



## **Altiris® Software Virtualization Solution™ 2.0**

**White Paper**

January 31, 2006

## ABOUT ALTIRIS

Altiris, Inc. is a pioneer of IT lifecycle management software that allows IT organizations to easily manage desktops, notebooks, thin clients, handhelds, industry-standard servers, and heterogeneous software including Windows, Linux, and UNIX. Altiris automates and simplifies IT projects throughout the life of an asset to reduce the cost and complexity of management. Altiris client and mobile, server, and asset management solutions natively integrate via a common Web-based console and repository. For more information, visit [www.altiris.com](http://www.altiris.com).

### NOTICE

The content in this document represents the current view of Altiris as of the date of publication. Because Altiris responds continually to changing market conditions, this document should not be interpreted as a commitment on the part of Altiris. Altiris cannot guarantee the accuracy of any information presented after the date of publication.

Copyright © 2006, Altiris, Inc. All rights reserved.

Altiris, Inc.  
588 West 400 South  
Lindon, UT 84042

Phone: (801) 226-8500  
Fax: (801) 226-8506

BootWorks U.S. Patent No. 5,764,593.  
RapiDeploy U.S. Patent No. 6,144,992.

Altiris, BootWorks, Inventory Solution, PC Transplant, RapiDeploy, and RapidInstall are registered trademarks of Altiris, Inc. in the United States.

Carbon Copy is a registered trademark licensed to Altiris, Inc. in the United States and a registered trademark of Altiris, Inc. in other countries.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Other company names or products mentioned are or may be trademarks of their respective owners.

Information in this document is subject to change without notice. For the latest documentation, visit [www.altiris.com](http://www.altiris.com).

## CONTENTS

<b>Introduction.....</b>	<b>1</b>
<b>Technology Overview.....</b>	<b>2</b>
Install the Software Virtualization Agent	2
Create VSPs	2
Create VSAs	2
Deliver VSAs to Users' Workstations	3
Import VSPs on Users' Machines	3
Activate Layers	3
Run Applications Normally	3
Perform Application-Management Tasks	4
<b>Software Virtualization Agent .....</b>	<b>5</b>
Fslx.sys	5
Fsllib32.dll	6
<b>Layer Anatomy .....</b>	<b>7</b>
Registry Redirection Area	7
File Redirection Area	8
Redirection: It makes software virtualization possible	8
Layer Prioritization	9
<b>Configuration and Management Options .....</b>	<b>11</b>
Virtual Software Packages	11
Single program capture	11
Global capture	11
Data capture	12
Empty VSPs	12
Excludes	12
Virtual Software Package size	12
Using Other Management Interfaces	12
Future Management Options	13
<b>Conclusion .....</b>	<b>14</b>
<b>Glossary.....</b>	<b>15</b>



## INTRODUCTION

*By placing applications and data into managed units called Virtual Software Packages (VSPs), Software Virtualization Solution allows you to eliminate the intractable problems of application management.*

If your company is like most, someone on the payroll spends a lot of time managing applications—resolving application conflicts, repairing damaged applications, migrating to new versions of applications, or simply installing and patching applications. These tasks are not only frustrating but also time-consuming and, therefore, expensive. Fortunately, Altiris® Software Virtualization Solution™ 2.0 offers a new approach to managing applications—an approach that works regardless of whether the applications reside on stand-alone or networked computers. By placing applications and data into managed units called Virtual Software Packages (VSPs), Software Virtualization Solution allows you to eliminate the intractable problems of application management. With Software Virtualization Solution you can:

- **Activate and deactivate applications.** Software Virtualization Solution allows you to instantly turn on or turn off users' access to applications and data, which reduces the time it takes to provide users with the resources they need.
- **Recover damaged applications.** Software Virtualization Solution allows you to instantly reset broken applications to a known good state, without fear of damaging other applications. You can even use Altiris software management solutions to schedule regular application resets. In fact, you can use nearly any software management solution, including Microsoft Systems Management Server (SMS) and Novell ZENworks, to centrally manage virtualized applications.
- **Eliminate application conflicts.** Software Virtualization Solution ensures that each virtualized application has its own copy of DLL files that it would normally share with other applications. This eliminates conflicts that occur when two or more applications require different versions of the same DLL, thereby rescuing you from a condition that is commonly called "DLL hell."
- **Simplifies the process of migrating to new versions of applications.** Software Virtualization Solution allows even different versions of the same application to peacefully coexist. Among other things, this means you can keep older versions intact and available while you test new versions. Even after you have fully migrated to a new version, you can quickly roll back to the previous version any time.

