Deploying Software with Group Policy

Whitepaper

Written by
Darren Mar-Elia
Chief Technology Officer
Microsoft Group Policy MVP
SDM Software, Inc.

Abstract

Group Policy is the feature in Microsoft Windows that provides configuration management for Windows servers and desktops in an Active Directory environment. The Software Installation feature within Group Policy provides a software distribution capability for your Windows network, leveraging the Windows Installer packaging and installation technology to provide targeted, unattended installation of applications to your users and computers.

The Software Installation feature provides a number of capabilities, but they are not always obvious. Best practices and preferred techniques for using Software Installation are captured in this whitepaper. In addition, **SDM Software's Desktop Policy Manager** product is presented as a simplified means of deploying applications—building on top of Group Policy's Software Installation feature.

Table of Contents

Overview

There are many ways to automate the deployment of software to your Windows servers and desktops. Some solutions require special re-packaging of application setups and require complex server infrastructures to provide deployment services. Fortunately, for many organizations, these complex requirements aren't needed to automate simple desktop or server deployment tasks. Windows Group Policy can provide tremendous value for most organizations. Group Policy provides software installation features that lets you deploy Windows applications on a per-computer or per-user basis to your Active Directory-based Windows environment. And while Group Policy Software Installation (GPSI) has limitations, it meets the needs of many organizations. In this paper, I'll take an in-depth look at the GPSI feature and reveal practical tips and best practices on how you can use this technology to its greatest effect.

Group Policy Software Installation Features

As I mentioned, Group Policy provides the ability to deploy software to your computers and users within an Active Directory environment. (Note that the GPSI feature is not available on the local Group Policy Object (GPO).) In fact, GPSI supports two different types of installations—**publishing** and **assigning** of applications. The differences between each are subtle, yet important. Assignment is available on either a per-computer or per-user basis whereas publishing is only available per-user.

> 🕮 **NOTE:** By per-computer or per-user, I mean that you can install an application so that it is deployed to a computer, regardless of which user is logged on, or to a user, regardless of which computer they log onto.

Applications that are assigned provide for a mandatory installation option. That is, when you assign an application to a computer or user, you are saying that you want that application installed regardless of whether the user chooses to install it or not. By contrast, when you publish the application, you give the user the option of installing it, and they can do so by optionally visiting the Add/Remove Programs control panel applet and selecting the published application to install.

Application assignment also presents an additional deployment option. You can designate a user-assigned application to be installed on first-use rather than when the user logs onto their workstation. This saves time when deploying a large application that may or may not be used by all users immediately. The install-on-first-use behavior lets the user dictate when they install the application—the installation is activated when the user tries to open a document associated with the application or a shortcut on their Desktop or Start Menu that points to the application.

Publishing and assignment options provide flexibility for making applications available to your user population. You might decide that you need to assign mandatory applications such as Microsoft Office or a line-of-business application to ensure that all users have access to it. But, for those optional applications that are not licensed for the entire organization, you may choose to publish the application setup to select users that can install it as they need it. The advantage of Group Policy-based software deployment is that you can use the same targeting mechanism for software deployments that you use for other Group Policy settings. For example, you can control what users get a published or assigned application by controlling where a GPO is linked, how it is security filtered, or how it is affected by a Windows Management Instrumentation (WMI) filter.

Application Deployment Lifecycle Management
In addition to providing two modes of software installation, the GPSI feature provides the ability to manage the complete lifecycle of application deployment—from install to upgrade to patching and even removal. Much of this lifecycle management is built into the GPSI feature but is not explicitly called out; It requires using best practices that I'll describe later in this document. But by and large all phases of application deployment are supported.

Packaging Requirements
Many commercial software deployment solutions require you to repackage your application setups into proprietary setup formats. The GPSI feature supports the Microsoft standard Windows Installer (MSI) packaging format. The MSI format is the most common packaging format in use today and the GPSI feature integrates tightly into the Windows Installer engine to provide a number of unique features that add value to your software deployment processes.

