realtimepublishers.com™

The Administrator Shortcut Guide™ To

Configuration Management for the Windows Enterprise

Configuresoft

Don Jones

Copyright Statement

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

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

Chapter 2: Understanding Continuous Configuration Management

In the previous chapter, I mentioned the term *continuous configuration management*. It's the word *continuous*, of course, that sets it apart from the configuration management you hear folks talking about these days. But what does continuous configuration management really entail? How do you implement it in your environment? In this chapter, I'll explore what continuous configuration management really means, introduce you to some of the tools and technologies it involves, and discuss the underlying Windows technologies that make it possible.

Defining Continuous Configuration Management

Most administrators define configuration management as a set of processes and tools designed to manage changes to an environment's configuration. This definition is fairly accurate, except that configuration management is generally a one-way street, including only *provisioning* and *software deployment*.

📖 I'll discuss both provisioning and software deployment shortly.

The problem with traditional configuration management is that it assumes that nothing is changing in the configuration unless managed changes are made. This assumption is ridiculous, of course, because change is *always* occurring to computers' configurations, no matter how tightly you lock down the user environment. Continuous configuration management includes the addition of a *continuous management and enforcement* step, which provides an interactive facility between the management process and the managed configurations. This step ensures that the configuration doesn't drift away from the desired standard over time.

Provisioning

Provisioning is the first step in any configuration management process. It provides a stable, standard, approved initial configuration. In the Windows world, this initial configuration generally includes a standardized setup routine, or perhaps an entire disk image that includes the computer's operating system (OS) and applications. Provisioning represents your "clean start" for configuration management, enabling each managed computer to start at a common baseline configuration.

You're likely to have several baseline configurations in your environment. You might have a standard Windows 2000 (Win2K) Professional disk image, a Windows XP Professional disk image, and several server images for new servers. You might also maintain separate images for laptop and desktop computers. A shortcoming of traditional configuration management is that *it all relies on a standard starting point*.

Suppose that your company decided to offer your users a lot of flexibility, allowing them to select any of a half-dozen laptop computer models or perhaps one of four or five desktop computers. Users could then select a computer with the size, shape, and performance characteristics that best met their needs. Creating a standard image across those platforms would be practically impossible, which is why most companies restrict users to one or two different models at best. In short, traditional configuration management, with its need for a standardized starting point, has placed a limitation on the flexibility users have when it comes to selecting appropriate technologies.

Tools and Technologies

Several types of tools and technologies exist to aid in provisioning. One simple option is built right into Windows: support for unattended installation answer files. Listing 2.1 shows a sample—the unattended installation answer file provided with Win2K Professional.

```
[Unattended]
UnattendMode = FullUnattended
OemPreinstall = No
TargetPath = Winnt
Filesystem = LeaveAlone
[UserData]
FullName = “Your user name”
OrgName = “Your organization name”
; It is recommended that you avoid using spaces in the ComputerName
; value.
ComputerName = “YourComputer_name”
; To ensure a fully unattended installation, you must provide a value
; for the ProductId key.
ProductId = “Your product ID”
[GuiUnattended]
; Sets the TimeZone. For example, to set the TimeZone for the
; Pacific Northwest, use a value of “004.” Be sure to use the
; numeric value that represents your own time zone. To look up
; a numeric value, see the Unattend.doc file on the Windows 2000 CD.
TimeZone = “YourTimeZone”
; It is recommended that you change the administrator password
; before the computer is placed at its final destination.
AdminPassword = AdminPassword
; Tells Unattended Setup to turn AutoLogon on and log on once.
AutoLogon = Yes
AutoLogonCount = 1
[LicenseFilePrintData]
; This section is used for server installs.
AutoMode = “PerServer”
AutoUsers = “5”
[GuiRunOnce]
; List the programs that you want to start when you log on to the
; computer for the first time.
[Display]
BitsPerPel = 8
XResolution = 800
YResolution = 600
VRefresh = 70
[Networking]
; When you set the value of the InstallDefaultComponents key
; to Yes, Setup will install default networking components.
; The components to be set are TCP/IP, File and Print Sharing,
; and Client for Microsoft Networks.
InstallDefaultComponents = Yes
[Identification]
; Identifies your workgroup. It is recommended that you avoid
; using spaces in this value.
JoinWorkgroup = “YourWorkgroup”
```

Listing 2.1: The unattended installation answer file provided with Win2K Professional.

As you can see, this answer file offers the ability to customize installed components, allowing you to create a standard set of components for your Windows installations. This file works with Windows' built-in provisioning abilities, so it works across hardware platforms. However, it does nothing for installing applications, configuring security, and so forth. Windows does include tools to deploy applications, such as Sysprep, but administrators don't often like to use these tools because the tools can be slow and difficult to work with.

Microsoft's Remote Installation Services (RIS) combines Microsoft's Sysprep imaging and unattended installation answer files to create an automated provisioning system. Essentially, computers with a Preboot Execution Environment (PXE)-compliant network adapter attach directly to a specially configured Windows server running the RIS server software. RIS selects and deploys an OS, along with applications and initial settings, to the computer. Although RIS is eminently usable, it is still relatively complex to work with and isn't a favorite of all administrators.

More straightforward products such as Symantec's Norton Ghost create block-by-block copies of hard drives. These copies allow you to provision systems by essentially copying an existing system that you have set up manually. Everything is deployed: applications, OS, and security settings, as well as any misconfigurations you might have set up. These tools allow you to deploy an image to multiple computers at once; however, they are often limited in their ability to deploy images to dissimilar hardware. These tools are perhaps the most popular way to provision new computers, although their limitations have a direct impact on the flexibility provided to end users.

So what are the drawbacks to these provisioning tools? There aren't many. Obviously, it can take some time to create a properly configured system that you can image using either RIS or a tool such as Norton Ghost; a misconfiguration will be imaged along with everything else, and you could wind up deploying dozens of improperly configured machines before you realize it. But with a little caution, RIS and imaging tools are useful provisioning tools. The problem is that these images provide a one-time-only configuration, meaning you should regard them strictly as a convenience and not as a step toward security.

Business Analysis

There is an assumption about provisioning that I want to challenge: The purpose of provisioning is to provide an initial baseline configuration that is acceptable, secure, stable, and complete, all in accordance with your organization's needs. I suggest that, with proper continuous configuration management, provisioning isn't necessary. Before you get upset, let me clarify: I'm not saying that provisioning isn't *desirable* or *convenient*; I'm saying that it isn't—or shouldn't—be *necessary*.

From a deployment perspective, tools such as RIS and Ghost are inarguably a convenient way to quickly deploy a lot of software to a new computer. With a couple of clicks, you can deploy to a computer the Windows OS, all your corporate applications, the latest hotfixes, and any other items you have deemed necessary. There is absolutely no argument from me that this deployment method is the most convenient way to get a new computer set up, and it beats installing all of those items manually.

However, the idea that provisioning is necessary from a configuration management standpoint is false. Configuration management needs to be able to operate no matter what is on your computers, so why should it matter what the starting point is? Again, drive imaging and other provisioning tools serve as a convenient way to get machines out the door with the right baseline, but that is *strictly* a matter of convenience and shouldn't have any impact on configuration management.