Does this sound too good to be true? By revealing Software Virtualization Solution's underlying technical design, this white paper illustrates how Software Virtualization Solution can provide all of the benefits listed above.

## TECHNOLOGY OVERVIEW

This section provides a brief description of the steps you would take to set up and use Software Virtualization Solution in stand-alone mode.

### Install the Software Virtualization Agent

To use Software Virtualization Solution, you must install the Software Virtualization Agent on users' workstations and on a base machine. Software Virtualization Agent is a small client application that includes all of the functionality you need to deploy Software Virtualization Solution as a stand-alone implementation on users' machines. A base machine is a non-production workstation or server running a new, fully patched installation of the Windows operating system and little—or (preferably) nothing—else.

### Create VSPs

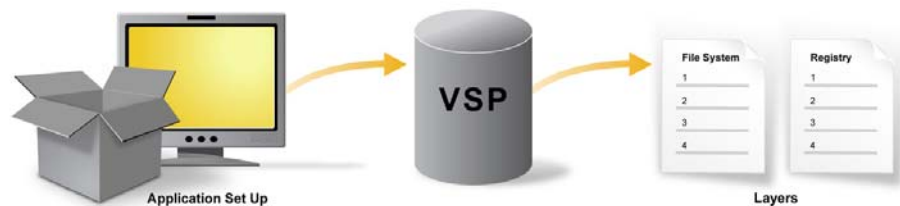
Software Virtualization Agent includes two management interfaces:

- SVSCmd.exe, a command-line interface
- Software Virtualization Admin, a GUI interface

You use one of these interfaces on the base machine to capture freshly installed applications into VSPs, which contain all of the files, processes, and settings required to successfully run these applications.

Figure 1

*Software Virtualization Solution captures applications in Virtual Software Packages (VSPs) and deploys the VSPs as layers.*



When you create VSPs on the base machine, Software Virtualization Solution automatically deploys them as layers—or virtualized applications. Each layer is a set of files and registry settings that contains everything the virtualized application needs to run successfully. As a system administrator, you use one of the Software Virtualization Solution management interfaces to edit the applications that reside in layers. For example, you would use Software Virtualization Admin to change an application's preference settings.

### Create VSAs

You use Software Virtualization Solution management interfaces to prepare layers for export to users' machines. To prepare a layer for

export, Software Virtualization Solution compresses its contents into a file called a Virtual Software Package Archive (VSA). Software Virtualization Solution does this by using the ZIP file format, which means you can use WinZip to view the contents of VSAs.

### **Deliver VSAs to Users' Workstations**

VSAs are the portable form of VSPs. You can use Altiris Notification Server or another software management solution to distribute VSAs to users' machines. Alternately, you can manually distribute VSAs using e-mail messages, CDs, or any other file-distribution medium. You can deploy VSAs on any machine running Software Virtualization Agent.

### **Import VSPs on Users' Machines**

After you distribute VSAs, you use one of the Software Virtualization Solution management interfaces to import them (as VSPs) on users' machines. When you import a VSP, Software Virtualization Solution automatically extracts its contents to a new layer.

### **Activate Layers**

When you, the system administrator, activate a layer (again, using a management interface), the application captured therein is available to users who access the machine. No setup or installation is required.