Some of these features include repair-on-demand, where an application with missing or corrupted files is repaired automatically when the user tries to run the application. Also included is the ability to have any application deployed via GPSI be automatically elevated in their privilege level during install. This allows the Windows Installer engine to install an application, either per-user or per-computer, without requiring the user who might be initiating that installation to be a privileged user (administrator or power user) on their Windows system. This has obvious security advantages and gets around the problems related to certain application setups requiring administrative rights to install.

The downside to this tight integration with the Windows Installer is that the GPSI feature requires all application setups to be packaged using the MSI format.

> 🕮 **NOTE:** GPSI does provide for a packaging mode called "ZAP" or Zero Application Packaging. ZAP files are simple text files that wrap an existing setup.exe or similar setup package. ZAP files can be deployed in GPSI via the user-specific publishing feature only (i.e. they cannot be assigned to a computer or user). In addition, ZAP files do not benefit from privilege escalation during setup, so if the user who is installing the application does not have the required privileges, the setup will fail. See Microsoft Knowledge Base article # 231747 for more details.

Now that we have a sense of what GPSI is capable of, let's take a look at how to use this Group Policy capability at its best.

Deploying Software Using Group Policy Software Installation

The first thing you need to know to get started using GPSI is how to find it within Group Policy! The GPSI feature is not available from the local Group Policy Object (i.e. by launching gpedit.msc). Microsoft did not implement this feature in the local GPO. Therefore, you'll need an Active Directory installation to start using this feature. Once you've created a GPO using the Microsoft Group Policy Management Console (GPMC) or the AD Users and Computers MMC snap-in, edit that GPO to bring up the Group Policy editor MMC snap-in. As I mentioned earlier, you can deploy software using GPSI as either a per-computer or per-user deployment. The per-computer feature can be found in the GP editor under Computer Configuration\Software Settings\Software Installation (see **Figure 1** below), while the per-user deployment feature is under User Configuration\Software Settings\Software Installation.