The Useless Starting Point

"What do you mean provisioning isn't necessary? That is crazy talk!" The thought seems to be that deploying a well-configured, known-stable, and known-secure image to new computers will somehow ensure that they are deployed correctly and you simply need to make sure that the correct patches get applied in a timely fashion to ensure that the systems are properly managed. *Nothing could be further from the truth.*

First, your image is outdated 20 minutes after you finalize it because OSs and applications are constantly being patched and updated by their manufacturers. Thus, whenever you deploy a new computer using an image, you immediately must start applying the updates issued since the image was created. That creates a vulnerable period for the computer: I've been in organizations in which freshly imaged machines have been infected by newer viruses.

Second, simply pushing patches out to computers *does not* ensure that they're kept up to date. If a user installs a new application, will you know to push patches for that application? If the machine is restored from a backup, where does that leave you in terms of updates?

Ensuring the security and stability of your systems requires you to define a baseline to which all computers must comply. You need to constantly analyze every computer for compliance and apply updates or make configuration changes to bring them into compliance. Computers that don't even run SQL Server, for example, need to be examined for SQL Server-related vulnerabilities *in case* SQL Server somehow was installed (it's not an outlandish idea—Microsoft's MSDE is really SQL Server, and it comes bundled with a number of other applications). If the analysis process determines that SQL Server isn't installed, fine, but you cannot rely on the fact that *you* never installed SQL Server as proof that it's not present.

This stability and security analysis needs to be continuous, so it doesn't really matter how the computer starts out. Your analysis process needs to consider each client without knowing about its past or how it got started: Each computer needs to be analyzed as if it was for the first time ever, every time the analysis is run. So although imaging tools make a *convenient* way to quickly get new systems up and running, they are not truly contributing to a secure, stable environment.

A properly deployed continuous configuration management process, along with the appropriate tools and technologies, will ensure that your computers have the appropriate software (and *only* the appropriate software), the appropriate security settings, and other configuration parameters, no matter what is installed on the computer or how the installed items are configured. In other words, with proper continuous configuration management, there is no need for a baseline configuration. This theme is one I will be revisiting throughout this chapter.

✎ Remember that many organizations settle on one or two laptops to, in part, reduce the number of baseline configurations they must deal with. If you reject the notion that a baseline is necessary for configuration management, you can expand your users' choices, offering computer hardware that more precisely fits their business needs. After all, technology should support the business, not the other way around, right?

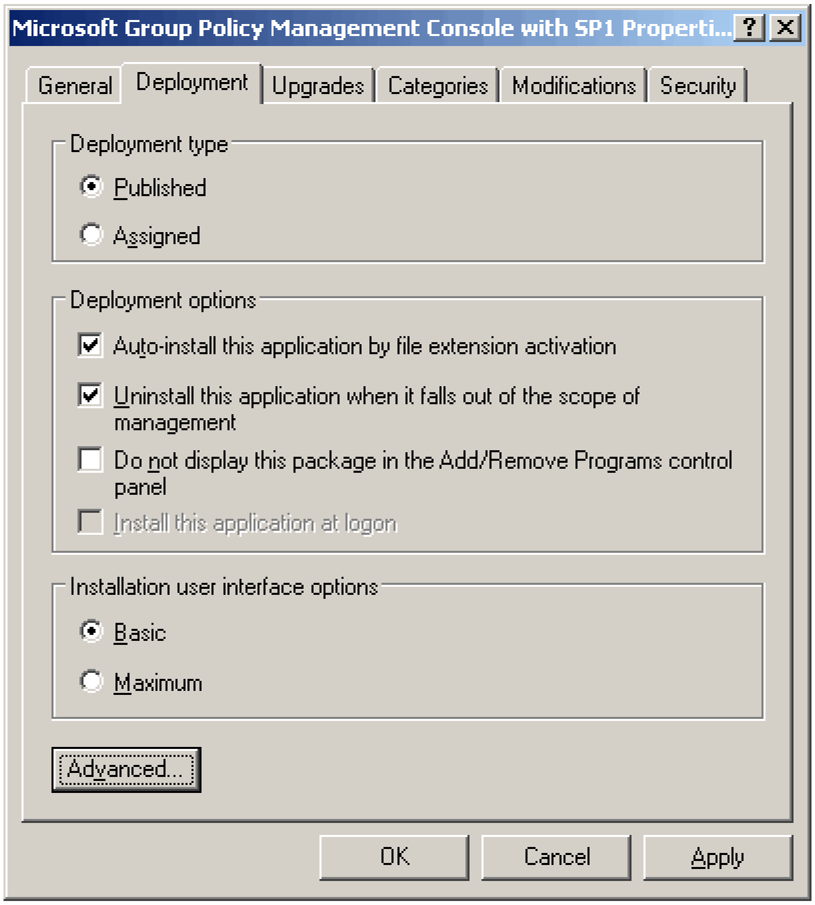
Software Deployment and Ongoing Configuration

Once you've provisioned a system, you can settle in for the long-term task of deploying software to it. Software deployment covers a wide range of needs, from deploying new versions of corporate applications to deploying security updates and patches; from deploying entirely new applications to uninstalling old ones. In traditional configuration management, software deployment is the primary means of keeping systems aligned to a baseline configuration.

> The term *software deployment* is held by some to refer only to deployment of applications, such as Microsoft Office; deploying OS and software updates is referred to as *patch management.* I believe the two areas are beginning to converge, but in the next sections, I'll cover tools for handling both ends of the software deployment spectrum.

Tools and Technologies

Several products can assist with software deployment. Microsoft Active Directory (AD), for example, provides an excellent choice in the form of its Group Policy-based software deployments. As Figure 2.1 illustrates, Windows Installer packages can be assigned to users for automatic installation or published as optional offerings.



