application. If you haven't been working with Windows Installer packages, then uninstalling may leave behind traces of the application leaving the operating system in a potentially unstable state. This is why the best time to do this is when you are deploying a brand new, "clean" PC.

There are several different types of application virtualization technologies, but all of them basically produce a similar feature set:

- All applications are isolated from the operating system to the point where the application thinks it is interacting with the system, but when you look at the system directly, you see that no changes are made to system files, folders or registry keys.
- All applications are isolated from each other, letting you concurrently run applications that are known to cause issues and conflicts when on the same system. Manufacturing applications which are often never updated can run together. Different Microsoft Access versions can run on the same PC at the same time and even interact with each other through cut and paste and other system services. The potential is unlimited.
- Device drivers or hardware components cannot be virtualized and must be integrated directly to the system.
- Applications that interact at a low level in the operating system cannot be virtualized. For example, antivirus software should not be virtualized. Many also include Internet Explorer in this category, but several organizations have been able to properly virtualize IE and have it run on their systems.
- Virtualized applications are often structured for data streaming or if they are not by default, can be teamed with streaming technologies to provide the same effect. Just like a video that is streamed over the Internet, streamed applications are divided into small blocks of data that can start working as soon as enough content has been delivered to the system. The remainder of the content is then streamed in the background.
- Virtualized applications do not capture an installation, but rather an *installed state*. This means that the application need only be copied to the system for operation and does not need to run through an installation process as with applications packaged for Windows Installer. In fact, the crudest form of installation is the XCopy which just copies the application's files to a system.
- Users do not need elevated privileges to “install” a virtualized application. Since the application does not need to reside in folders protected by WRP, no elevated rights are required.
- Application virtualization platforms include the ability to integrate application access with specific user groups, groups that can be managed and maintained within a central directory such as Active Directory.
- Application virtualization follows Vista's operating model because when virtualized, applications only interact with the system in user mode.
- Applications that have been packaged in virtualized layers will work on both Windows XP and Windows Vista because the virtualization layer is in charge of translating systems calls and operations in the appropriate manner for interaction with the OS. In some cases, virtualized applications will work with *any* version of Windows from NT on.
- Application virtualization supports the ‘software as a service’ model since no installation logic is required to run a virtualized application on a PC. Applications are copied to systems in their running state and can therefore support an on-demand delivery strategy.

These reasons and especially, the last reason, make application virtualization very attractive. And since you are in the midst of a migration project to Windows Vista, then this is the ideal time to be looking to the adoption of application virtualization.

 As mentioned in Chapter 4, there are several versions of application virtualization. Make sure you review each product's technical feature set before you select the one you wish to go with.

Altiris offers **Software Virtualization Solution (SVS)** which is a filter driver that is installed on the OS. The filter driver manages the virtualization process. More information on SVS can be found at: <http://www.altiris.com/Products/SoftwareVirtualizationSolution.aspx>. SVS applications can be combined with AppStream's **AppStream 5.2** server to offer streaming capabilities. More information on AppStream and SVS can be found at: <http://www.appstream.com/products-application-virtualization.html>.

Citrix offers the integration of its Tarpon beta technology within its **Presentation Server** version 4.5. Tarpon uses a principle similar to Microsoft's SoftGrid and streams applications to desktops. Presentation Server 4.5 supports Windows XP application virtualization but not Vista. More information can be found at: <http://www.citrix.com/English/ps2/products/product.asp?contentID=186>.

Microsoft offers **SoftGrid** as part of the **Desktop Optimization Pack for Software Assurance (DOPSA)**. Microsoft acquired SoftGrid in mid-2006 and has since been reprogramming the SoftGrid client to get it to run with Windows Vista and x64. Support for Vista is slated for release in mid-2007 while support for x64 will be in 2008. More information on SoftGrid can be found at: <http://www.microsoft.com/windows/products/windowsvista/buyorupgrade/optimizeddesktop.mspx>.