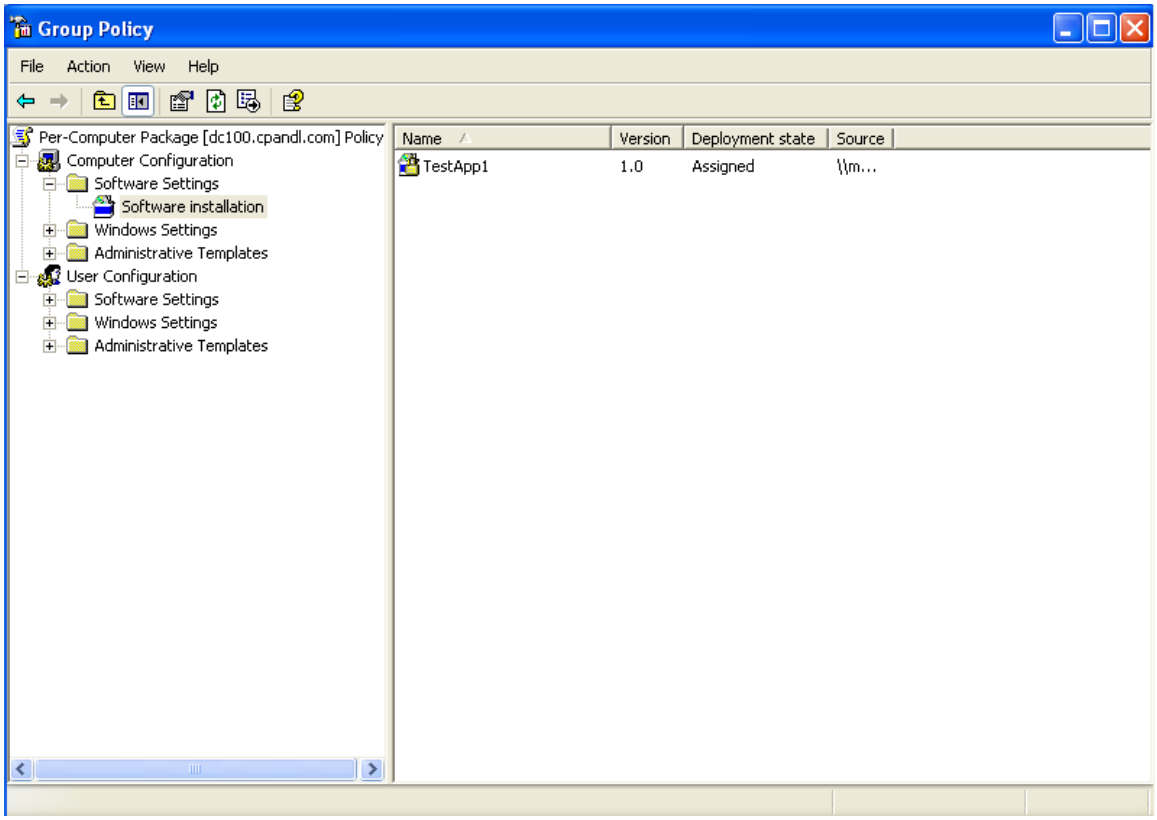

Figure 1: Viewing per-computer package deployments

To deploy a new package, you right-click the "Software Installation" node and choose New, Package. The next dialog you receive will be the familiar File, Open dialog that lets you choose the .MSI package (or .zap in the case of user published packages). Doing the right thing at this dialog is important, but not obvious. If you just browse to your package using a literal path (e.g. c:\packages\myapp.msi) then that path will be stored in the GPO exactly that way. This means that when client computers or users read the GPO, they will look for the package at c:\packages\myapp.msi. This is fine if the package is stored on every desktop that processes the GPO. But if the package is, most likely, on a server share, then the client will never find it. So, if you want your package reference to point to a server share, you need to specify that within this File Open dialog. To do that, simply enter the UNC path to your package share on the file name input box, as shown in **Figure 2**.