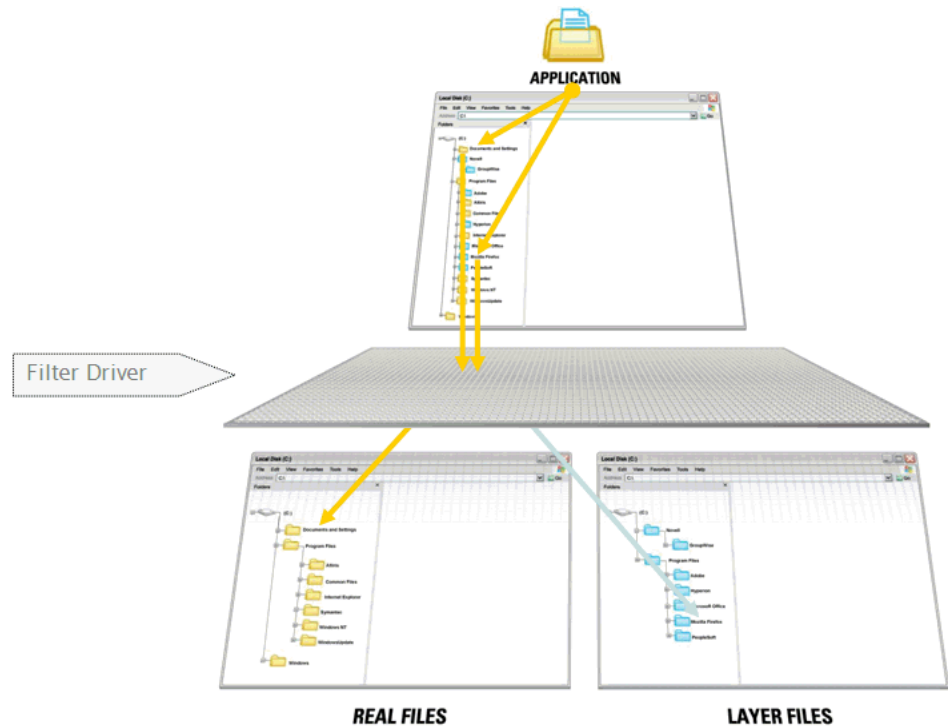
### **Run Applications Normally**

From the perspective of users and the Windows operating system, virtualized applications are indistinguishable from conventionally installed applications.

To make applications that reside in active layers available and indistinguishable from conventionally installed applications, Software Virtualization Solution redirects system calls for the applications' files, processes, and settings. When layers are not active, Software Virtualization Solution uses this same redirection technology to obscure the existence of the layers' contents—from users and from the Windows operating system.

**Figure 2**

*The file-system filter driver redirects application calls to the layers in which virtualized applications reside; it then presents back to the Windows operating system an aggregated view of files and data.*



After you've imported and activated virtualized applications, you can manage them as discrete units. For example, you can use a software-management solution such as Altiris® Notification Server™ to centrally update an application on all users' machines. Because the application that you are updating is virtualized, you never need to worry about inadvertently affecting other applications.

### **Perform Application-Management Tasks**

You can manually manage virtualized applications using SVSCmd.exe or Software Virtualization Admin. For example, you can use one of these management interfaces to reset a broken application, or to deactivate an application you no longer want to provide.

## SOFTWARE VIRTUALIZATION AGENT

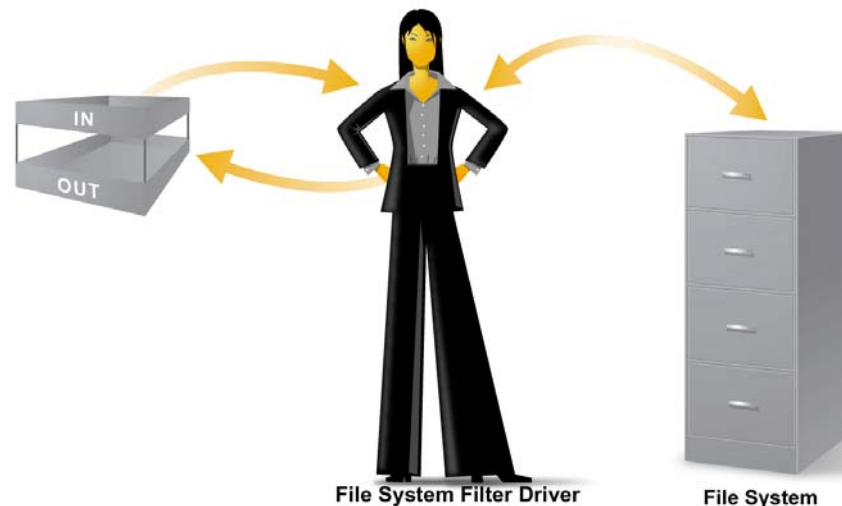
The core of the Software Virtualization Solution system is Software Virtualization Agent, a small—under 150 KB—client application. With this single application installed on users' workstations, you have a basic functioning Software Virtualization Solution system.

