

Software Deployment Done Right!
– *Combine the right tools and processes to gain complete control of software deployment through Active Directory Group Policy in your network –*

Nelson Ruest & Danielle Ruest

A Report by Resolutions Enterprises



Sponsored by



Abstract

Most system administrators have discovered that when it comes to software management in their network, they're often faced with having to design, implement and deploy completely new system infrastructures, even if they have taken the time, sometimes, considerable time to deploy Active Directory. That's because despite the fact that AD provides a system management infrastructure through Group Policy, most software delivery systems rely on completely different technologies for management. This is not the case for all software management tools. This paper outlines a strategy that supports a completely automated software purposing system that can help ensure your organization is always in compliance in terms of software licensing. This is achieved through the selection of the proper tools for software management and the implementation of a simple software deployment strategy.

In short, this white paper aims to answer questions in regards to reuse of the Active Directory infrastructure for enterprise software deployment and make sure that organizations learn how to gain the full benefits automated software deployment can bring.

About the Authors

Danielle Ruest and Nelson Ruest are IT professionals specializing in systems administration, migration planning, software management and architecture design. They are authors of multiple books, notably two books published by McGraw-Hill Osborne, "Windows Server 2003: Best Practices for Enterprise Deployments", ISBN 0-07-222343-X and "Windows Server 2003 Pocket Administrator", ISBN 0-07-222977-2 as well as "Preparing for .NET Enterprise Technologies", published by Addison Wesley, ISBN 0-201-73487-7. They have extensive experience in software packaging and managing large packaging projects.




Table of Contents

| | |
|--|----|
| Introduction | 1 |
| Software Management through Group Policy | 2 |
| Use a Logical Model for System Construction | 3 |
| Package for Windows Installer | 5 |
| How to use AD Group Policy for Software Deployment..... | 8 |
| Reuse Your Active Directory Structure..... | 8 |
| Work with Group Policy | 9 |
| Implement a DFS Structure for Software Delivery Points..... | 13 |
| Deliver the Right Software to the Right Target | 16 |
| Using the Automated Software Management Model | 17 |
| Best Practices for a Simplified Software Deployment Model..... | 19 |



Power Your Active Directory investment


*Take Group Policy
based Systems
Management
to the next
level*




Specops Deploy™
*Group Policy based
Software Deployment*



Specops Inventory™
*Group Policy based
Asset Management*



Specops Password Policy™
*Active Directory
Security enhanced*



Active Directory Janitor™
*Keeps your Active
Directory clean*

Introduction

Software management in an enterprise deals with a lot of activities which include, but are not limited to inventory, lifecycle management, usage metering, and support. When asked point blank, “How many copies of Brand X software do you have in your network?” few organizations today can answer immediately. Most must search their acquisition records, look in partial inventories, calculate how many PCs they have in the network and, eventually, give you nothing more than their best guess. This begs the question as to why it’s so difficult for organizations to know exactly what software is on their network. Given all the management technologies available for software administration support today, it should be simple for organizations to gain complete and constant control over the software content of their networks. This control is important if only to be able to ensure that the organization conforms to its legal engagements in terms of software licensing and to facilitate long-term software inventory management.

Do you know how many copies of each software product you have in your network today? If not, why not?

Part of the answer lies with the way organizations manage the *software lifecycle* (see Figure 1). This includes evaluations, acquisitions, software installation packaging, remote installations, software upgrades, service pack and patch deliveries, maintenance fixes, and, most importantly, but most often forgotten, software *removal*. Managing the software lifecycle means managing the entire software process from the moment you make the acquisition decision to the moment you retire the software product from your network. It’s in the latter portion of this process that organizations tend to have the most difficulty. That’s because most rarely take the time to push the software lifecycle management process to its fullest and manage the complete lifecycle.

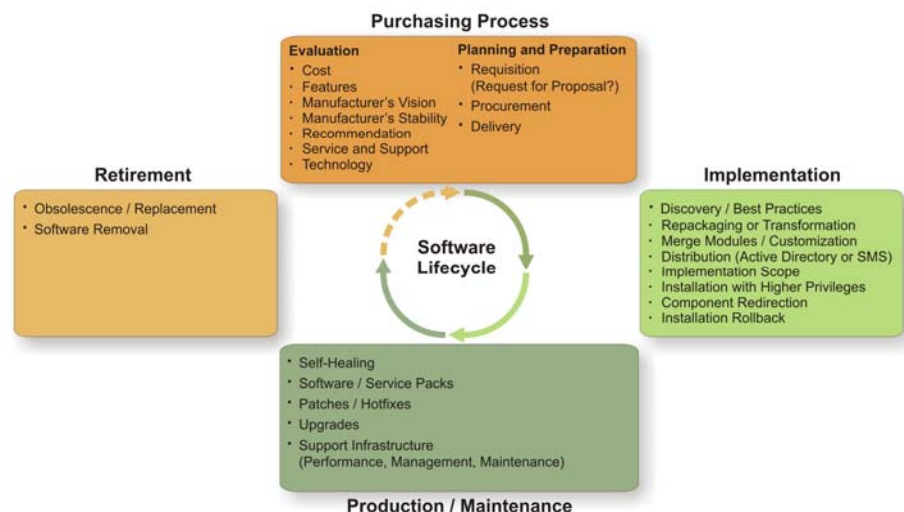


Figure 1. The Software Lifecycle

Take the case of PC repurposing: Employee A is given a PC with all the software products required for her work. Later, Employee A is moved to another position within the organization, and her PC is reassigned to Employee B. Employee B has a different role in the enterprise, which

requires he have different software. In most cases, while the IT department might install the necessary software products on Employee B's computer, they don't take the time to *remove* the software products Employee B doesn't require.

In many ways, this situation is understandable. Software removal is traditionally a risky task at best. Due to the nature of Windows, software products often share components. Removing software from a PC might also remove shared components, endangering the stability of the PC. Many IT departments chose to avoid the risk and leave the product on the PC.