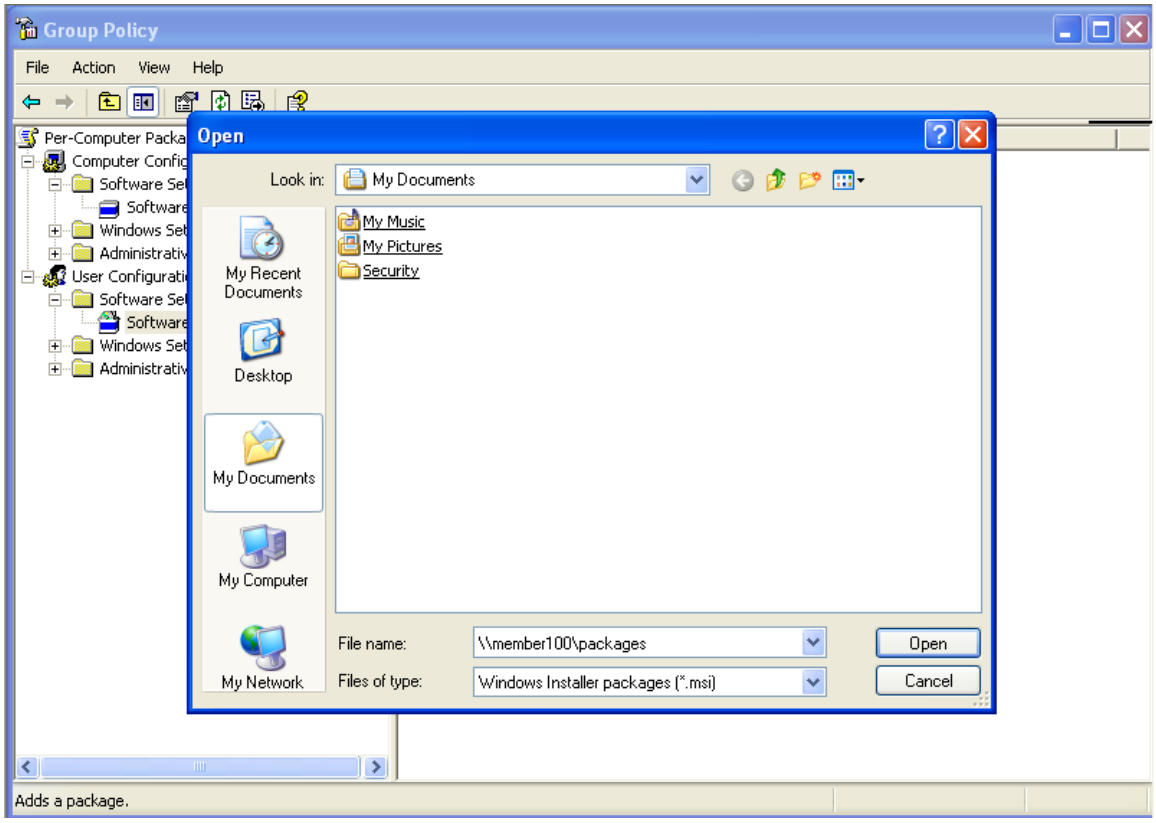

Figure 2: Entering the correct path to a package

Entering the UNC path here ensures that path is stored within the GPO, so that clients will always find the correct path to the package (assuming the computers or users have permissions to read that package off the server share).

> ☞ **TIP:** The best way to deploy packages using GPSI is to use the Distributed File System (DFS) feature built into Windows Server. This features allows you to abstract the file path from the physical location of the file so that if you need to move application packages from one server to another, the file path stored in the GPO for that package will not need to change. This is especially important because the native GPSI feature does not support changing the package path for existing packages--you need to create a new package, which has an impact on clients that have already installed the package via Group Policy.

Once you've entered the path to the package, the next step is to choose the deployment mode--published or assigned. In the case of computer-specific deployments, your only choice is assigned. You can then choose the advanced option, if desired, which lets you modify additional options prior to deploying the packages. In most cases, you will want to choose this advanced option. The advanced option lets you do such things as: add Windows Installer transforms to your deployments (transforms are special files created by your packaging tool that let you modify the behavior of a .msi file during installation); set removal options for the application when the GPO falls out of scope for the computer or user; and modify the security

of the package to control which users or computers can deploy it. Let's go through and see what each of the tabs under the advanced dialog does:

- **General**: Lets you change the name of the package as it appears to the client and add a URL for support
- **Deployment**: Allows change of the deployment mode from published to assigned (if available) and control of the deployment behavior if the package falls out of scope of the user or computer. The option to "Auto-install by file extension activation" provides the install-on-first-use behavior described earlier. This is also known as "advertisement." You can also change the level of user interface the user sees during the installation. In most cases, for an unattended installation, "Basic" is the best choice here.
- **Upgrades**: The upgrades tab provides a useful feature for the management of an application's lifecycle. If you have upgrades to an application over time, you can use the upgrade tab to create relationships between application versions deployed via Group Policy. This allows you to specify, for example, that if a user has version 1 of an application deployed via GPSI, you want them to get version 2 as well. This feature, however, assumes that the Windows Installer packages have logic built in to upgrade existing installations. If not, then creating the upgrade relationship will not result in an upgraded application at the client computer.
- **Categories**: The categories tab allows you to assign a package to a category. This is only relevant for packages that will appear in the Add/Remove Programs control panel applet. If you categorize a package, when a user goes to install that package from the Add/Remove Program applet, that package will appear under that particular category. This feature is useful if you have many published applications and wish to make it easier for users to find the right package. You can create categories as well as set other defaults for software installation by right-clicking the Software Installation node in the GP editor tool, then choosing Properties from the context menu.
- **Modifications**: The modifications tab is where you add Windows Installer transform files (i.e. files with a .mst extension) to your package. **NOTE**: **You can only add transforms during the initial deployment of the package. You can't add them once the package has been deployed.**
- **Security**: This is the familiar Access Control List (ACL) editor dialog, where you can modify the permissions of an individual package. By default, when you create a package, the Authenticated Users group, which includes all computers and users in an Active Directory domain, is granted read access to the package. This is all that's needed for a computer or user

to be able to install the package. Using this dialog, however, you could remove the Authenticated Users group from the ACL and grant read access to a specific user or computer group in order to control which users and computers can install this application. This is useful if you have multiple packages deployed in a single GPO but don't want all users or computers that process that GPO to have equal access to all applications.

