

realtimepublishers.com<sup>tm</sup>

*The Administrator  
Shortcut Guide<sup>tm</sup> To*



**Configuration  
Management**

*for the Windows Enterprise*



*Don Jones*

---

Chapter 3: Roll-Your-Own Configuration Management .....	36
Tools for Configuration Management .....	36
Group Policy .....	37
Microsoft Baseline Security Analyzer .....	40
SUS and WUS.....	43
Security Templates.....	45
Resource Kit Tools .....	46
Scripting Your Own Tools.....	47
Closing the Gaps in Configuration Management.....	49
Scalability .....	50
Dynamically Grouping Computers .....	51
Periodic Deployments.....	52
Continuous Management and Enforcement.....	53
Centralization.....	55
Analysis and Reporting Tools.....	55
Ease of Use .....	55
Summary .....	56

## Copyright Statement

© 2004 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

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

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

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

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

If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at [info@realtimedpublishers.com](mailto:info@realtimedpublishers.com).

## Chapter 3: Roll-Your-Own Configuration Management

How difficult is it to create a configuration management system? As I've discussed in the previous two chapters, a *lot* of information comes into play for configuration management. A lot of that information, however, is accessible through well-defined interfaces such as Windows Management Instrumentation (WMI), the Windows registry, and so forth—so why not simply create your own configuration management system? Microsoft also provides many freely available tools and utilities to form the building blocks of a configuration management system, meaning you should be able to pull these together and create a fairly effective solution.

In today's "do more with less" economy, it's certainly tempting to just roll your own configuration management system. You won't need to buy any tools, and, with a little work, you could have all of the pieces in place. In this chapter, I'll explore the limits of what you can do entirely on your own by using entirely free tools and technologies to create a configuration management system. I'll also look at the business ramifications of doing so, and offer some insight about the gap that will exist between your homegrown solution and the available commercial offerings.

### Tools for Configuration Management

It's amazing that so many tools and technologies are available built-in to Windows or as free additions to Windows that can help with configuration management. It's even more amazing that so few administrators know about many of these tools, how they work, and how they can be used to create a more stable and secure environment.

Before we launch into the details of these tools, however, let's quickly review the goals of a configuration management solution. As you're reading through the descriptions of the various tools and technologies, evaluate them to see which of these goals they help achieve:

- **Inventory**—A good configuration management system starts with some kind of inventory, which collects information from target systems and evaluates that information as part of the configuration management process.
- **Baselines**—Ideally, a configuration management system should allow you to define baselines (or templates, rule groups, or whatever you want to call them) that specify your desired configuration for various types of systems (clients, IIS servers, SQL Server computers, and so forth).

- **Analysis**—As I discussed in the previous chapter, simply pushing configuration information (including patches) out to target systems is insufficient; a good configuration management system needs to constantly evaluate systems with respect to your defined baselines, and continually reconfigure systems to meet those baselines. Called *continuous configuration management*, this technique ensures that your desired configuration remains in effect, or that you are at least notified of any deviations.
- **Deployment**—A configuration management system needs some means of deploying both software (especially patches) and configuration settings to target machines. This factor represents the active part of configuration management—going a step beyond simple analysis to actually delivering configurations to target machines.

At the end of each tool or technology discussion, I'll assess how that tool or technology helps achieve these goals.