This methodology, however, presents several problems. First, it puts the organization at risk legally. Each enterprise has a legal obligation to ensure it follows the software license agreements for each product in its network. Because most products in the Windows world are licensed according to installation and not concurrent number of users, each installation means one license paid. Not removing a software product when it's no longer used should cost the organization an additional license. Second, leaving the product on a PC leads to eventual inventory chaos. If you use this approach each time an employee moves from one place to another, you end up with licenses left on computers here and there. Then, when it's time to migrate to new operating systems, you'll have to invest a lot of resources in re-inventorying the network. It's that or invest in software installations that aren't really required by the PC's user.

There are several ways to address this problem, but the best is to have a structured software management strategy. This strategy must be made up of several elements which include:

- Software management policies including usage and removal policies
- Software removal policies
- Proper software lifecycle management tools such as software deployment and software inventory
- A system construction model
- A role-based software deployment strategy
- A software management team including members from IT and Finance as well as end users as subject matter experts

Putting all of these elements in place will go a long way towards simplifying software management in your organization. One of the first places to start, often because it is one of the easiest to perform is the selection of the right tools. Let's start with software management.

Software Management through Group Policy

When Microsoft invented Active Directory (AD) over 5 years ago, they built software deployment right into the product. It was part of the IntelliMirror strategy Microsoft wanted to build into Windows 2000—a strategy that was designed to provide complete support for end user provisioning including operating system deployment, software

deployment, user settings management and user document protection. Software deployment in particular is especially well suited to Active Directory since the base structure of the directory contains all of the required information such as user accounts, computer accounts, site and subnet information, automated replication between sites and most importantly Group Policy Objects (GPO).

Group Policy is the core management engine in AD. It can control everything from setting preferences in Internet Explorer to assigning login scripts, controlling how applications function, assigning security settings and much more. There are now well over 1,000 different computer and user settings that can be controlled through Group Policy with more being added each time Microsoft releases a new version of the operating system or even just a new service pack for the operating system. In addition, third-party manufacturers are adding their own extensions to Group Policy letting organizations use this central engine for the management of application compatibility, patch and software update deployments, inventory collections and software delivery. The major advantage of using Group Policy for systems and software management is that you use one single interface and one single engine to manage all aspects of your AD network.

But, instead of building upon the IntelliMirror model, Microsoft focused on enhancing and upgrading Systems Management Server (SMS), leaving AD's software deployment capabilities pretty much as is in Windows Server 2003. This makes software delivery through Group Policy the weakest point of the Group Policy management infrastructure. Microsoft's approach was that organizations could use AD's software deployment capability if they wanted to but they would lack special features such as delivery reporting, legacy software delivery, bandwidth control and delivery server control. If organizations want all of these features, Microsoft still recommends the installation and use of SMS.

This strategy leads to several problems. While SMS is a powerful software management tool, it is based on older code that does not integrate well with Active Directory. In addition, when you deploy AD, you go through all the work it takes to set up an Active Directory architecture, place domain controllers (DC) strategically throughout your network, make sure that replication of data is working properly between all the DCs, create your AD object management structure through organizational units (OU), design your Group Policy strategy and so on. Then you have to start all over again with the SMS architecture determining where to position the various servers required by SMS; in fact, duplicating much of the same infrastructure costs and systems you need for AD to function properly. Wouldn't it make more sense to rely on the existing AD infrastructure to manage software along with everything else? That would simplify software management considerably.

Use a Logical Model for System Construction

Another great way to simplify enterprise software management is to use an architectural model for system design. One such model is the

Point of Access for Secure Services¹ (PASS) (see Figure 2). This model is based on the construction of a computer system that responds to corporate needs in four ways:

- The **PASS system “kernel”** is designed to meet the needs of the average corporate user. It contains all the software components required to perform basic office automation and collaboration tasks. In addition, it’s divided into a series of layers similar to the Open System Interconnect (OSI) Networking Model. In fact, 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.
- **Role-based applications and software** are added on top of the kernel to meet the requirements of special information technology roles everyone plays within the organization. Applications are in-house programs that meet mission-critical or mission-support functions. Software is identified as commercial software products that aren’t required by all personnel, but only for specific roles or tasks.
- Finally, an **ad-hoc layer** responds to highly specialized IT requirements often expressed on an individual basis. This ad-hoc layer can be applied to software or applications.
- In addition, the PASS model is based on a **standard hardware** deployed throughout the organization. Standard hardware is often hard to achieve, even from the same manufacturer, but what is important at this level is to ensure all machines use the same hardware abstraction layer (HAL) so that they can all use the same disk image during deployment.

Using Disk Images

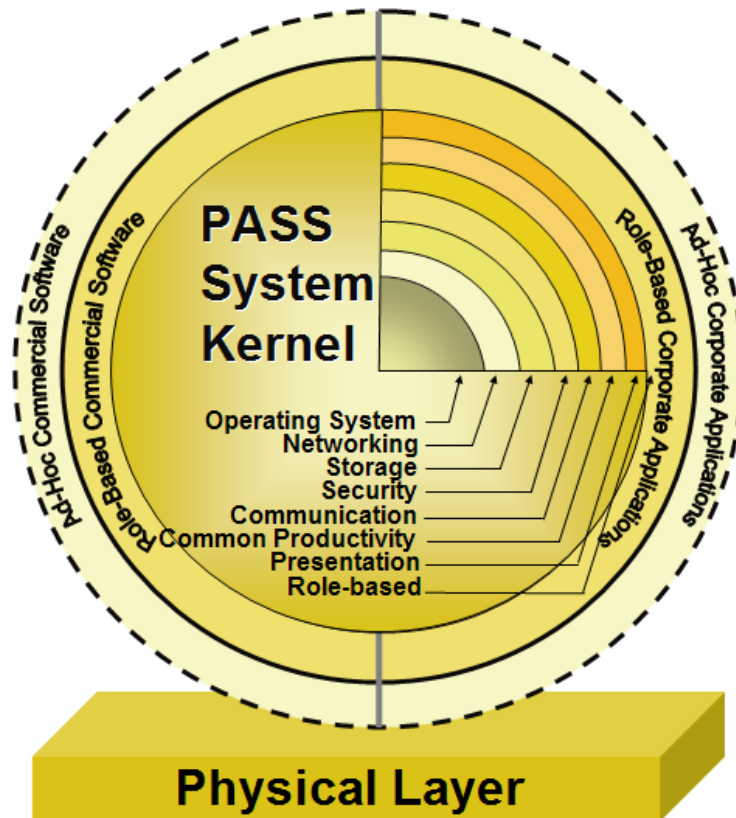
Disk images must use the SysPrep deployment tool Microsoft makes available for Windows deployments. SysPrep ensures the same image can be installed on different machines. But, in order to use the same image on multiple machines, each machine must use the same HAL.