Software Virtualization Agent is actually not an agent in the traditional meaning of the word. We use the word “agent” in this application's name to maintain naming consistency with other Altiris client applications. Software Virtualization Agent includes:

- A file-system filter driver—Fslx.sys
- A DLL—Fsllib32.dll—that provides two application programming interfaces (APIs): a C API and a Windows Management Instrumentation (WMI) API

**Figure 3**

*In the Windows operating system architecture, the file-system filter driver is located between the input/output subsystem and the file subsystem.*



### **Fslx.sys**

As Figure 3 illustrates, in the Microsoft Windows technology stack, file-system filter drivers are logically located between the input/output (I/O) subsystem and the file system—that is, the File Allocation Table (FAT), New Technology File System (NTFS), or, eventually, Windows Future Storage (WinFS) system. From this vantage point, Fslx.sys and other file-system filter drivers, including anti-virus file-system filter drivers, see every request that comes through the computer's I/O subsystem. This unique position enables Fslx.sys to intercept and redirect system requests, a capability that allows it to virtualize applications and data.

In addition, Fslx.sys provides a small set of commands and layering, redirection, and prioritization technologies to help you manage VSPs. When VSPs become layers, only the Fslx.sys file-system filter driver can

see and access them during normal use. However, as a Software Virtualization Solution administrator, you can access and alter these layers using one of the Software Virtualization Solution management interfaces. For example, by using one of these interfaces, you can add an in-house application to an empty layer and can then export the layer to a VSA. An empty layer, as its name suggests, is a layer that does not contain files, settings, or processes.

### **Fsllib32.dll**

This DLL provides the main interface for accessing all of the functionality that is available on the Software Virtualization Solution system. All interfaces, including third-party interfaces that communicate through Software Virtualization Solution's C and WMI APIs, use Fsllib32.dll to communicate with Fslx.sys.

## LAYER ANATOMY

Layers have one of the following states, which determine how Fslx.sys handles requests from the Windows operating system.

State	Definition
Activated	Layer contents are visible to the system (and, therefore, to end users).
Deactivated	Layer contents are visible only to Fslx.sys.
Deleted	The layer is marked (using the ShouldDelete attribute) for deletion in the registry-redirect area. Layers are contextually deleted, which means Fslx.sys does not physically delete them until it can do so without affecting processes that are currently running.
Exported	Software Virtualization Solution has compressed the files and registry settings that comprise a layer into a VSA.
Imported	Layers are ready for activation. Software Virtualization Solution has extracted the contents of VSAs to the layers' redirect areas.

Each layer is composed of two sublayers: a read-only sublayer and a writeable sublayer. Only administrators can edit, update, and delete the contents of read-only sublayers, which Fslx.sys uses to reset applications to a known good state. (When you reset applications, Fslx.sys deletes the existing writeable sublayer and replaces it with new, empty writeable sublayer.) When users alter applications—by changing default preferences, for example—Fslx.sys writes these changes to the writeable sublayer. Users cannot—even unintentionally—make changes to the read-only sublayer.

Fslx.sys stores components of each layer (and its sublayers) in one of two redirect areas, areas to which it redirects system calls:

- It stores registry settings and attributes in a redirect area for the Windows registry (located in the registry at `HKEY_LOCAL_MACHINE\System\Altiris\Fsl`)
- It stores files in a redirect area for the Windows file system (located at the system root [usually the C drive]\Fslrdr).

To obscure these areas from end users, Fslx.sys uses redirection.