Deployment type

● Published

○ Assigned

Deployment options

☑ Auto-install this application by file extension activation

☑ Uninstall this application when it falls out of the scope of management

□ Do not display this package in the Add/Remove Programs control panel

□ Install this application at logon

Installation user interface options

● Basic

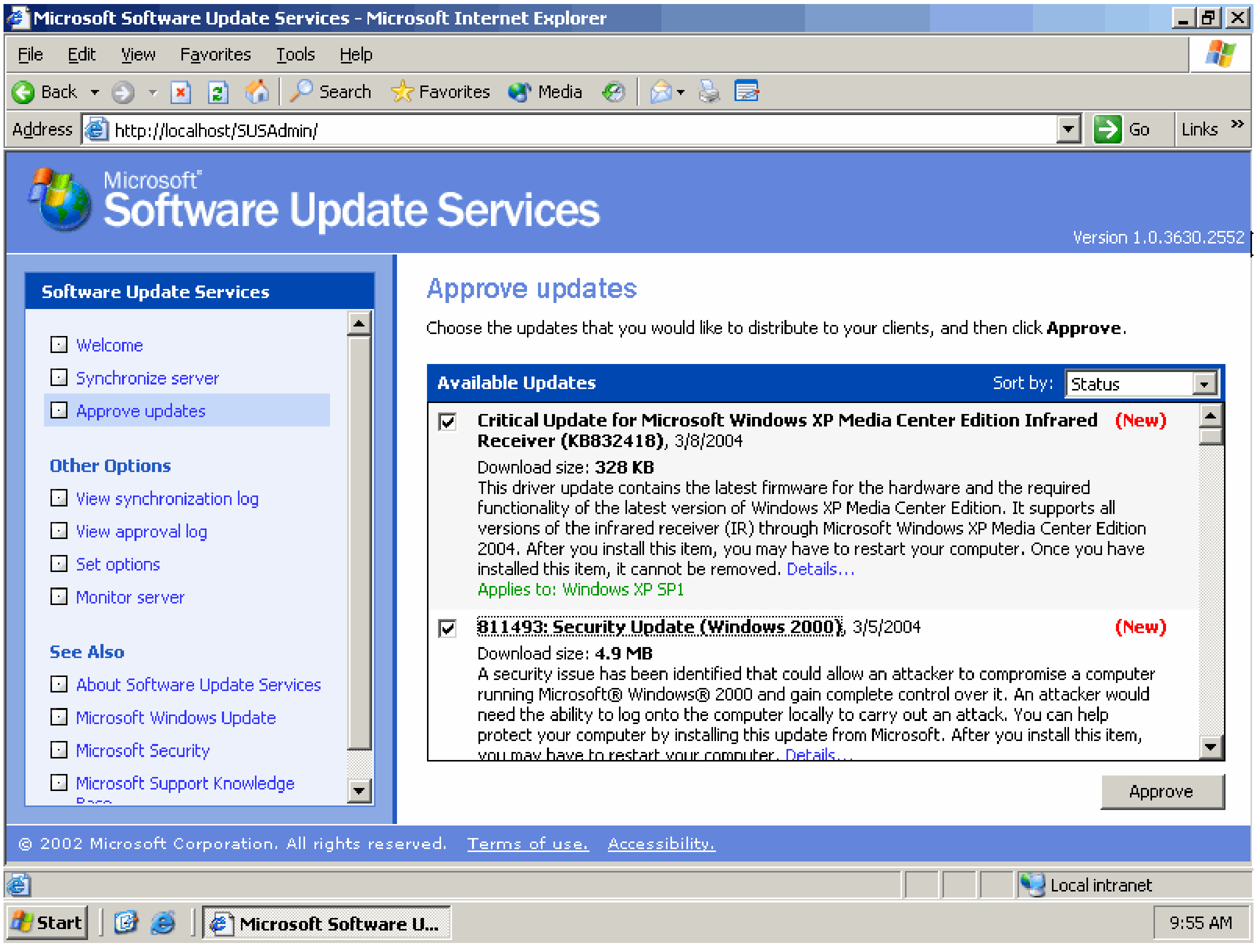
○ Maximum

Figure 2.1: Publishing a Windows Installer package by using Group Policy.

This AD feature, also called IntelliMirror, can even be used to uninstall old applications that are no longer needed. The feature isn't, however, well-suited for deploying the myriad application updates and OS patches released by Microsoft. For that, you will need to rely on the Windows Update Web site or, for internal corporate use, Microsoft's Software Update Services (SUS).

> Version 2.0 of SUS is referred to as Windows Update Services. WUS handles updates for a wide range of Microsoft applications, whereas version 1.0 of SUS focused only on Windows OS updates.

Figure 2.2 shows the SUS management console, which allows you to approve the updates provided by Microsoft before having them deployed to your client computers and servers.


Microsoft Software Update Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/SUSAdmin/
Go Links
Microsoft® Software Update Services
Version 1.0.3630.2552
Software Update Services
Welcome
Synchronize server
Approve updates
Other Options
View synchronization log
View approval log
Set options
Monitor server
See Also
About Software Update Services
Microsoft Windows Update
Microsoft Security
Microsoft Support Knowledge Base
Approve
© 2002 Microsoft Corporation. All rights reserved. Terms of use. Accessibility.
Local intranet
Start
Microsoft Software U...
9:55 AM

Approve updates

Choose the updates that you would like to distribute to your clients, and then click **Approve**.

Available Updates — Sort by: Status

☑ **Critical Update for Microsoft Windows XP Media Center Edition Infrared Receiver (KB832418)**, 3/8/2004 (New)
Download size: **328 KB**
This driver update contains the latest firmware for the hardware and the required functionality of the latest version of Windows XP Media Center Edition. It supports all versions of the infrared receiver (IR) through Microsoft Windows XP Media Center Edition 2004. After you install this item, you may have to restart your computer. Once you have installed this item, it cannot be removed. Details...
Applies to: Windows XP SP1

☑ **811493: Security Update (Windows 2000)**, 3/5/2004 (New)
Download size: **4.9 MB**
A security issue has been identified that could allow an attacker to compromise a computer running Microsoft® Windows® 2000 and gain complete control over it. An attacker would need the ability to log onto the computer locally to carry out an attack. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer. Details...

Figure 2.2: The SUS management console.

These tools are definitely valuable ways to deploy software and help maintain a stable configuration on your machines. Because these tools are centrally managed, you maintain complete control over software deployments, thus retaining more control over the final configuration of your computers.

✐ The Windows Update Web site does *not* provide you with that control; most companies practicing even traditional configuration management will use SUS/WUS or another tool to handle updates and patches because the Windows Update Web site provides no administrator control over the patches that are offered or applied.

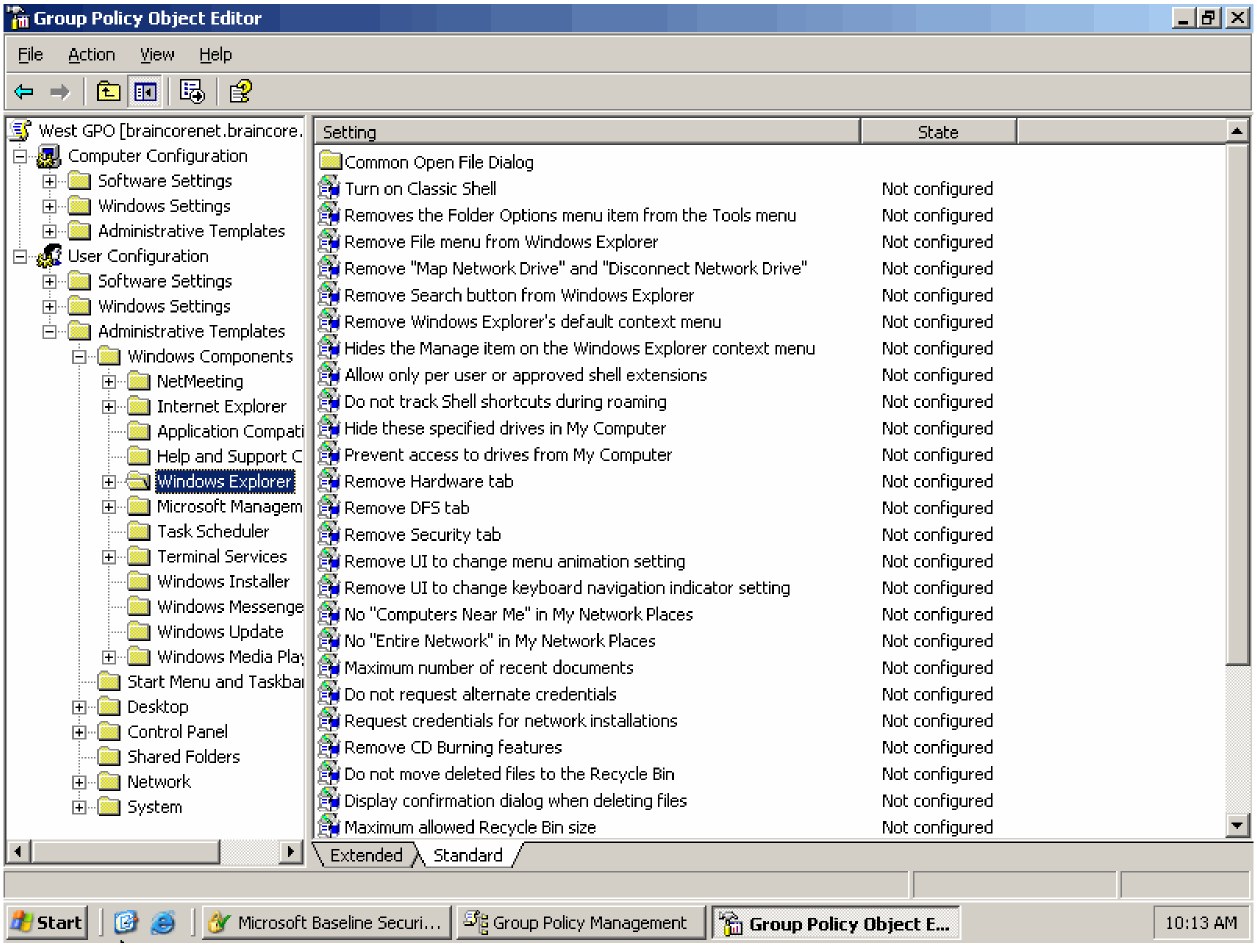
☞ If you have implemented an internal means of deploying patches, consider restricting users' access to the Windows Update Web site, preventing their ability to bypass your control over deployed patches. Win2K and later computers can be controlled by Group Policy to prevent access to Windows Update. The appropriate Group Policy setting is located under User Configuration, Administrative Templates, Windows Components, Windows Update.