Constructing systems based on a model such as this reduces system and software management efforts because it diminishes the number of programs that must coexist on any system. A good portion of systems—sometimes up to 50 percent—will require only the system kernel. Remember that the kernel should contain every program that is royalty-free and required by the entire organization (such as Adobe System’s Acrobat Reader or the Microsoft Reader) or every program the organization obtains an enterprise-wide license for (such as Microsoft Office).

Also, by grouping programs into role-based configurations, organizations can reduce the number of programs that must coexist on a system. Role-based configurations include every program required by every member of the IT role grouping. For example, Web editors require a Web editing tool, a graphics tool, a Web-based animation tool, and other Web-specific utilities. You can package this group of tools separately, but it should be delivered as a single unit on all systems belonging to the IT role. Role-based configurations often include no more than 5 to 20 individual programs, depending on the role.

¹ *The PASS Model was first introduced as the SPA Object Model in Preparing for .NET Enterprise Technologies by Ruest and Ruest (see Resources).*

Ad-hoc programs reduce system management efforts even further because they are required by few users in the organization. They are still packaged to enable centralized distribution and automated installation in order to enable remote delivery.



© 2005, Resolutions

Figure 2. Consider Using the PASS Model. This system construction model is based on three major components: a system kernel that provides basic functionalities to the entire enterprise, role-based software and application configurations that provide advanced functionalities to specific groups of users, and ad-hoc components that target specific users. The entire model relies on standardized hardware for all users.

Package for Windows Installer

In addition to the use of a system construction model and proper software deployment tools, organizations should consider making use of the Windows Installer Service (WIS) Microsoft has built into the Windows platform. This service is specifically designed to simplify software installations as a whole. It provides a wealth of features that simplify system administrator's tasks when it comes to software management. The most famous of these is self-healing; that is, automatically correcting installation or configuration errors for applications as they are launched by users. That's because Windows Installer creates a consistency database on the local system during the

installation of the software product. This database is checked each time software is launched. If errors are detected, WIS connects to the original installation source and corrects it. This is not the only feature WIS offers—it can install software with elevated privileges in locked-down environments, it automatically interacts with Group Policy Software Installation (GPSI), it can be used to patch products as they evolve, and it supports clean removal of software from the system. In fact, WIS provides comprehensive support for the software lifecycle (see Figure 3).

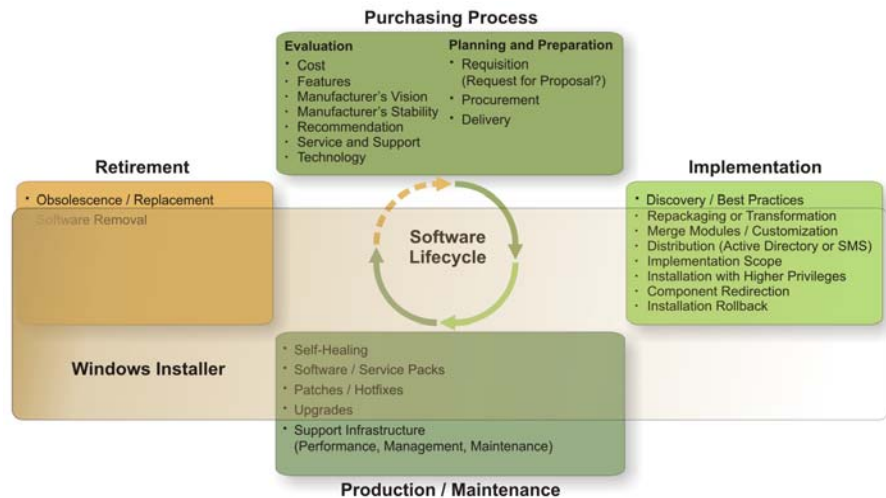


Figure 3. Windows Installer and the Software Lifecycle. WIS covers many of the activities included in the software lifecycle.

The Windows Installer consistency database can also perform clean software removals. If you have experience with software removal prior to Windows 2000 (or older systems with the Windows Installer service installed), you know the concept of a clean uninstall is a myth. Fortunately the myth becomes reality with Windows Installer-enabled software. One of Windows Installer's main functions is to manage software conflicts and ensure shared system components aren't damaged by other software installations. So, if conflicting components are added during a software product installation, Windows Installer ensures these components are installed in a manner that avoids potential conflicts automatically. Although simplistic, the definition of this function should help you understand that any application installed through Windows Installer will cleanly uninstall because its components are isolated by this service. Therefore, uninstalling software has little or no impact on the rest of a computer system.

Given the major advantages you can gain from integrating the installation of software products to the Windows Installer service, you should seriously consider migrating all your software programs and applications to versions integrated with this service. Of course, most corporations won't be able to achieve this through upgrades for several reasons. First, some programs, especially internally-developed programs might not be upgraded easily. Second, the average corporation (with more than 1,000 users) has about 300 different software applications in its network. Upgrading all these products is

cost-prohibitive and often unnecessary. Third, some applications simply don't offer new versions. Finally, some manufacturers, unfortunately, still don't integrate their software products to the Windows Installer service though these are becoming fewer and fewer.