Thinstall offers the **Thinstall Virtualization Suite (ThinstallVS)**. Thinstall incorporates its virtualization engine directly into the software package it creates. As such no pre-deployment preparation is required. More information on ThinstallVS can be found at: <http://www.thinstall.com/>.

At the time of this writing, the only two solutions that worked with Windows Vista were **Altiris SVS** and **ThinstallVS** and both worked very well. Keep this in mind when you choose the application virtualization solution you want to implement.

Pricing for each solution is relatively similar as some require direct acquisition costs and others are subscription based. In time, all costs become equivalent.

Do away with Application Conflicts

Despite Microsoft's best efforts, application conflicts still exist in Windows. This is partly due to the sheer number of applications organizations must run in order to support their operations.

While small organizations may get away with running small numbers of applications to operate, medium to large firms often find themselves running hundreds of different applications with all sorts of functionalities and features, each one requiring some specific component to run properly within Windows.

Using application virtualization, you can do away with conflicts once and for all and never have to worry about them again. Just package the application in the right format and then deliver it on an as needed basis. Unlike installed applications, virtualized applications do not embed themselves deep into the OS structure (see Figure 6.10). The virtualization layer protects the OS from any changes affected by applications.

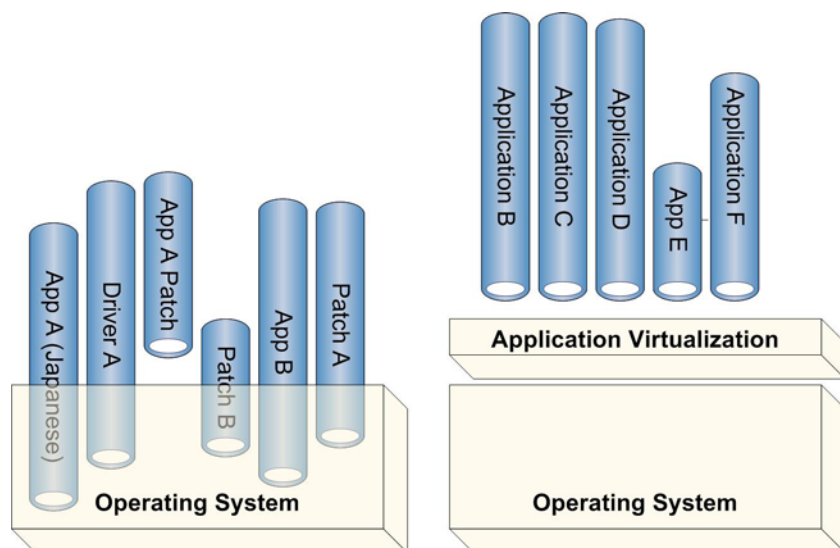


Figure 6.10. Application Virtualization protects operating systems from modifications

Consider your strategies. If you decide to include software virtualization into your deployment project, you will be able to obtain considerable savings in both time and effort. Virtualization reduces the time required to package applications since you no longer have to verify potential application conflicts. You still need to affect proper quality assurance on each package, but still, each package should take considerably less time to prepare. One of the most demanding processes for software packaging is the need to constantly return to a ‘clean’ system image to ensure the package is as clean as possible, but if you capture applications into virtualization layers, the machine is not affected at all. Therefore, no need to return to a pristine OS—the OS is always pristine. This alone will save considerable time and effort.

In addition, since application virtualization does not require significant infrastructure modifications, especially when the virtualization technology consists of either a driver or is included into the package itself, you can take advantage of it immediately. As soon as an application is packaged for virtualization, you can use it in either XP or Vista. Because of this, you might consider packaging your most troublesome applications immediately and deliver them to existing systems without waiting for the deployment project to update the OS. Several examples are available:

- **Access applications.** Most organizations cannot afford to upgrade the multitude of Microsoft Access applications their user community has developed. Properly converting these applications to a client-server structure, using back end databases and front end screens which can operate through the Access runtime is the best way to deal with this issue, but if you haven’t taken this step, then virtualize them! This will completely isolate them from the latest version of Access you need to deploy.