Myriad other tools exist for software deployment: Microsoft Systems Management Server (SMS), Novell's ZENWorks, and others are all popular choices in different companies. All of them, however, share one common characteristic with Group Policy-based deployment and even SUS/WUS: They are essentially *push* technologies. In other words, an administrator decides what is going to be deployed, then the product or tool deploys it. Although many products (such as Group Policy, SMS, and ZENWorks) allow an administrator to target specific computers for deployments (such as only those running Windows XP Professional), not one of these tools make any analysis about the current condition of the computer before performing the deployment.

For example, suppose you have several computers in your environment running Internet Information Services (IIS). A new patch is released for IIS that you need to deploy immediately. You already know—or think you know—which computers are running IIS, so you can use a tool such as SMS or Group Policy to deploy the patch just to those computers.

But what if you've forgotten a computer? Most tools provide no means for performing a system analysis to determine which computers *need* a patch, so they rely entirely on your knowledge. The same holds true for any application, including internal corporate applications. Traditional configuration management is, at its heart, a one-way process through which you push updates to computers, and rarely deal in any kind of interaction between managed computers and your central deployment system. This lack of interaction also means that software deployment is a one-time thing: If you deploy a patch and a user subsequently uninstalls or overwrites the patch, the user's system is unpatched and most configuration management systems won't catch this configuration change.

Ongoing, continuous configuration management—the process of ensuring that machines remain configured that way you want them over a long period of time—faces a similar shortcoming. In the Windows universe, ongoing configuration management is provided primarily through Group Policy. As Figure 2.3 illustrates, Windows Group Policy offers literally hundreds of configurable settings, all of which can be applied to computers in an AD domain.



Setting	State
Common Open File Dialog	
Turn on Classic Shell	Not configured
Removes the Folder Options menu item from the Tools menu	Not configured
Remove File menu from Windows Explorer	Not configured
Remove "Map Network Drive" and "Disconnect Network Drive"	Not configured
Remove Search button from Windows Explorer	Not configured
Remove Windows Explorer's default context menu	Not configured
Hides the Manage item on the Windows Explorer context menu	Not configured
Allow only per user or approved shell extensions	Not configured
Do not track Shell shortcuts during roaming	Not configured
Hide these specified drives in My Computer	Not configured
Prevent access to drives from My Computer	Not configured
Remove Hardware tab	Not configured
Remove DFS tab	Not configured
Remove Security tab	Not configured
Remove UI to change menu animation setting	Not configured
Remove UI to change keyboard navigation indicator setting	Not configured
No "Computers Near Me" in My Network Places	Not configured
No "Entire Network" in My Network Places	Not configured
Maximum number of recent documents	Not configured
Do not request alternate credentials	Not configured
Request credentials for network installations	Not configured
Remove CD Burning features	Not configured
Do not move deleted files to the Recycle Bin	Not configured
Display confirmation dialog when deleting files	Not configured
Maximum allowed Recycle Bin size	Not configured

Figure 2.3: Group Policy settings.

An advantage of Group Policy is that it continually reapplies itself, so if users are able to somehow change one of these settings on their own, the Group Policy will eventually reapply and reassert the centrally configured settings. However, Group Policy is an entirely push mechanism. If, for example, a set of SQL Server-related Group Policy settings aren't assigned to a particular computer that is running SQL Server, that computer will remain unconfigured, which could create a potential operational problem or security vulnerability.

Group Policy's primary use by most administrators is to lock down the Windows environment so that users have minimal ability to change their computers' configurations. This lockdown might range from locking down the ability to independently install software to forcing certain configuration settings in applications such as Microsoft Office. The stated purpose for most of these lockdowns is to reduce support costs by enforcing a single known configuration on all machines and reducing the variables that a support technician must deal with. This lockdown also, of course, reduces the flexibility of the computer to meet different users' needs.

Business Analysis

Most software and configuration deployment solutions should be considered reliable only for application-level deployments and basic configurations. For example, Group Policy is an effective tool for publishing a new application to your entire user base; SMS is an excellent tool for getting new versions of an application out to computers that are running the old version. Group Policy is also a useful way to gain centralized control over selected configuration settings, such as basic Windows configuration, Internet Explorer (IE) settings, and so on.

From a continuous configuration management viewpoint, however, these tools should be more accurately regarded as conveniences. Group Policy, SMS, nor most other tools have the capability to ensure that deployed software *remains* deployed, meaning they don’t provide enforcement capabilities. For this reason, they are unreliable for deploying security patches and updates that affect operational stability. It’s simply too easy for a patch or update to be overwritten or removed, and Group Policy, SMS, and other tools won’t continuously check to ensure that removal or overwrite hasn’t occurred; they simply push the update out one time. Once the update is installed on a target machine, these tools don’t constantly check to make sure it *stays* installed.

The Myth of the Push

The problem with most of the technologies I've discussed in this section is that they're absolutely blind, push-based technologies. Even Microsoft SMS, which has powerful inventory and querying capabilities, isn't capable of performing an analysis to determine what needs to be pushed out.

For example, suppose a new patch comes in that needs to be applied to all computers running Windows XP Service Pack 2 (SP2). SMS makes this deployment possible because you can query its inventory for computers that meet that criteria, then target the query results to receive the patch. SMS will even track which systems have successfully installed the patch. What SMS *won't* do, however, is continue to monitor those systems to make sure the patch hasn't been removed or overwritten. It won't automatically deploy the patch to any *new* machines that hit the network running Windows XP SP2. You decide what gets pushed out, and SMS pushes it.