In most cases, you'll have to consider repackaging software installations yourself to take full advantage of the many features of the Windows Installer service. Several third-party tools are available on the market for this repackaging process. You can also take advantage of this opportunity to implement a model such as the Point of Access for Secure Services (PASS model) and categorize your software products during this software repackaging or transformation process. This undertaking requires considerable effort, but it's an essential step toward software lifecycle management. It also provides excellent return on investment, because many of the software installation issues you live with today should be eliminated. In addition, repackaging all your software and applications in Windows Installer format will simplify your software delivery mechanisms because all products will use the same installation format.

How to use AD Group Policy for Software Deployment

As discussed earlier, there are several advantages in using Active Directory to manage software deployment.

- The first is that you can reuse your existing Active Directory infrastructure.
- The second is that you continue to manage your infrastructure with one simple tool: Group Policy.
- The third is that you can simply add software distribution points to facilitate software delivery without having to rethink your entire network.
- The fourth is that with Group Policy, you can properly target software delivery *and* make sure it is *automatically* removed from systems when it is no longer required.
- The fifth is that your infrastructure can continue to grow without requiring changes to your software deployment strategy.

These advantages make a strong case for software deployment through Group Policy. But it doesn't all happen by itself. Let's see why.

Reuse Your Active Directory Structure

Reusing Active Directory

Any system management tool that does not integrate directly to Active Directory requires the deployment of a secondary infrastructure for the purpose of software management.

Deploying Active Directory requires a lot of forethought and planning since it will form the underlying infrastructure for the management of objects within your network, but also provide the central point of management for all authentication and authorization in the organization. Because of this, you need to take the time to plan the directory properly before you deploy it. In some cases, this endeavor seems so daunting that some organizations have still not managed to deploy AD yet. Well, it's not that hard. In fact, when you follow the proper guidance, setting up an Active Directory² that will last you for the coming years is relatively simple. That's not to say that it doesn't involve a fair amount of work, but when done right, the benefits are quite valuable.

Windows Server 2003 "R2"

One great advantage of implementing version R2 of Windows Server 2003 is the new File Replication Service which now offers byte-level file delta replication, greatly reducing replication overhead.

When you use AD to deploy software, there are no changes required to your directory structure. You can target software in any way that you can target Group Policy (see Working with Group Policy below) and since the domain controllers in an AD are able to replicate content to each other, all software deployments will be properly replicated to each site in your network. To reduce the amount of replication managed by the directory and to avoid software installations performed over the wide area network (WAN), you should also implement a distributed file system infrastructure in support of the software distribution points you will use to deploy software.

² For complete guidance in setting up your Active Directory please download Chapter 3: *Designing the Active Directory of Windows Server 2003, Best Practices for Enterprise Deployments* by Ruest and Ruest for free at <http://www.reso-net.com/livre.asp?p=main&b=winsvr&m=12>.

Work with Group Policy

Windows Server 2003 includes a set of Group Policy objects that can be used to deliver software to both users and computers. These GPOs are closely tied to the Windows Installer Service which is available for both PCs and servers. Group Policy Software Installation can also deploy applications that are not integrated to WIS but they have to be wrapped into special “ZAP” format files to work properly. This makes it less than practical to deploy these legacy applications through the default AD capabilities.

In addition, software can be targeted to either users or computers. Active Directory can also either publish or assign software. When computers are targeted, software is assigned; this means that the software is completely installed on the target system. Of course, this requires elevated rights in locked down environments—environments where users do not have installation rights—but this is a feature of WIS. When users are targeted, you can either assign or publish software. Publishing software does not install it on the target system; instead, it displays the software shortcut in the Windows Start Menu. When the user first clicks on the shortcut, the Group Policy installs the software.

One of the most important decisions you will make in terms of software management will be to determine whether you will deploy to users or computers. It is highly recommended to deploy to computers only, especially if you use installations integrated to WIS. The reason for this is the way WIS works. When installations are sent to users, there may be a negative experience for them and the enterprise. First, software is installed on each system they log on to, multiplying the licensing issue, unless software is uninstalled each time they log off. This strategy would also consume a lot of bandwidth. Users would also be confused by the Windows Installer launching each time they log into a new machine. But the worst impact deploying software on a per user basis causes is when patching software.

As discussed earlier, WIS supports software maintenance. This means that it will automatically patch software that has been installed through WIS. The issue is that when software is installed at the computer level, it only requires one single patch. When software is installed on a per user basis, you have to patch each per user installation on each machine. In environments where multiple users share machines, this can become quite prohibitive. **The recommendation:** install software on a per computer basis.

Despite its inherent capabilities, the default Group Policy Software Installation also lacks features comprehensive software deployment tools can provide. These include:

- **Delivery guarantee** — To guarantee that a software installation has occurred before a given time. GPSI does not provide delivery reporting. Software is either installed or is not and can only be verified by looking on the target computer.
- **Scheduling** — To control delivery times for non-working hours. GPSI does not offer scheduling. Software is delivered as soon as it is deployed.

Specops Deploy

Specops Deploy boosts GPSI by providing all of these features, except for inventory reporting, to software delivery through Active Directory. In addition, providing your AD is properly designed, it requires no changes to AD at any level—no schema modifications, no replication changes, no structural changes.

Specops Inventory

Special Operations Software will release a complete Active Directory-based system inventory solution by the end of 2005. This tool will report on hardware, operating system, services and drivers, software and more, adding full software management functionality to the current Specops Deploy tool.

- **Bandwidth control** — To control bandwidth usage and compress data when sent over the WAN. GPSI does not control bandwidth.
- **Inventory** — To ascertain that target systems have the required resources to install and operate the software and to keep abreast of where software has been installed. You can create Windows Management Instrumentation (WMI) filters that identify if the target machines have enough disk space or meet other conditions prior to delivery, but it will not tell you which of the target machines actually matched these requirements and installed the software.
- **Status** — To be able to determine the status of a software delivery job across the WAN to multiple geographic locations. GPSI does not report on this.
- **Reporting** — To be able to generate comprehensive activity reports. There are no reports in GPSI.