Once a package is deployed, the policy will be processed by computers and users during their next **foreground** Group Policy processing cycle. That is, a package will not install during the periodic, background refreshes of policy that normally occur. It will only occur during computer startup (in the case of computer-assigned packages) and user logon (in the case of user-assigned or published packages). However, changes to existing package options, such as changing the option that tells Windows whether or not to remove a package that is no longer in scope, will be updated during background refreshes.

That's the basics. Next we'll look at best practices for using GPSI.

Best Practices for Deploying Software Using GPSI

The previous sections discussed the capabilities around GPSI and how to deploy a package using it. Now let's look at some practical tips for using this feature effectively.

Creating Package Installation Points

We've already talked about using DFS to refer to package paths to shield your packages from server changes. This is important because once a package is deployed via GPSI, the path to the package cannot be changed. But how do you get the installation packages onto your server shares in the first place? Is it just a matter of copying the setup files from the application CD? In many cases, that's probably not the case. Many Windows installer packages support the notion of an administrative setup. For example, Microsoft Office lets you create a network share of its setup files using the /a option during setup. Why is this desirable for GPSI? Primarily because it is the supported method for patching an existing application package, should the need arise. Windows Installer patch files (typically named with an .msp extension) need to be applied to either the client or an administrative installation of the original package. Let's walk through how you might package an Office 2003 package deployed via GPSI.

Patching Existing GPSI Deployments

Say you've deployed Microsoft Office 2003 using GPSI. You want to patch the administrative install on your package server share with the latest Office update and have all computers or users that had previously installed Office automatically pick up and install the new patch on their local installations. To do that, you simply need to

extract the Windows Installer patch file (.msp file) from the update and apply it to the administrative installation source using the following msiexec.exe command:

msiexec /a \\server\packages\pro11.msi /p c:\officeupdate\officepatch.msp

This command can be executed from the command line. The path shown in the /a option is the name of the main .msi setup file for the package you're patching (in this example, Office 2003's main setup .msi file is called pro11.msi), and it's referred to using the path to the package on the server. The /p option specifies the .msp patch file you wish to apply to the original install package.

Once you've completed patching of the original administrative installation, you're not quite finished. You need to use the Re-deploy feature within GPSI to trigger the re-deploy so that your computers and/or users can pick up the changed package. Re-deploy tells the client to perform a Windows Installer repair of the existing application, picking up any new or updated files from the server while doing so. To issue a re-deploy, open up the GPO that contains the patched package using GP Editor. Right-click on the package and choose All Tasks, Re-Deploy application from the context menu, as shown in **Figure 3.**