More information on running and managing Access applications in-house can be found at <http://www.reso-net.com/articles.asp?m=8> under **Decentralized Development Strategies**.

- **Custom in-house applications.** If you have custom applications that just won’t cohabitate with any other, you can virtualize them and have them finally cohabitate with any other on any system.

- **Custom industrial applications.** If you have custom industrial applications, for example, manufacturing applications, that require different settings for each manufacturing plant you run, you can now easily virtualize them to run them all on the same system.

This approach will let you get your feet wet with application virtualization and learn what advantages it truly brings to application management while you're waiting for the deployment project to complete.

Review your System Construction Strategy

When you're ready to integrate application virtualization with your operating system deployment, you might change the way you perform this deployment. For example, you might change the way you create your machine build. Before, organizations tended to create a massive system 'kernel' that included all of the most common applications and utilities found within the organization (remember Figure 6.7?). This 'fat' kernel is difficult to build and even more difficult to test because it requires the integration of vast numbers of components. In addition, using a fat kernel makes it more difficult to deploy because massive amounts of data must be sent to each machine. Using multicasting technologies reduces the time to deploy to each machine, but if there is a way to thin down the image, then why not take advantage of it? Finally, using a fat kernel means more work when it is time to update its core components. The sheer number of components means more updates more often.

Using application virtualization is the best argument for a 'thin' kernel. Application virtualization lets you create a new layer in the PASS system stack: the Generalized Layer. Each application in this layer is virtualized, but still distributed to each user in the organization. The kernel itself becomes much thinner because it no longer needs to include these applications (see Figure 6.11).

Your thin kernel is composed of the core operating system along with any required updates, adding core utilities such as antivirus, anti-spyware, firewalls, management agents, and virtualization agents if required. You still create a single core image that will include everything that is common to all desktops, but now, you can focus on the proper construction of your core operating system and expect it to maintain its pristine state for the duration of its existence within your organization. Just imagine the benefits!