Since GPOs do not support these features and since an enterprise will not want to use multiple software delivery procedures (not if you use standard operating procedures), you will have to integrate a comprehensive software management system with your Active Directory.

Specops Deploy³ from Special Operations Software is a powerful software management solution that relies entirely on the existing structure of your AD. That's because it is nothing more than a set of additions that are integrated to AD—a couple of changes to the Group Policy Object (GPO) Editor, some GPO client-side extensions and a new set of services for the GPSI server. Installing Specops Deploy can be as simple as that if you want it to. But, if you want to make sure everything is as fine-tuned as possible, you'll probably want to add managed software distribution servers which you can set up through the distributed file system (DFS). You might also want to make sure the binary transfer information transfer service (BITS) version 2.0 is deployed to your clients. This can actually be done through Specops as your first deployment and will immediately help control bandwidth usage during deployments from that point on. But that's it.

Installing Specops Deploy is a lot simpler than installing any other software deployment tool. In Specops Deploy, you just run through the tabs on the startup screen (see Figure 4) and it automatically tells you what to do. It does require the Microsoft Message Queuing service (MSMQ) and you do require your original Windows Server 2003 installation CD for this, but besides that it's a breeze. For a database, you can use the built-in Microsoft SQL Server Desktop Engine (MSDE) database or point it to an existing SQL Server 2000 database server. It is recommended to use a full version of SQL Server just because software deployment is a corporate service and as such must be properly architected. See Figure 5 for an illustration of a complete Specops Deploy architecture.

³ For more information on SpecOps Deploy, visit <http://www.specopssoft.com/products/specopsdeploy/Default.asp>.

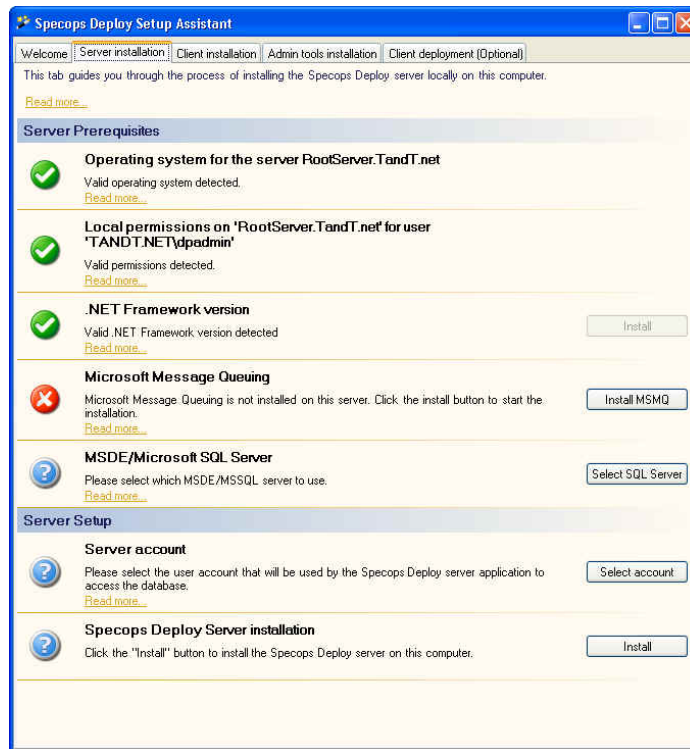


Figure 4. Installing Specops Deploy. Just follow the prompts. Installation can take as little as one hour or less when properly prepared.

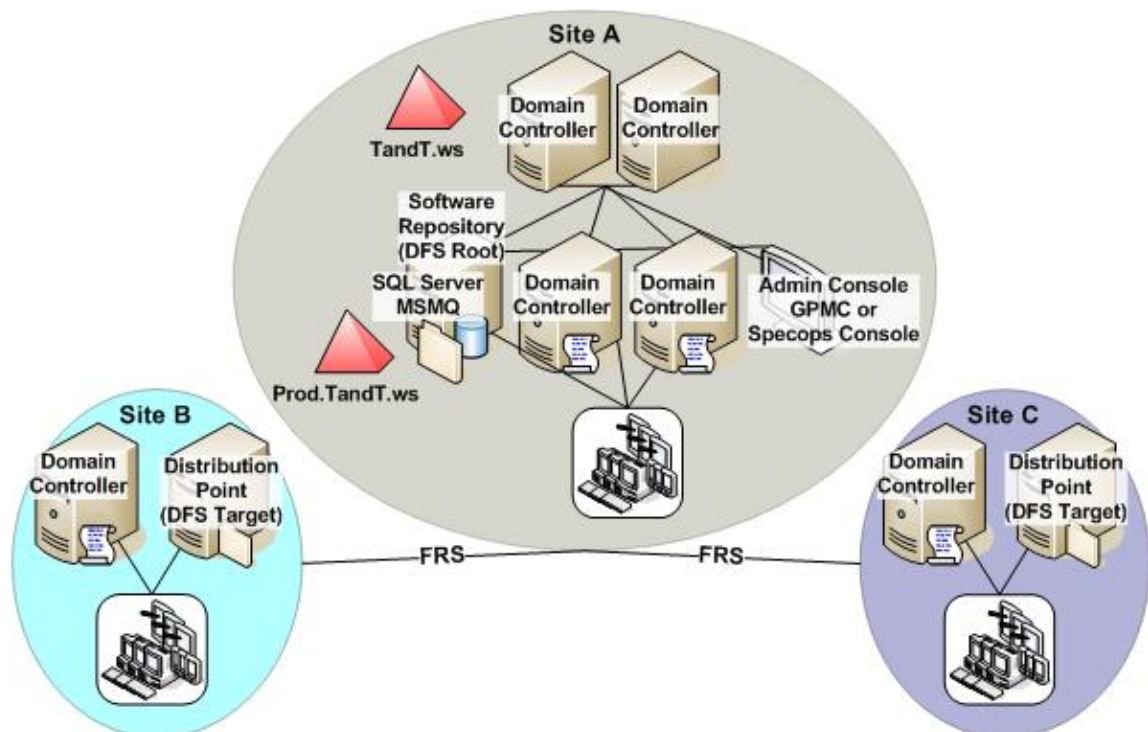


Figure 5. Architecting Specops Deploy. The Specops Deploy architecture is simple because it reuses the Active Directory structure.