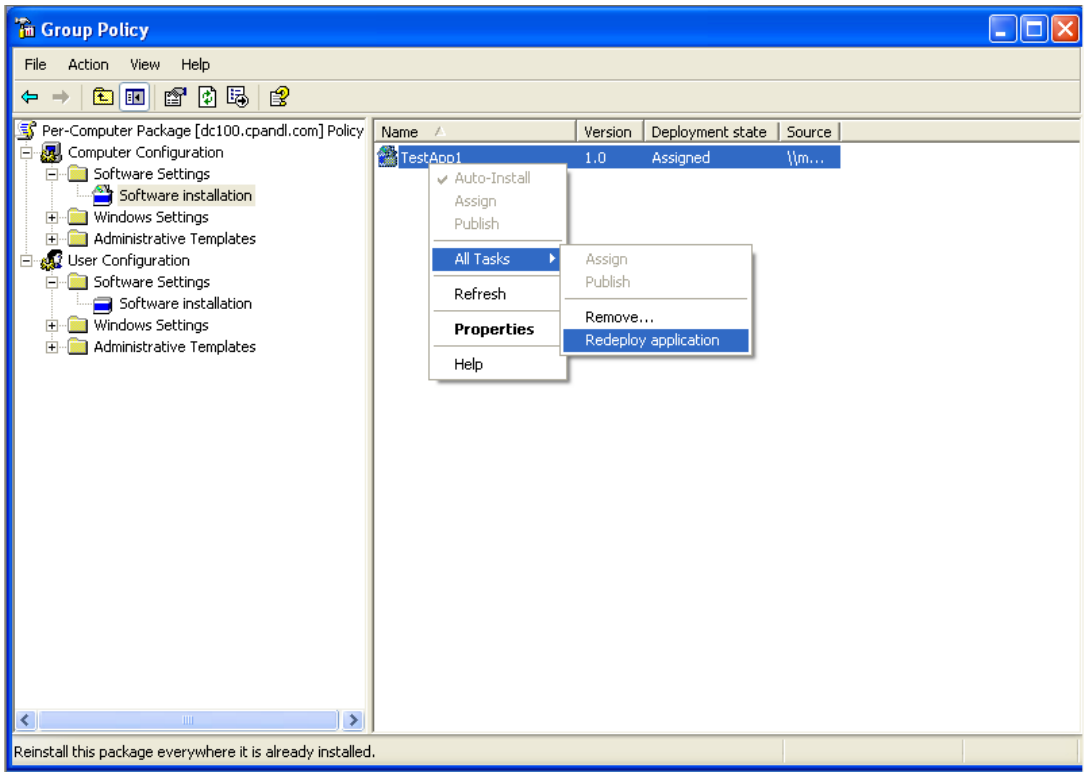

Figure 3: Re-deploying a patched GPSI package

Once you re-deploy the patched package, the computer or user will pick up the patch during the next foreground refresh cycle.

NOTE: With the Release of Office 2007, Microsoft reduced the ability to deploy this version of Office using Group Policy. Namely, you can no longer fully customize Office 2007 using MSI transforms. As a result, the ability to deploy Office 2007 using GPSI is very limited. For more information about what you can do with Office 2007 and GPSI, see the following TechNet article: http://technet.microsoft.com/en-us/library/cc179214(TechNet.10).aspx

Per-User or Per-Computer?

How do you know whether it's best to deploy a GPSI package per-user or per-computer? It depends on how ubiquitous you want the application to be. For example, per-computer deployments make an application available to every user on a given computer where the package is installed. By contrast, per-user deployments will install everywhere a targeted user logs on, ensuring that the application is available on every PC they use. This might not be a good thing if you want the application to only be available to a particular user of that PC. Ideally, per-user packages will only install per-user--that is, the icons to start the application will only install within the targeted user's profile. But that does not prevent the intrepid user who logs onto the computer where the per-user package was installed from finding the application and running it. Per-user deployments in environments where users move around from PC to PC can also have the unwanted effect of "littering" PCs with applications that should not have been installed but for the one time a targeted user happened to log onto them. Given these circumstances, there is no one right answer for this question. Generally, per-computer assignment is the most popular deployment mode used with GPSI.

Uninstalling Applications the "Right" Way

Sometimes you might be tempted to use the Add/Remove Program (ARP) Control Panel applet to uninstall an application that was deployed via GPSI – not good if you ever want that package re-deployed via GPSI. Using the ARP method breaks the relationship between the GPSI package and workstation, because there is additional data stored in that computer's registry which tells GPSI whether or not the application has been installed. Removing the application outside of GPSI orphans that data, so that the next time GP is processed, GPSI still thinks the application is there and will not try to reinstall it. In order to remove a GPSI application, it's best to use the Remove feature (right-click a package in the GP Editor and choose All Tasks, Remove). Or, when deploying the package, select the option to have GPSI remove an application when the computer or user is no longer in scope. That way you can remove the application by moving the computer or user account outside the scope of the GPO or by putting the user or computer in a security group that is permissioned away from the GPO.

Simplifying Application Deployment with SDM Software's DESKTOP POLICY MANAGER

Now that we've discussed some of the features available in GPSI, let's look at how SDM Software's **Desktop Policy Manager** product, which provides a simplified web interface for managing desktop configuration using Group Policy, can help make application deployments a snap! **Desktop Policy Manager (DPM)** supports both per-computer and per-user software deployments. Within the product, you can define configuration profiles that contain any number of different policy areas, including Software Deployment. Using DPM's web UI, you can easily add MSI packages to a given profile—such as the example in Figure 4 where we've added the Adobe Acrobat Reader setup to a profile to be deployed to all Marketing users in my domain.