### Registry Redirection Area

In each sublayer's registry redirection area, Fslx.sys stores information about the state of the layer to which the sublayer belongs. It also stores a reference to the sublayer's file redirection area, reference counts, and subkeys that—through the Windows Service Control Manager—enable it to handle duplicate services running in multiple layers. In addition, Fslx.sys stores references to registry keys that contain user identity

information (HKEY\_CURRENT\_USER or HKEY\_USERS), data-layer specifications, a subkey for information about exclude entries, and a list of variables that govern the location of layer-specific files. As the name suggests, exclude entries define application files or processes that you want to exclude from layers.

### **File Redirection Area**

In the root of the file redirection area, Fslx.sys stores variable entries that abstract operating system-specific file locations so that you can deploy VSAs to computers running several types of Windows operating systems. When you load Fslx.sys on a user's computer, Fslx.sys determines (on the fly) values for user and system variables. Note that Fslx.sys supports several versions of the Windows operating system. For a list of these operating systems, see *Installing Software Virtualization Solution in Altiris Software Virtualization Solution 2.0 Reference Guide*.

Fslx.sys stores the files that comprise sublayers in numbered folders under the file-redirection-area root directory. Folder numbers correspond to sublayer numbers in the redirection area of the registry.

### **Redirection: It makes software virtualization possible**

Fslx.sys uses registry and file redirection areas to present to the Windows operating system—and by extension, applications and users on the system—an aggregate view of virtualized files and data.

For example, suppose a user launches Windows Explorer to view the contents of the Program Files folder. Further, suppose that the user's machine is running a virtualized application, such as Mozilla Firefox, and that the Firefox layer is active. Fslx.sys intercepts Explorer's calls to the file system. From the base, which comprises all files, settings, and processes that do not reside in layers on the user's machine, Fslx.sys gathers a list of non-virtualized applications that reside in the Program Files folder. It also redirects Explorer's calls to include Firefox, which would normally reside in the Program Files folder but which, in this case, resides in the file redirection area. Fslx.sys then responds to Explorer's calls with a complete list of folders that the system expects to see in the Program Files directory, including Firefox. In this example, if the Firefox layer were deactivated, Fslx.sys would have obscured its existence.

However, suppose that Explorer calls the file system to list the contents of the Windows directory and that two virtualized applications are currently running on the machine, each of which is using its own instantiation of a particular DLL file; for example, Xyz.dll. Which instantiation of the Xyz.dll file will the Fslx.sys file-system filter driver reveal to the system? The answer to this question hinges on which file resides in the layer with the lowest priority.

## Layer Prioritization

Because Fslx.sys must sometimes decide which instance of two identical DLL files to reveal to the system or which of two virtualized applications to open if users click a file that both applications can read, Fslx.sys assigns a priority to each layer. Specifically, for each system request, Fslx.sys:

1. Builds a list of active layers
2. Assigns each layer a priority value
3. Orders a list of active layers based on these priorities

By default, Fslx.sys assigns layers the following priority values:

Type of Request	Default Priority Value
Data layer	45.5
Normal owner	55.5
Base owner	65.5
Base	75.5
Normal	85.5
Normal owner	55.5
Normal	65.5
Base owner	75.5
Base	85.5

Fslx.sys bases priority assignments on the type of request it receives: either a normal request or an HKEY\_CLASSES\_ROOT request. (HKEY\_CLASSES\_ROOT requests are based on file type. For example, the system would make an HKEY\_CLASSES\_ROOT request if a user clicked on a text file.) If the request involves a process, Fslx.sys also takes into consideration the owner layer of the process. The owner layer of a process is the layer that houses the executable file that starts the process. The owner layer of a child process is the same as the owner

layer of its parent process, even if the executable file that starts the child process lies in a different layer.

The following example illustrates how Fslx.sys uses prioritization.