Once installed, Specops Deploy puts GPSI on steroids. Delivering software is really straight forward: select or create a GPO, identify who the targets are: computers, users, groups or sites, and select the package to deploy. It's as easy as 1-2-3 (see Figure 6) and since it is done through a Group Policy, no one needs to learn new tools. In fact, Deploy can be much more sophisticated in its delivery targets. These include:

- Computer manufacturer and model
- Specific BIOS settings
- Hard drive size
- Processor speed
- Memory size
- IP address range
- Client language
- Environment variables
- Registry values
- INI/XML file content
- Installed software
- Files on the client system
- Custom Windows Management Instrumentation (WMI) queries
- Imported computer or user lists

This makes software delivery very granular. The most interesting of these is installed software because this means that you can create dependencies in your distributions, ensuring that product A will only be delivered if product B is already on the system and so on. This is a very powerful delivery condition.

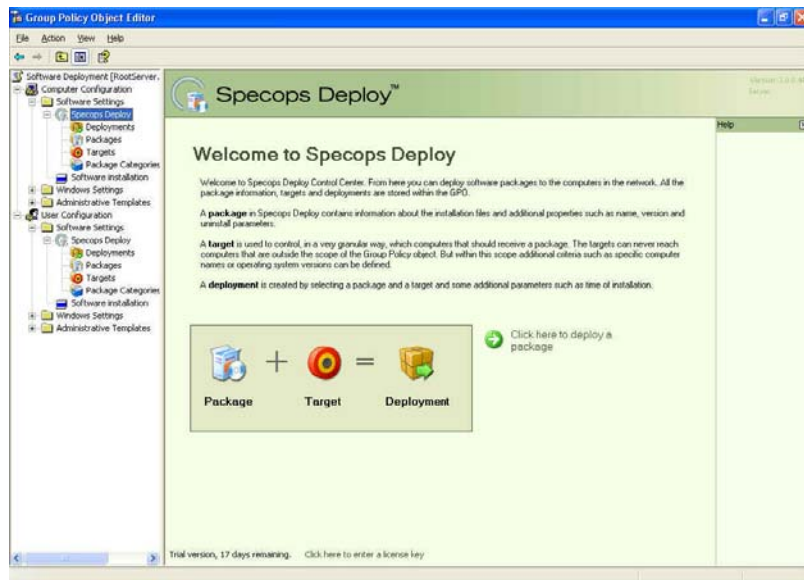


Figure 6. Deploying Software. It's as easy as 1-2-3!

For delegation purposes, Specops Deploy offers a special administrative console that can be deployed to operators. This lets them manage software deployments without needing access to the Group Policy or Active Directory consoles. As for software delivery, Specops Deploy brings a lot of flexibility to this service along with great reporting. Reports will let you drill down on any issues so you can see exactly what happened and why.

Implement a DFS Structure for Software Delivery Points

Software delivery is a good place to use the Distributed File System (DFS) since it allows you to use a single alias for all deposits wherever they are. Domain-based DFS modifies the way you use the universal naming convention (UNC) for shared folders. Traditionally, UNC's are composed of [\\servername\sharename](#) limiting them to one single server. Domain-based DFS uses [\\domainname\sharename](#) and points it to the appropriate local target in distributed sites. This supports the use of a single UNC address or alias for software installation. The use of a single alias makes software packaging and deployment much easier because it only requires a single source folder in the WIS source list.

Windows Installer and Source Lists

Windows Installer files use source lists to identify locations they can use for installation, self-healing and patching. This means two things: first, you must make sure the software installation sources are always available in your network for these features to work; and second, you must include proper source lists in your WIS software packages.

Support for Mobile Users

Specops Deploy first caches software on the local system before launching the installation. Because you can tell Specops Deploy to leave the cached program on the target system, mobile users can automatically benefit from self-healing, even if they are away from the corporate network.

DFS offers several enterprise features for the support and administration of file shares:

- DFS creates a file share alias that is unique through which users can access files on a server. This means that you can change the target file share without impacting users because they access the alias and not the physical file server. This is especially useful for software delivery.
- The DFS namespaces can be linked to any number of actual physical file shares or targets. This is because the DFS namespace can be replicated. If a server must be shut down for any reason, users continue to work by being redirected by DFS to another physical server. The same for software delivery.
- DFS can provide load balancing by distributing file access to a number of physical locations or targets.
- DFS provides transparent consolidation of distributed file shares. If files for a given department are distributed on several physical servers, then DFS can make it appear as if they are all located within a single virtual DFS structure.
- DFS is site aware—that is, it can identify AD sites and use them to redirect users to a file server located within their site. Thus DFS is ideal for distributing file shares that span regions and can be relied on to make sure software installations are always performed locally.
- DFS clients can cache referrals for DFS roots or links for a definable period of time, improving performance by accessing resources without having to perform an AD lookup.

Creating DFS structures is relatively simple, but like all service deployments, they require proper planning. Use the Decision Tree illustrated in Figure 7 to help define your domain-based DFS structure.

Use DFS and FRS in Windows Server 2003 R2

Domain-based DFS works in conjunction with the File Replication Service (FRS) to maintain coherence between DFS targets. Currently, FRS performs only file-level replication. This means that when you change an installation package of 300 MB on the source server, FRS will replicate 300 MB to all file servers in the DFS structure.

Microsoft is set to release version R2 of Windows Server. In this version, FRS will offer byte-level delta replication. This means that if only 20 MB are different in the 300 MB file, FRS will only replicate 20 MB, alleviating the need for third-party replication tools.