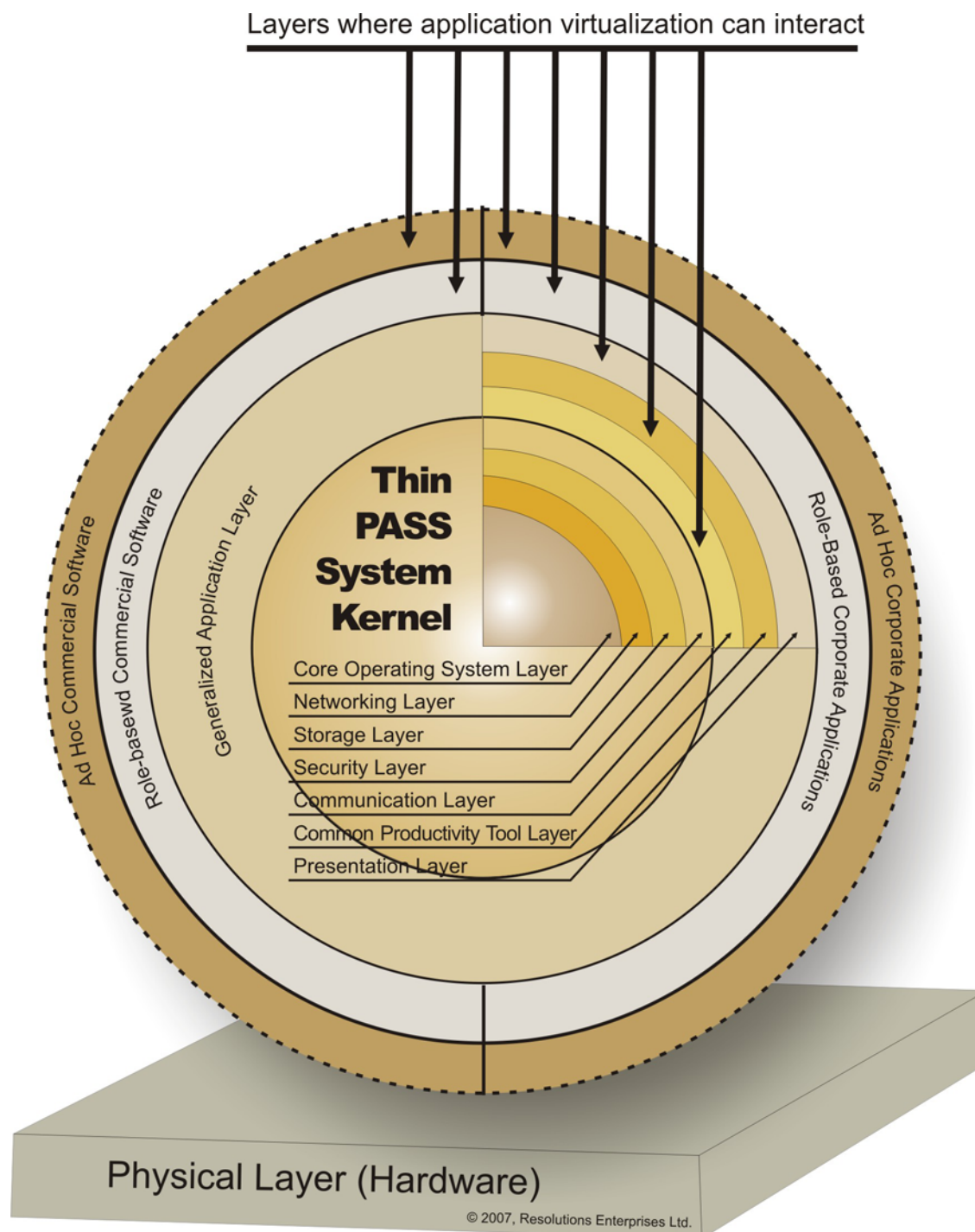


Figure 6.11. Application Virtualization affects several layers of the PASS model and supports a Thin Kernel

Reference computers are now much easier to construct. Very few applications or non-OS components are included in the kernel making it simpler to update, maintain and deploy.

In addition, many virtualization platforms let you convert MSI installations to virtual applications, saving you time and effort during your migration project. This is the case with Altiris' SVS since it integrates directly with Wise Package Studio.



Chapter 7 will focus on the creation of a Thin PASS Kernel. Software virtualization is here to stay and should be used by every organization that wants to reduce application management overhead costs. This is one of the core recommendations of this guide and the processes it recommends will reflect this recommendation.

Integrate Application Virtualization to the Migration Process


In addition to changing the way you construct systems, using application virtualization will also change the way you deploy systems, especially if you include streaming technologies. You can continue to rely on multicasting technologies to deploy the system kernel and then, you can immediately stream the applications in the generalized layer to each system as soon as the system is up and running. Because the applications are streamed, you don't need to worry about bandwidth issues as much. Users get the components they need to begin working immediately even if the streaming is not complete.

Then once the generalized layer is delivered, you can begin the deployment of the role-based layers if they are required. Again, it uses the streaming process so there is little impact on bandwidth. You can also use the same process for ad hoc applications.

Streaming technologies rely on Quality of Service (QoS) as a control mechanism to provide different priorities to users or data flows in routers and switches. Because of this, you may want to involve your networking group in preparation of the deployment to make sure you don't run into network bottlenecks.

Application virtualization and streaming running on thin kernel systems makes a lot of sense in the modern datacenter—more sense than using massive servers to offer desktop services to end users, mostly because to avoid a single point of failure, you'll always need more than one server to provide the service. Each endpoint has its own resources—CPU, RAM, hard disks and other peripherals. Streaming servers are nothing but file servers and do not require massive amounts of memory or processing capabilities unlike Terminal or Citrix Servers which in fact replace the processing power of the endpoint. Endpoints are also easier to manage since the only thing they contain is the thin kernel and can therefore be reimaged with little impact. After reimaging, the applications the user needs are streamed back onto the system. And, since applications are stored in a central repository and streamed from there to endpoints, you only have to update one single source when patching is required. They will automatically be re-streamed to each one of the endpoints that requires them.