Microsoft offers a scanning tool in the Microsoft Baseline Security Analyzer (MBSA). However, the tool isn't automated; you have to run it manually (although you can have it scan multiple computers). In addition, the tool isn't integrated with SMS, meaning it can't scan a machine, realize that patch XYZ is missing, and automatically have SMS deploy patch XYZ to that machine. Microsoft is doubtless working on this level of integration, but it doesn't exist today.

"Push" refers not only to patch management but also to configuration management. For example, most administrators know that a SQL Server computer with a blank password for the built-in sa account is a bad thing. MBSA will even report this situation as a security problem in its scans. However, there is no way provided in Windows to continually, automatically check that account password and *fix the problem* if it's found.

The lesson here is that pushing anything—through WUS, SUS, SMS, or any other tool—only provides the illusion of security and stability. What's needed for true security and stability is a system that can automatically analyze systems, compare their current state with a predefined baseline, and modify their current state, if necessary, to meet that baseline.

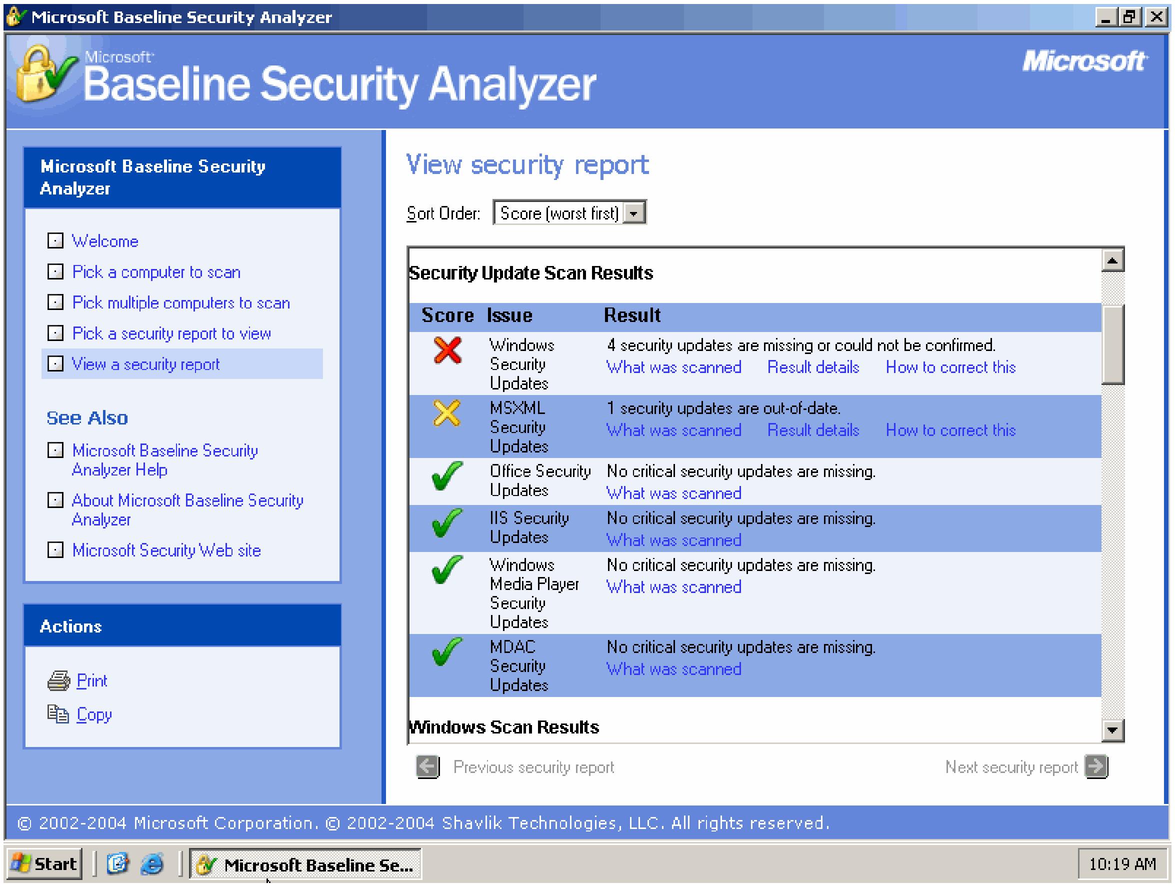
Group Policy is a convenient way to centrally deploy configuration settings that would otherwise be difficult or tedious to deploy. For example, using Group Policy to configure your users' machines with the appropriate Web proxy is an effective practice. Using Group Policy to configure computers' wireless networking settings is another excellent idea. However, Group Policy is *entirely* push-based, meaning it has no way of ensuring that its configuration settings are being enforced. Although it is difficult for a computer under Group Policy control to undo Group Policy-enforced settings, it's not impossible. For this reason, Group Policy by itself isn't a reliable way of ensuring that critical operational or security configuration settings remain properly configured over time.

Continuous Management and Enforcement

Microsoft has not, unfortunately, made great strides in continuous configuration management. Group Policy is perhaps the company's closest attempt, and Group Policy's means of implementing continuous configuration management is to simply re-push configuration settings to computers every hour or so (under most conditions). There is still no actual enforcement in terms of scanning for specific conditions or vulnerabilities and applying a configuration or patch to fix them.

Tools and Technologies

From a scanning standpoint, Microsoft does provide the MBSA, pictured in Figure 2.4.



Microsoft Baseline Security Analyzer
Microsoft
Baseline Security Analyzer
Microsoft
Microsoft Baseline Security Analyzer
Welcome
Pick a computer to scan
Pick multiple computers to scan
Pick a security report to view
View a security report
See Also
Microsoft Baseline Security Analyzer Help
About Microsoft Baseline Security Analyzer
Microsoft Security Web site
Actions
Print
Copy
Previous security report
Next security report
Start
Microsoft Baseline Se...
10:19 AM

View security report

Sort Order: Score (worst first)

Security Update Scan Results

Score	Issue	Result
✗	Windows Security Updates	4 security updates are missing or could not be confirmed. What was scanned Result details How to correct this
✗	MSXML Security Updates	1 security updates are out-of-date. What was scanned Result details How to correct this
✓	Office Security Updates	No critical security updates are missing. What was scanned
✓	IIS Security Updates	No critical security updates are missing. What was scanned
✓	Windows Media Player Security Updates	No critical security updates are missing. What was scanned
✓	MDAC Security Updates	No critical security updates are missing. What was scanned

Windows Scan Results

© 2002-2004 Microsoft Corporation. © 2002-2004 Shavlik Technologies, LLC. All rights reserved.

Figure 2.4: The MBSA console.

MBSA uses a Microsoft-provided, XML-based database of all Microsoft-documented security vulnerabilities, including those that are the result of an improper configuration (rather than simply the result of a software bug). MBSA scans for these vulnerabilities and, as shown, highlights ones that are present on the scanned computer. MBSA 1.2 supports Windows, Microsoft Office, SQL Server, Exchange Server, IIS, MSXML, Microsoft Data Access Components (MDAC), Host Integration Server, Content Management Server, BizTalk Server, and a host of other Microsoft products.

Although you can use MBSA to scan systems—even several at a time—it doesn't do anything about the problems it finds. As Figure 2.4 shows, MBSA has found four missing or unconfirmable security updates in Windows. This information is all that is provided by MBSA; the tool doesn't offer the functionality to actually deploy those updates for you. It also doesn't maintain an ongoing database, which means each time you want to see a report, you will need to run a fresh scan. In addition, MBSA is entirely central, not agent-based; meaning the process of scanning your entire network can take a *long* time and be very resource-intensive.