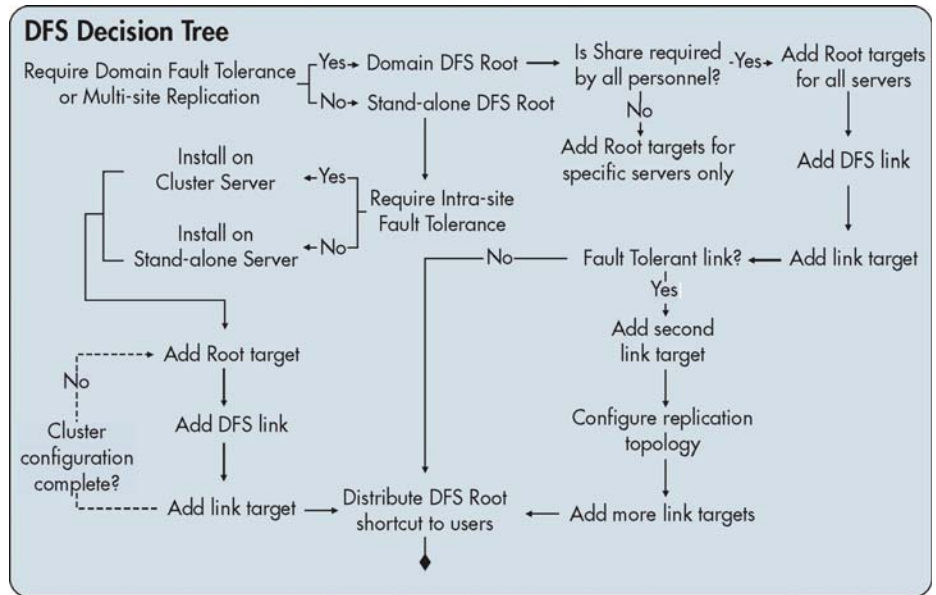


Figure 7. Use the DFS Decision Tree. This decision tree will help you plan and create your DFS structure. Make sure you plan to have a software distribution point in each site where you have existing domain controllers or file servers.

Note that DFS also supports stand-alone DFS roots. Stand-alone roots are not fault tolerant in the same way as domain DFS roots because they are located on a single machine or server cluster. Because of this, standalone DFS roots are not suited to become software distribution points.

Deliver the Right Software to the Right Target

Several advantages can be derived from the implementation of AD-based software delivery. For one thing, AD supports automatic software removal when it falls out of the scope of deployment. This feature is extremely valuable, especially in support of license management. That's because you don't have to create a new deployment to uninstall software. If any PC or user in your organization no longer meets the conditions required for software deployment, AD will automatically remove the software from the system. This works with both GPSI and Specops Deploy. Now, you can create a complete and simple software management strategy. Here's how.

Microsoft System Management Server (SMS)

While the previous version of SMS fully supported automatic software removal, the current version, 2003, no longer includes this feature. This means that users of SMS must create explicit software removal packages in order to control software licenses. This increases the license management total cost of ownership (TCO).

Let's say that you've decided to deliver software on a PC basis because it provides a lower management cost. Let's also say that you've decided to build systems based on a model like the PASS model. This means that you will be deploying software to user roles, but delivering the software to the PC instead of the user. However, if you deliver software on a PC basis, you must ensure your enterprise software delivery strategy supports PC repurposing. To do this, you must maintain a coherence database linking users and their PCs. This allows you to deploy software to role-based configurations tied to user groups, but assign the software to the PC instead of the user. The best way to do this is to deploy software to global security groups that include only computers. Both Windows 2000 (with Service Pack 1) and Windows Server 2003 allow you to assign machine accounts to the source global groups. When group memberships change, software will either be added or removed automatically. To guarantee that this works, you should also make sure that all the software you manage and deploy is prepared in Windows Installer packages.

But before you can do this, you need to make some preparations. Start by doing an inventory of all software in your network. You'll want to use the software kernel concept, and identify all non-kernel software. Regroup non-kernel software into role-based categories—groupings of software that are the same for given IT user roles within the enterprise. Next, create Global Security Groups for each role within AD. Assign principal machines to each user, and create an inventory tying together user, principal machine, and software category for each user. Assign the machines in AD to the appropriate Global Groups. Then assign the software products for each software role to the appropriate security groups. See Figure 8 for the relationship of the different required databases. These include:

- Package Repository — A central deposit for all authorized software
- System Kernel Inventory — An inventory of all the components in the kernel
- Role-based Configuration Deposit — Identify each of the packages found within each configuration
- Vocational groupings — Regroup all of the users or servers belonging to a given IT role

- Core Inventory Database — To serve as a central data repository
- Web-based Report System — To provide detailed information on all inventories at all times.