Of course, you need more services in support of such a strategy. For example, if you want to be able to freely re-image PCs, then you need to make sure user data is protected at all times. The best way to do this with Vista is to use Folder Redirection Group Policy Objects (GPO). Folder Redirection automatically redirects key user folders such as Documents, Pictures, Application Data and more to server-based file shares. These file shares are configured with offline caching to ensure a copy of the data is always local to the PC, but since the original data is on the server, it is protected and backed up on a regular basis. This also means you need to fully support user data and to do so, you will need lots of backend storage, but disks are cheap today unlike massive processing servers. Another useful technology is the Distributed File System Namespace (DFSN). DFSN maps local shares to a global share name stored in the directory. Then, the directory redirects users to a share in the local site whenever they try to access the global share. And, to make sure the same content is in each local site, you can use DFS Replication (DFSR)—Windows Server 2003 R2's remote differential compression (RDC) replication engine—to keep each share in synch without breaking your WAN bandwidth. This makes it much simpler for users to access data throughout your network.

 More on the Folder Redirection, DFSN and DFSR strategy will be discussed in Chapter 7 as you build the system image and prepare the services required to support it.

In addition, streamed applications are cached locally for better operation. Most streaming solutions will let you control the duration of the cached application, letting you control license costs and making sure mobile users have access to applications even when they are not connected (see Figure 6.12). And, if you don't have a streaming solution, you can always place the virtualized applications on a network share configured for offline caching.

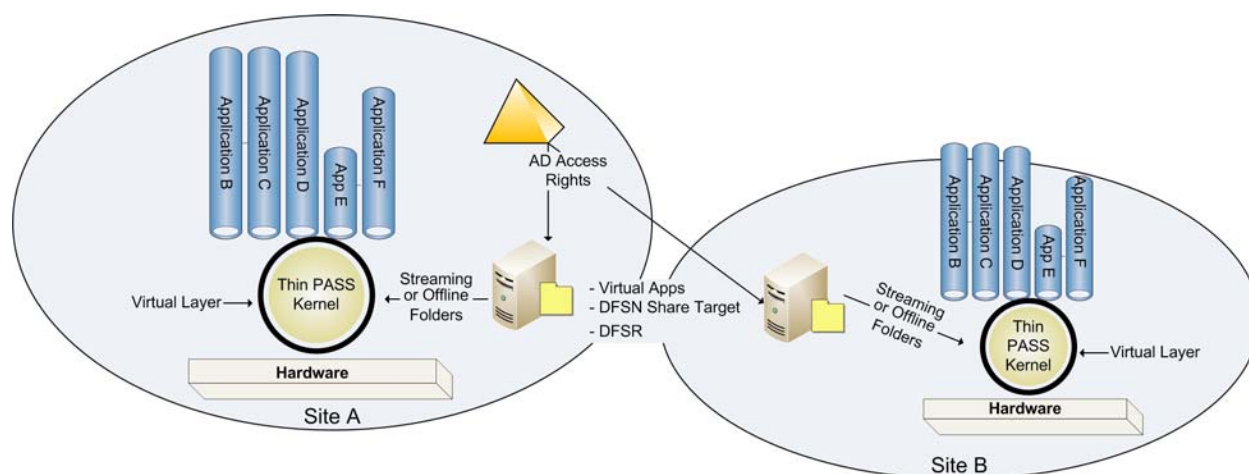


Figure 6.12. A Simple Application Virtualization and Streaming Architecture

Application virtualization may not apply to every single situation, but in many cases, it can be considered the Terminal Services 'killer' and do away with virtual desktop infrastructures (VDI) where you store virtualized OS images on massive servers and use them to provide services to users. In each scenario, the user requires access to an endpoint anyway. Why not make an intelligent choice and do away with the requirement for a massive server? Better yet, why not take the monies you would spend on monster servers and use it to pay for your application virtualization solution? Application virtualization finally lets you use the client-server model to its fullest. Don't miss this opportunity to upgrade your data center to the 21st Century!

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.