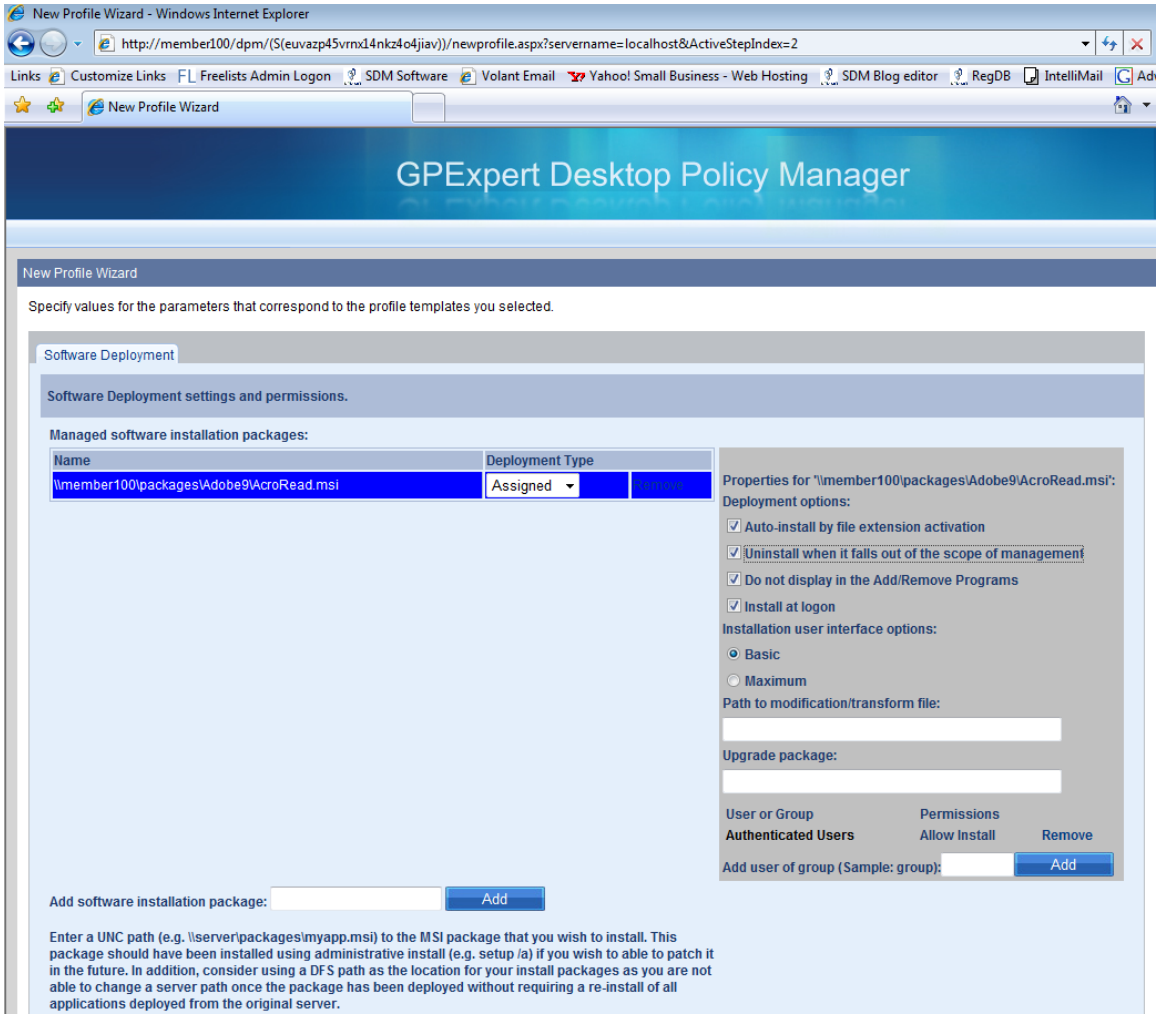

Figure 4: Using DPM to deploy Adobe Acrobat Reader

As you can see in the figure above, DPM simplifies the many options available to you when deploying software using Group Policy. In addition, it makes targeting of applications simpler, because you can either target specific applications within a profile to specific groups, or you can use the simplified targeting of profiles in general, as shown in Figure 5, to more quickly and easily ensure that the right users or computers get the right applications.