Other third-party tools do a better job of implementing continuous configuration management. For example, ConfigureSoft's Enterprise Configuration Manager uses an agent-based architecture to continually scan managed computers and report configuration information to a centralized database. Responses can be defined for various conditions, and the company's Security Update Manager can be used to respond to security vulnerabilities by deploying the appropriate updates or modifying the appropriate configuration.

TripWire for Servers also provides continual monitoring of server computers, with full reports about any changes made to those computers. This capability can, for example, notify you when a security patch is removed or overwritten; you can't, however, configure it to automatically redeploy the update, which makes it somewhat less automated.

📖 I'll discuss these and other continuous configuration management tools in Chapter 4.

Business Analysis

Continuous configuration management is truly the best way to maintain security and operational stability in your environment. You should be able to define a set of practical security policies, which describe various software and configuration conditions that you desire for your managed computers. A continuous configuration management solution—whether an automated one or one you build yourself—should continuously analyze computers for compliance with your standard and report any deviations. Ideally, the solution should also repair those deviations, taking some defined response to reconfigure the machine or deploy an update.

The beauty of proper continuous configuration management is that *it doesn't matter what the baseline is or how computers are currently configured*. The baseline is actually an abstract ideal, which you define in policies; the solution scans *every* computer for compliance, regardless of the computer's original provisioning or subsequent deployments. A proper continuous configuration management system will allow computers completely outside your configuration management process to be connected to your network, scanned, and brought into compliance.

Continuous Configuration Management: The Ever-Fresh Consultant

One of the reasons companies hire consultants is to get a "fresh view" of their environment, their practices, and their problems. Working in an environment every day makes it too easy to get caught up in the day-to-day grind, the history behind problems and how they came to be, and other baggage. Consultants can come in knowing nothing and help make sweeping changes that don't consider how things came to be in the present.

Continuous configuration management should work the same way. Yes, your computers might all have started on Windows NT Workstation 4.0 and been gradually upgraded to Windows XP and that is why there are some holes in permissions and missing security patches. Continuous configuration management doesn't consider this history: It looks at every system with a fresh view, every time, comparing it with what you've defined as an acceptable set of configuration parameters and patch levels. The continuous configuration management solution won't get caught up in how secure the machine's initial image was nor how often you've been deploying patches, because every machine is a brand-new, fresh analysis.

This tool is invaluable. For example, suppose you're the perfect administrator and you deploy every new patch, the day it is released. Your company deploys a new sales application, and you follow the instructions for deploying it to the letter. You continue to keep on top of patch deployment, even forgoing vacations to ensure the stability and security of your network environment.

What you don't know is that the application overwrote a key file—let's say rpc.dll, the file that the Blaster worm attacked—and replaced it with an older version. Sure, that's against all rules of software deployment politeness, but it *does* happen. So without evening realizing it, your machines are all vulnerable. None of your tools knew about it, because they're all telling you that you deployed the Blaster patch months ago. A continuous configuration management solution would catch this shortcoming, because it looks at each system fresh. The minute you installed that sales application, your continuous configuration management solution would have started barking about the outdated version of rpc.dll, and might have even triggered a redeployment of the patch, keeping you stable and secure.

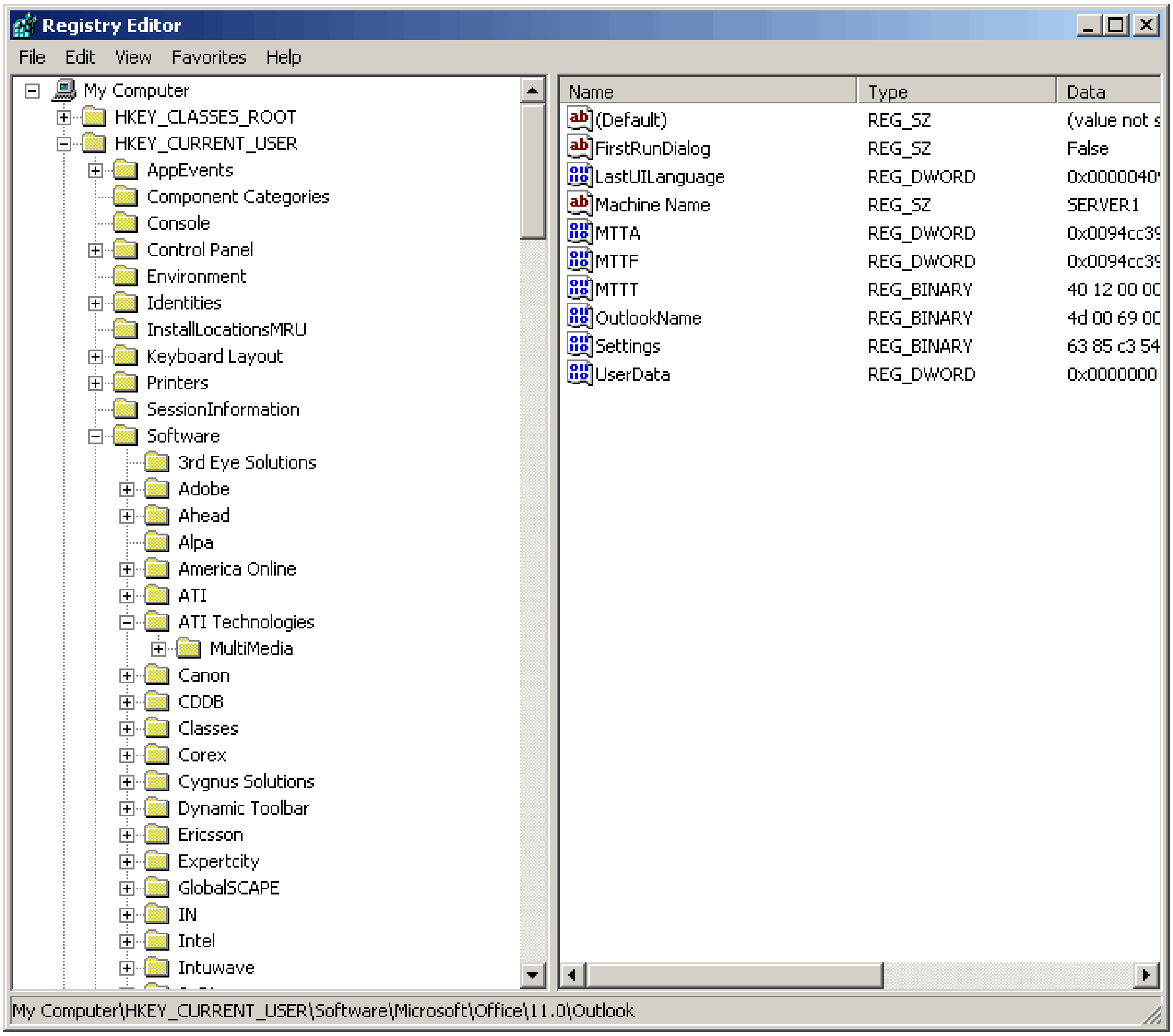
Proper continuous configuration management means no longer worrying about configuration drift. In other words, you don't need to worry about what your users might have done to their computers because they were originally provisioned—the continuous configuration management process re-evaluates every computer on an ongoing basis for compliance with a desired *end state* no matter what its starting state or current condition. Regardless of which tools and technologies you select for provisioning and software/configuration deployment, a robust continuous configuration management system is where your desired end-state configuration is actively enforced.