Suppose a user has on his machine two virtualized browser applications: Mozilla's Firefox and Opera Software's Opera. Further, suppose that the layers both applications occupy are currently active on the user's computer. When the user double-clicks an HTML file on his desktop:

1. Explorer calls the system registry to determine which program handles HTML files. This is an HKEY\_CLASSES\_ROOT request. Fslx.sys intercepts the call.
2. Fslx.sys considers the Firefox layer, the Opera layer, and the base. Because both Firefox and Opera are registered to open HTML files, Fslx.sys checks certain values in the registry redirection area. Specifically, it checks values in HKEY\_CURRENT\_USER and HKEY\_LOCAL\_MACHINE.
3. Based on these values, Fslx.sys assigns a priority of 65.5 to the Firefox layer, a priority of 65.4 to the Opera layer, and a priority of 85.5 to the base.
4. Fslx.sys begins its search for the files with which it will respond to the system's request. By beginning at the layer with the lowest numerical value (which represents the layer with the highest priority), Fslx.sys determines that it should use the following search order: Opera layer, Firefox layer, and the base. Fslx.sys performs these searches quickly, without creating noticeable system delays.
5. Fslx.sys directs the request to the Opera layer.
6. The system launches the Opera browser.

Although Fslx.sys assigns priorities automatically based on the conditions noted above, you can manually assign priorities.

## CONFIGURATION AND MANAGEMENT OPTIONS

Software Virtualization Solution provides a number of options for creating VSPs and managing them as layers. Through its management interfaces, you can create several types of VSPs. You can also perform basic management tasks, such as deleting, activating, deactivating, and editing layers. Software Virtualization Solution even provides information about licensing keys, allowing you to ensure that all active application layers are properly licensed.

### Virtual Software Packages

Software Virtualization Solution provides the following options for creating VSPs.

#### Single program capture

Using the single-program capture option, you specify the setup program for the application you want to capture. Software Virtualization Solution launches the setup program and captures all of the application's files and processes, including child processes and process-induced changes, in a single VSP. It also captures Microsoft Installer (MSI) and Service Control Manager changes. When the setup process is complete, Software Virtualization Solution automatically stops the capture process. As an administrator, you can then edit the VSP to include settings that you want to distribute to users. The resulting VSP contains everything you need—and only the things you need—to deploy the application virtually on users' machines.

#### Global capture

Using the global capture option, you manually launch an application setup program. Software Virtualization Solution then captures everything it does when you use the single-program capture option—and more. It captures all processes that make changes to the Windows registry and file system, regardless of the application generating these changes. In other words, it also captures background noise.

After Software Virtualization Solution has successfully captured the application in a VSP, you can alter the application by selecting new settings. You can also manually launch the setup program for another application. The global capture option enables you to capture several applications in a single VSP. Although we recommend using the single-program capture option to avoid capturing background noise, the global capture option can be useful. For example, you might use this option to capture two or more applications that require extensive configuration to work well with one another.

### **Data capture**

Using the data capture option, you can capture in VSPs collections of data, the types of which you define by specifying one or more of these data sources:

- File extension; for example, .doc, .pdf, or .fm
- Directory
- Subtree

Data VSPs collect data globally; that is, across the entire computer, including data from non-virtualized applications. In the future, Altiris plans to synchronize data layers with network storage.

### **Empty VSPs**

In addition to data and application VSPs, Software Virtualization Solution allows you to create VSPs that contain nothing at all. These empty VSPs are ideal vehicles for deploying in-house and other applications that don't include installation packages. Simply use one of Software Virtualization Solution's management interfaces to copy the application files into the empty layer.

### **Excludes**

Software Virtualization Solution also gives you the option to specify application elements that you don't want included in VSPs. For example, when you capture Microsoft Word, you may want to exclude Word document files. The system would then preserve these files if you needed to reset Word. As an alternative to using excludes, you can create data VSPs to capture files that you want to keep separate from application layers.

### **Virtual Software Package size**

Virtualized applications use slightly more disk space than do conventionally installed applications. This is because virtualized applications include their own copy of the DLL files that their conventionally installed counterparts share with the Windows operating system. This is what eliminates application conflicts. However, the amount of additional disk space that virtualized applications use is so negligible that we can safely say virtualized applications and data consume the same amount of disk space as would their non-virtualized counterparts.

### **Using Other Management Interfaces**

As mentioned earlier, you can use Notification Server and other third-party interfaces to distribute VSAs. If you are already an Altiris customer, you can use the Altiris Console, the management interface for

Notification Server and Altiris® Deployment Server™, to manage Software Virtualization Solution. Behind the scenes, the Altiris Console uses SVSCmd.exe commands and operations, allowing you to centrally manage the Software Virtualization Solution resources running on your network.