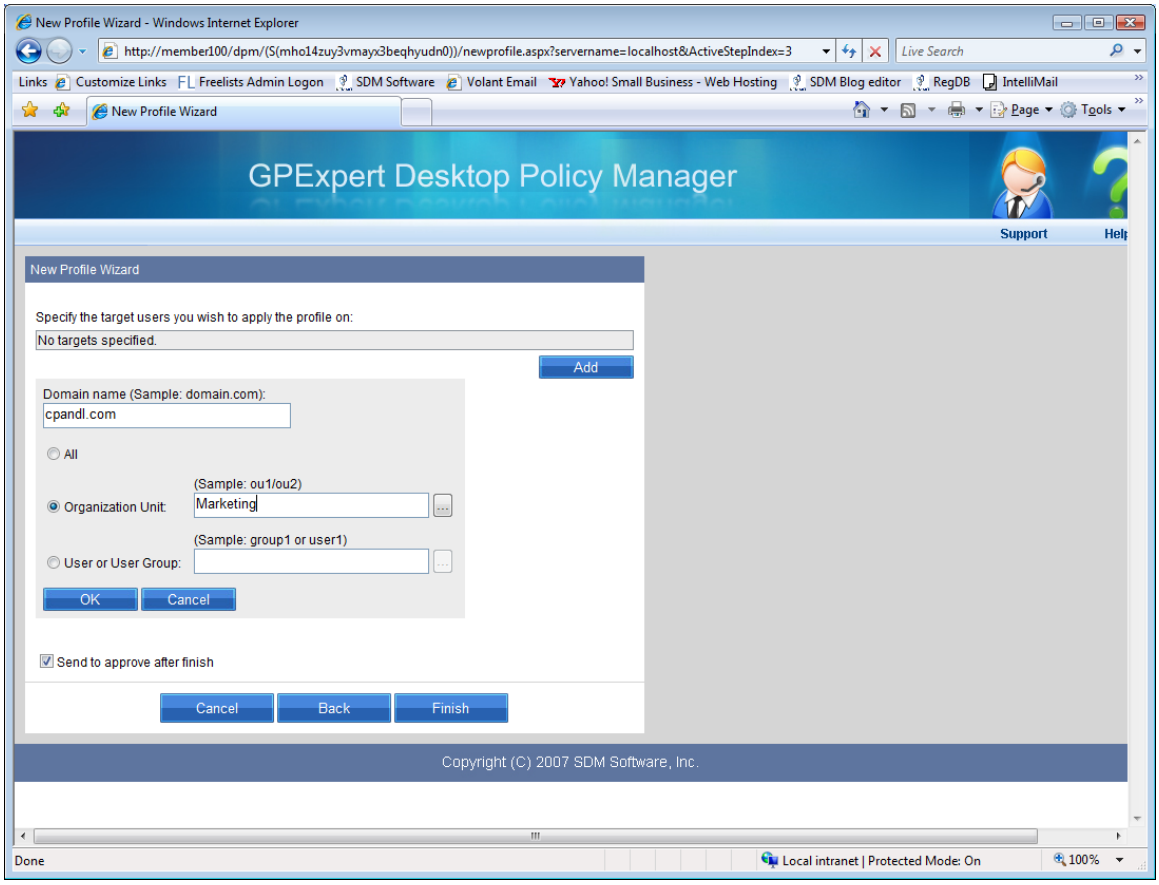

Figure 5: Targeting the Marketing OU using DPM's Advanced Targeting Feature.

Summary

The GPSI feature in Group Policy provides a number of excellent features for automating software deployment in your Active Directory environment without spending tens of thousands of dollars on expensive software distribution solutions. From support for Windows Installer packages to deployment-patching-removal lifecycle management, GPSI provides a mechanism for managing packages. And while you might have to go through extra steps to perform steps like patching and changing package paths, the tradeoffs in cost are worth it for many environments.

The process can be further simplified by using **SDM Software's Desktop Policy Manager** product, which provides a full set of features for not only simplifying application deployment, but also a range of other Windows configuration management tasks.