Underlying Technologies

Continuous configuration management systems rely on Windows' own underlying technologies to retrieve information, make configuration changes, and so forth. Understanding these underlying technologies is important to understanding the limitations of some configuration management solutions. They also give you an appreciation for the complexity of an effective continuous configuration management system, which must be able to work with a dozen or more core services contained within Windows.

Windows Registry

Microsoft describes the registry as the central configuration database for the Windows OS. Although this definition is not wholly accurate, the registry is certainly an important repository for configuration settings. Figure 2.5 shows the Windows Registry Editor, a graphical user interface (GUI) used to manually access registry settings.



- My Computer
  - HKEY_CLASSES_ROOT
  - HKEY_CURRENT_USER
    - AppEvents
    - Component Categories
    - Console
    - Control Panel
    - Environment
    - Identities
    - InstallLocationsMRU
    - Keyboard Layout
    - Printers
    - SessionInformation
    - Software
      - 3rd Eye Solutions
      - Adobe
      - Ahead
      - Alpa
      - America Online
      - ATI
      - ATI Technologies
        - MultiMedia
      - Canon
      - CDDB
      - Classes
      - Corex
      - Cygnus Solutions
      - Dynamic Toolbar
      - Ericsson
      - Expertcity
      - GlobalSCAPE
      - IN
      - Intel
      - Intuwave

Name	Type	Data
(Default)	REG_SZ	(value not s
FirstRunDialog	REG_SZ	False
LastUILanguage	REG_DWORD	0x0000040
Machine Name	REG_SZ	SERVER1
MTTA	REG_DWORD	0x0094cc3
MTTF	REG_DWORD	0x0094cc3
MTTT	REG_BINARY	40 12 00 0
OutlookName	REG_BINARY	4d 00 69 0
Settings	REG_BINARY	63 85 c3 54
UserData	REG_DWORD	0x0000000

My Computer\HKEY_CURRENT_USER\Software\Microsoft\Office\11.0\Outlook

Figure 2.5: The Windows Registry Editor.

The registry is divided into five major sections, or *hives*:

- HKEY_CLASSES_ROOT contains all of the classes that have been registered on the local computer. These classes define file type associations as well as COM and COM+ components that have been installed on the computer.
- HKEY_CURRENT_USER contains configuration information specific to the currently logged-on user.
- HKEY_LOCAL_MACHINE contains global configuration information that applies to the entire computer.
- HKEY_USERS contains all users' configuration information; HKEY_CURRENT_USER is actually a subset of HKEY_USERS.
- HKEY_CURRENT_CONFIG contains several configuration settings for the system itself; HKEY_CURRENT_CONFIG is actually a subset of HKEY_LOCAL_MACHINE.

Windows provides well-documented application programming interfaces (APIs) for accessing the registry, making it easy to retrieve configuration information or even to make changes. However, the sheer size of the registry, which can often exceed several megabytes' worth of information, makes retrieving registry data in bulk a complex task. Most efficient configuration management systems don't even try to retrieve the entire registry; instead, their manufacturers program them to retrieve only sections of the registry that contain relevant configuration information (which might still be thousands of configuration settings).

Windows Management Instrumentation

Windows Management Instrumentation (WMI) is a major new management component developed by Microsoft. WMI is based on an industry-standard Common Information Model (CIM), developed by the industry's Desktop Management Task Force (DMTF). The CIM defines a model for describing computer configuration information; WMI follows this model.

WMI is a relatively mature technology, having first been offered for Windows NT 4.0. It continues to grow with each new Windows release, offering access to additional configuration information. It can be readily access via a number of well-documented APIs. For example, the VBScript that Listing 2.1 shows queries a great deal of information about the current OS.

```
On Error Resume Next
Dim strComputer
Dim objWMIService
Dim propValue
Dim colItems

strComputer = “.”
Set objWMIService = GetObject(“winmgmts:\\” & _
 strComputer & “\root\cimv2”)
Set colItems = objWMIService.ExecQuery( _
 “Select * from Win32_OperatingSystem”,,48)
For Each objItem in colItems
      WScript.Echo “BootDevice: “ & objItem.BootDevice
      WScript.Echo “BuildNumber: “ & objItem.BuildNumber
      WScript.Echo “BuildType: “ & objItem.BuildType
      WScript.Echo “Caption: “ & objItem.Caption
      WScript.Echo “CodeSet: “ & objItem.CodeSet
      WScript.Echo “CountryCode: “ & objItem.CountryCode
      WScript.Echo “CreationClassName: “ & _
       objItem.CreationClassName
      WScript.Echo “CSCreationClassName: “ & _
       objItem.CSCreationClassName
      WScript.Echo “CSDVersion: “ & objItem.CSDVersion
      WScript.Echo “CSName: “ & objItem.CSName
      WScript.Echo “CurrentTimeZone: “ & objItem.CurrentTimeZone
      WScript.Echo “Debug: “ & objItem.Debug
      WScript.Echo “Description: “ & objItem.Description
      WScript.Echo “Distributed: “ & objItem.Distributed
      WScript.Echo “EncryptionLevel: “ & objItem.EncryptionLevel
      WScript.Echo “ForegroundApplicationBoost: “ & _
       objItem.ForegroundApplicationBoost
      WScript.Echo “FreePhysicalMemory: “ & _
       objItem.FreePhysicalMemory
      WScript.Echo “FreeSpaceInPagingFiles: “ & _
       objItem.FreeSpaceInPagingFiles
      WScript.Echo “FreeVirtualMemory: “ & _
       objItem.FreeVirtualMemory
      WScript.Echo “InstallDate: “ & objItem.InstallDate
      WScript.Echo “LargeSystemCache: “ & _
       objItem.LargeSystemCache
      WScript.Echo “LastBootUpTime: “ & objItem.LastBootUpTime
      WScript.Echo “LocalDateTime: “ & objItem.LocalDateTime
      WScript.Echo “Locale: “ & objItem.Locale
      WScript.Echo “Manufacturer: “ & objItem.Manufacturer
      WScript.Echo “MaxNumberOfProcesses: “ & _
       objItem.MaxNumberOfProcesses
      WScript.Echo “MaxProcessMemorySize: “ & _
       objItem.MaxProcessMemorySize
      WScript.Echo “Name: “ & objItem.Name
      WScript.Echo “NumberOfLicensedUsers: “ & _
       objItem.NumberOfLicensedUsers
      WScript.Echo “NumberOfProcesses: “ & _
       objItem.NumberOfProcesses
      WScript.Echo “NumberOfUsers: “ & objItem.NumberOfUsers
      WScript.Echo “Organization: “ & objItem.Organization
      WScript.Echo “OSLanguage: “ & objItem.OSLanguage
```

```
       WScript.Echo “OSProductSuite: “ & objItem.OSProductSuite
       WScript.Echo “OSType: “ & objItem.OSType
       WScript.Echo “OtherTypeDescription: “ & _
        objItem.OtherTypeDescription
       WScript.Echo “PlusProductID: “ & objItem.PlusProductID
       WScript.Echo “PlusVersionNumber: “ & _
        objItem.PlusVersionNumber
       WScript.Echo “Primary: “ & objItem.Primary
       WScript.Echo “ProductType: “ & objItem.ProductType
       WScript.Echo “QuantumLength: “ & objItem.QuantumLength
       WScript.Echo “QuantumType: “ & objItem.QuantumType
       WScript.Echo “RegisteredUser: “ & objItem.RegisteredUser
       WScript.Echo “SerialNumber: “ & objItem.SerialNumber
       WScript.Echo “ServicePackMajorVersion: “ & _
        objItem.ServicePackMajorVersion
       WScript.Echo “ServicePackMinorVersion: “ & _
        objItem.ServicePackMinorVersion
       WScript.Echo “SizeStoredInPagingFiles: “ & _
        objItem.SizeStoredInPagingFiles
       WScript.Echo “Status: “ & objItem.Status
       WScript.Echo “SuiteMask: “ & objItem.SuiteMask
       WScript.Echo “SystemDevice: “ & objItem.SystemDevice
       WScript.Echo “SystemDirectory: “ & objItem.SystemDirectory
       WScript.Echo “SystemDrive: “ & objItem.SystemDrive
       WScript.Echo “TotalSwapSpaceSize: “ & _
        objItem.TotalSwapSpaceSize
       WScript.Echo “TotalVirtualMemorySize: “ & _
        objItem.TotalVirtualMemorySize
       WScript.Echo “TotalVisibleMemorySize: “ & _
        objItem.TotalVisibleMemorySize
       WScript.Echo “Version: “ & objItem.Version
       WScript.Echo “WindowsDirectory: “ & _
        objItem.WindowsDirectory
Next
```