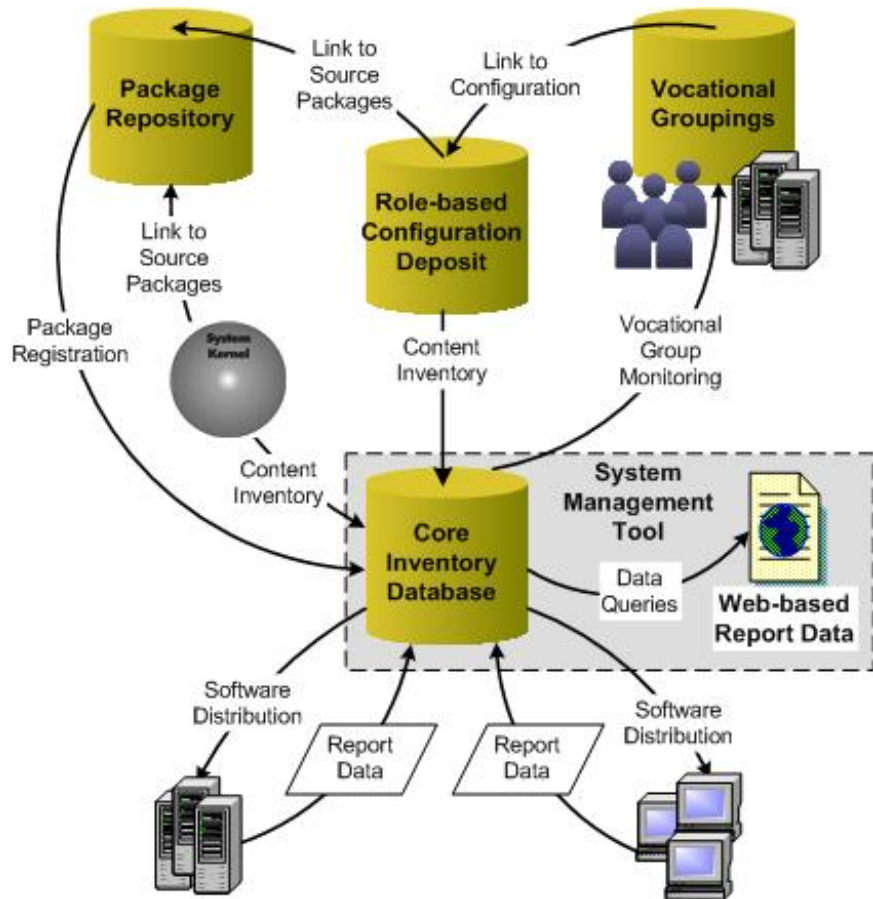


Figure 8. A Complete Automated Systems Management Solution. Combine several different databases to simplify software management through Active Directory. Using both Specops Deploy and Specops Inventory fully supports this automated model.

Using the Automated Software Management Model

Once this system is in place, all you need to do to deliver the proper software to a system is ensure it's a member of the appropriate group within AD. Then, if the PC's vocation changes and it needs repurposing, just change its group memberships. Specops Deploy will automatically uninstall unnecessary software and install software belonging to the new vocation. This is a powerful license management and software deployment model since it is much easier to teach an operator to change the groups to which a machine belongs than to teach them to use a software distribution console (see Figure 9).

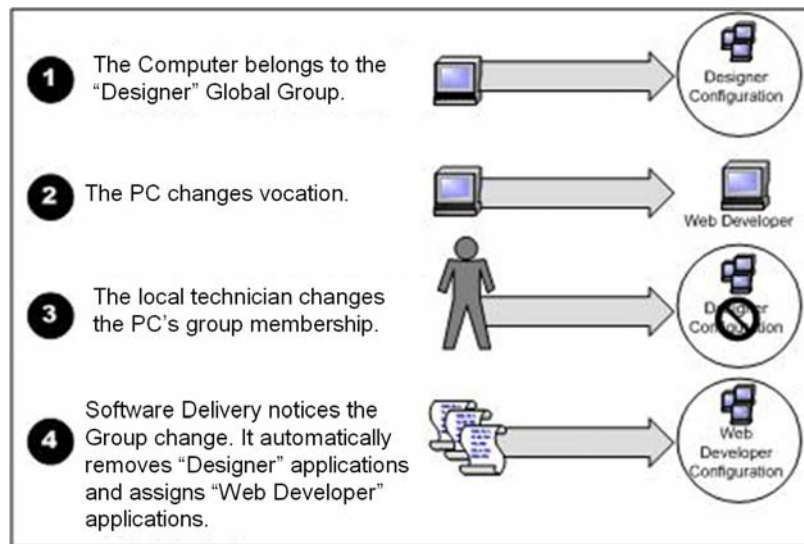


Figure 9. Managing Software Deployment through Group Memberships. Software management becomes very simply when an automated deployment strategy is put in place.

It's important to note that for this process to work you must include the proper *uninstallation* instructions within each software delivery package. If you don't, the software you deploy won't be removed automatically when a PC is removed from a group authorizing the installation and use of the software. In AD, you do this by selecting the Uninstall this application when it falls out of the scope of management check box in the package properties. Because WIS files always uninstall cleanly, this process should always work (see Figure 10).

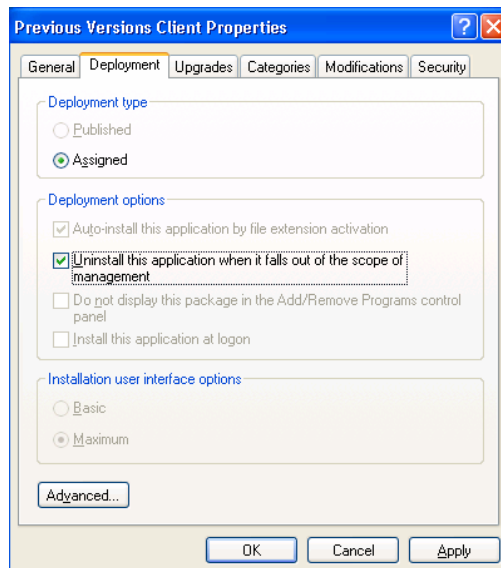


Figure 10. Managing Software Deployment through Group Memberships. Software management becomes very simply when an automated deployment strategy is put in place.

Best Practices for a Simplified Software Deployment Model

There you have it; simple software management through group memberships in Active Directory. Not only does this provide the most powerful software management strategy, but it also creates *compliant* software delivery, matching license usage to actual software installations. Few organizations can claim that they have this capability today. If you want to implement this strategy, use the following steps:

1. Make sure your Active Directory is properly set up. Use the proposed sample chapter on page 8 to perform a sanity check on your AD.
2. Perform a complete inventory of deployed software. Specops Inventory can help here.
3. Modify your system construction strategy to match the goals of the PASS model.
4. Group all non-kernel software into role-based or vocational configurations.
5. Package all software for the Windows Installer Service.
6. Create a database mapping usernames to principal PCs and map it to the new vocational configurations you created.
7. Put your software deployment system in place. For example, install Specops Deploy and create DFS-based software distribution points.
8. Deploy all software based on global security groups containing only computer accounts. Also, make sure you enable the automatic removal of software.
9. Keep the software deployment consoles for your core software purposing team.
10. Train your technical staff to control software deployments through group membership management. Make sure they have access to proper AD delegation consoles for this purpose.

Now, you're ready to relax as all the software in your network will be properly managed.