Alternately, you can use one of the growing number of third-party software management applications that support WMI; for example, IBM's Tivoli software management products. You can also create a C- or WMI-based interface for managing your Software Virtualization Solution system. You can find documentation for these two APIs in the Altiris software developer kit (SDK). The C API exposes all of the functionality available in Fslx.sys, and the WMI interface exposes approximately 80 percent of this functionality. You can also download through the Altiris® Juice™ website useful utilities such as Trinket. Juice is the Software Virtualization Solution user community. For more information, visit <http://www.altiris.com/juice/>. Trinket is a user self-service tool that resides in the Windows system tray. With Trinket running on their machines, your trusted power users can reset broken applications without a network connection and without help-desk assistance. Trinket also enables these users to activate, deactivate, and delete applications.

### **Future Management Options**

In future releases of Software Virtualization Solution, Altiris plans to use Wise Package Studio® to enhance the editing, package comparison, and pre-deployment testing features currently available through Software Virtualization Admin. You will also be able to manage Software Virtualization Solution through Altiris® Real-Time System Manager Solution™, which gives you alternate ways of viewing the resources on your system.

## CONCLUSION

By enabling you to capture applications and data as units—and manage them as units—Software Virtualization Solution eliminates many of the costly and obtrusive aspects of provisioning and deprovisioning network resources. This approach to software management also mitigates possible damage from sources inside and outside of your network. If users or malware, such as viruses and Trojans, damage applications, you can simply reset the damaged applications to a known good state. Furthermore, you will soon be able to use Altiris® Protect™ to protect the operating system and applications included in the baseline image. Protect provides users in multi-user environments with a consistent computing experience, regardless of which PC they are using. With Software Virtualization Solution and Protect working together, you can protect users' machines from virtually any kind of user error or software malfunction.

You can purchase and use Software Virtualization Solution as a stand-alone product. For a 120-day trial, register at the Altiris download site. You can also use Software Virtualization Solution with Notification Server or Deployment Server to centrally manage network resources. In addition, you can purchase Software Virtualization Solution as part of Altiris® Client Management Suite™ Level 2 for a fully integrated management solution. For more information, visit <http://www.altiris.com/>.

## GLOSSARY

**Base machine**—A non-production workstation or server running a fresh installation of the Windows operating system.

**Base**—The group of files, settings, and processes that are not included in Software Virtualization Solution layers. For example, the Windows operating system is part of the base.

**Dynamic-link library (DLL)**—A collective name for a group—or library—of shared files that can contain code, data, and read-only data. Several applications can share a single DLL file. However, application conflicts often result when applications require different versions of the same DLL. This condition is commonly called “DLL hell.”

**Empty layer**—A layer that does not contain files, settings, or processes.

**File-system filter driver**—A driver that sits between the input/output subsystem and the file system in the Windows technology architecture.

**Layer**—A set of files and registry settings that reside in Software Virtualization Solution redirect areas. Application layers contain everything the applications captured therein need to run successfully.

**Owner layer (of a process)**—A layer that houses the executable file that starts the process.

**Redirection**—A technology that enables Software Virtualization Solution to intercept system calls and redirect them to one of its two redirect areas.

**Redirect area**—An obscured location in the Windows file or registry system that holds the contents of Software Virtualization Solution layers.

**Registry**—A database that stores settings and options for the Windows operating system and for applications running on the Windows operating system.

**Sublayer**—A layer component. Each layer contains two sublayers: a read-only sublayer and a writable sublayer.

**Virtual Software Package (VSP)**—A collection of files, processes, and settings that comprise captured data and applications. Application VSPs contain all of the files, processes, and settings required to successfully run captured applications.

**Virtual Software Package Archive (VSA)**—A compressed file that contains all of the files and settings that constitute a layer. VSAs are the portable form of Virtual Software Packages (VSPs).

**Windows Management Instrumentation (WMI)**—A set of environment-independent specifications that allow software management applications to monitor and control components of the Windows operating system.