Listing 2.2: VBScript that queries OS configuration information.

A difficulty in using WMI, however, is that there are literally hundreds and hundreds of classes to choose from—the sample script that Listing 2.2 shows is for just one class. As you can see, some classes have dozens of properties. That makes it difficult to create your own inventorying system using WMI, simply because there is so much information. That said, WMI isn’t the end-all-be-all of management information. Although Microsoft’s direction seems to be to make WMI the ultimate tool for acquiring management information, today’s WMI falls far short and simply ignores several major areas of the OS. Still, today’s WMI provides a great deal of information, and many configuration management solutions take advantage of it.

Security Accounts Manager and AD

The Security Accounts Manager (SAM) is an integral part of every NT-based OS, including Win2K, Windows XP, and Windows Server 2003 (WS2K3). The SAM contains all local user and group accounts, and on an NT domain controller, also contains all domain user, group, and computer accounts. In a Win2K domain, a separate service named AD contains domain-related security principles.

The SAM plays a critical role in configuration management. For example, Windows' NTFS assigns Access Control Lists (ACLs) to each file and folder on the computer's hard drive. These ACLs consist of one or more Access Control Entries (ACEs), each of which grants or denies specific permissions to users and groups. The ACEs actually store user and group Security Identifiers (SIDs), not names; in order for a configuration management solution to present meaningful information, it must access the SAM (or AD) to translate SIDs into actual names. The SAM also contains a few important local security configuration settings; AD contains an enormous amount of additional configuration information, including ACEs for objects contained within AD.

You can access both the SAM and AD via well-documented APIs. However, this set of APIs is now the *third* set of that a configuration management solution must accommodate, including the registry and WMI, which I discussed previously. The complexity of a Windows-based configuration management system is starting to become clearer.

Everything Else

Complex as they are, and containing as much information as they do, the registry, WMI, and SAM/AD surely represent the bulk of the configuration information that a continuous configuration management solution would need to access, right? Hardly. Windows has evolved over time, and most new features have brought their own management APIs. Although technologies such as WMI are *beginning* to consolidate those APIs into a single interface, WMI still has a long way to go. Consider the following features that still require access via their own APIs:

- Dynamic Host Configuration Protocol (DHCP) servers
- Aspects of NTFS auditing and permissions
- Aspects of hotfix/patch management
- System backup and restore information

In addition, add-on applications—including Microsoft's own—are just beginning to provide WMI classes for management. Microsoft SQL Server uses an API called SQL Server Distributed Management Objects (SQL-DMO), and Exchange Server 2003 was the first version of Exchange to provide significant WMI support—in the past, it used several different APIs to expose various aspects of itself to automated management solutions.

The point of all this information is that continuous configuration management solutions must do a lot more than simply copy a file or two or query a couple of WMI classes in order to obtain the configuration information they need to do their jobs. Windows is a complex, large OS with hundreds of thousands of configuration settings, making a continuous configuration management solutions' job a complex one indeed.

Case Study: More Reliable Security with Continuous Configuration Management

I recently finished working with a network solutions firm that hired me to help get their Windows desktop security under control. They had already made the move to Win2K Professional and Windows XP Professional, but the nature of their business made controlling security difficult. The company manages networks for large clients who choose to outsource that function rather than handling it internally; company field engineers were provided with laptops that they used to work on-site for company customers.

The engineers frequently needed to install software utilities that would allow them to configure and troubleshoot customers' networks. Unfortunately, many of these utilities would overwrite company-deployed OS updates, patches, and so forth. In addition, the utilities often required that the engineers have a high level of privilege on their machines, often requiring the engineers to be local administrators. The engineers, being tech geeks, took advantage of their administrative rights to install games, personal software, and more than a few viruses—purely by accident—on their laptops. The company's solution was to take away administrative rights and centrally deploy all the tools the engineers needed; thus, reducing the ability of the engineers to cause their laptops' configurations to drift from the company-approved standard.

The lockdown solution was pretty much a disaster. Engineers would often need new utilities while not connected to the company network, effectively making their jobs more difficult if not downright impossible. I was called in to help engineer a new software distribution system that could work over dial-up connections, allowing engineers to dial in and retrieve software even while at customer locations.

I suggested, however, that trying to maintain a rigid baseline was simply not going to work—there would always be problems simply as a result of the environment in which the engineers worked. Instead, the company needed to find a way to constantly analyze engineers' laptops whenever the engineers connected to the company network (which was at least a couple of times a week), and to enforce a centrally defined set of security policies. We sat down and created a starting set of policies, which defined minimum patch levels for certain software (including Windows), certain configuration options (such as the Internet Connection Firewall configuration), and other configuration settings. The company began to evaluate continuous configuration management solutions to find one that met their needs.

The lesson is one that I've mentioned several times in this chapter: *change happens*. There is almost nothing you can do to eliminate change without simply shutting off your users' computers completely. Accept that change will occur and accept that it will occur without your knowledge or control. Instead of locking down systems to try and stop change, create a system that acknowledges change. This idea is really what continuous configuration management is all about: Acknowledging change and maintaining a certain set of configuration standards even in the face of change.

Summary

In this chapter, I've tried to give you an idea of what continuous configuration management is all about, and how it differs from traditional, one-way configuration management. I've shown how businesses require their technology products to remain flexible, and how only continuous configuration management, with its constant interaction between central control and managed machines, provides that flexibility while ensuring certain centrally defined standards.

I've also introduced you to some of the underlying technologies that make configuration management possible. This introduction will come in handy in the next chapter, where I show you how you can create pieces of a configuration management solution on your own by using freely available tools and technologies. However, I've also shown you how complex configuration management can be, so another goal of the next chapter will be to highlight areas where "do it yourself" simply won't provide the functionality you really need for an effective continuous configuration management solution.