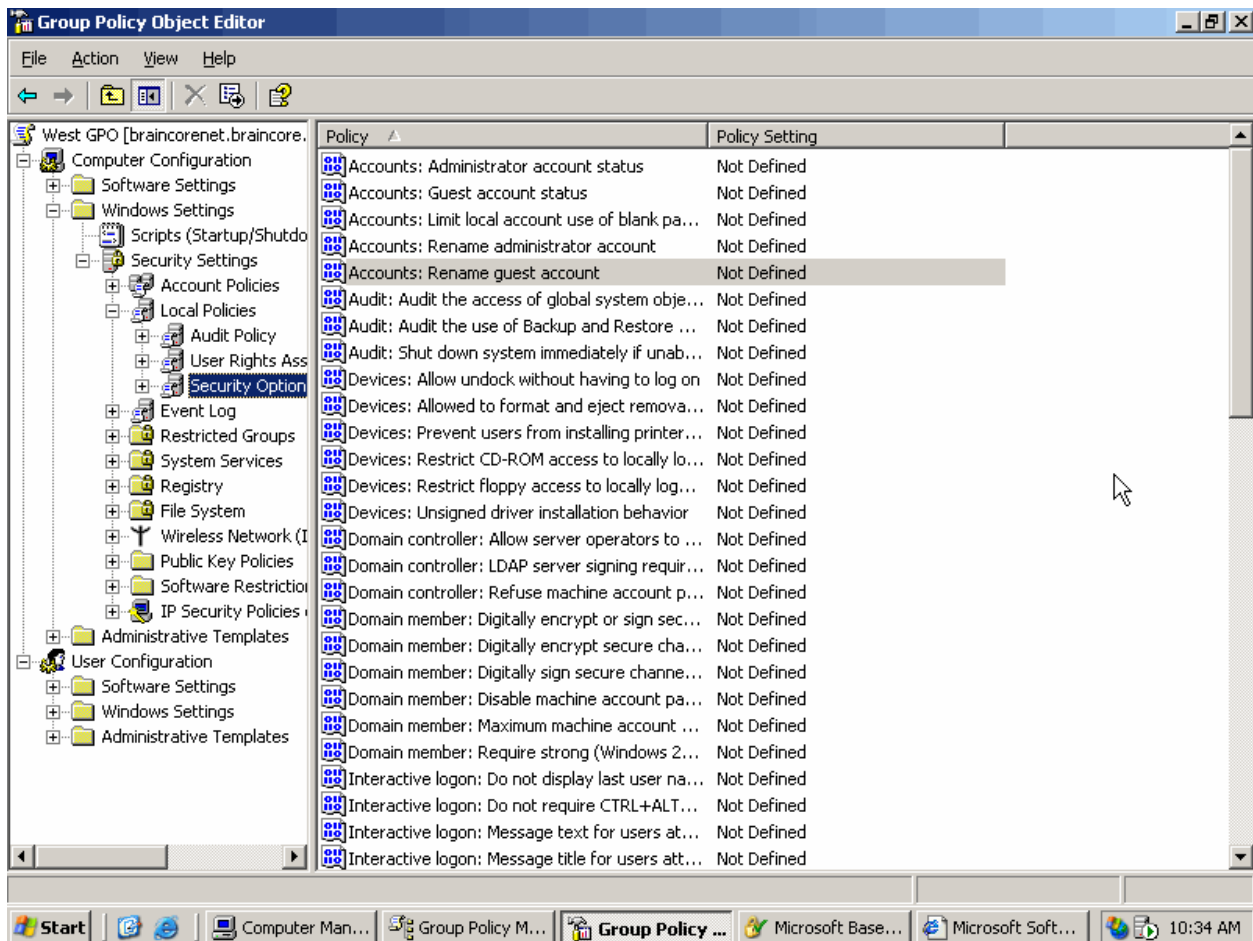
#### Tools that Are Not Free

There are, of course, several tools that do a much better job of configuration management than what I'm going to describe: Microsoft Systems Management Server (SMS), ConfigureSoft Enterprise Configuration Manager (ECM), and more. However, the point of this chapter (at least, this section of it) is to explore what you can accomplish entirely on your own. Later, however, I'll cover some of the commercial alternatives and explain what they can offer that a homegrown solution often can't.

### Group Policy

Win2K introduced Group Policy, the successor of Microsoft's earlier System Policy technologies and currently the most central and reliable way to configure Windows computers in a domain. Group Policy is essentially a centralized, GUI-based way of pushing configuration values and settings out to machines. Group Policy objects (GPOs) are files that are stored on a domain controller, and linked to a domain, site, or organizational unit (OU); GPOs are pushed to server and client computers contained within those domains, sites, or OUs, and the computers apply the GPOs. Computers re-check GPOs on a regular basis (every 30 to 90 minutes, approximately) to re-apply any GPOs that have changed.

As Figure 3.1 shows, GPOs can contain a staggering number of configuration values—literally hundreds for Windows Server 2003 (WS2K3). GPOs themselves are divided into two categories: Policy settings that affect an entire computer and all of its users fall under the Computer Configuration category, and policy settings that affect a specific user are part of User Configuration. Multiple GPOs can apply to a single user and computer; in the event of conflicting policy settings, the last GPO applied takes effect. GPOs are applied in a predictable order, beginning with the domain, then the site, and then OU-based GPOs starting with parent OUs and working down to the child OU that actually contains the user or computer in question.



**Figure 3.1:** Group Policy offers a large variety of configuration values that can be centrally controlled.

GPOs represent a powerful, flexible tool for configuration management. GPOs can, in fact, be extended to include even *more* policy settings through the addition of administrative templates (ADM files) that define the new policy settings and their possible values. For example, applications such as Microsoft Office can be centrally configured through GPOs simply by importing the appropriate manufacturer-supplied administrative templates. The administrative template format is also well-defined, allowing savvy administrators to create their own ADM files to control in-house applications and other values.

GPOs also encapsulate Microsoft's IntelliMirror technology, which can be used for software deployment (provided, in most cases, that the software is packaged in a Windows Installer—MSI—package).

How does Group Policy stack up as a configuration management tool? Let's examine Group Policy in relation to the four goals established earlier:

- **Inventory**—Group Policy doesn't perform any collection of information from client computers. Although Group Policy will act to continually keep target systems configured as desired, you won't receive any notifications for out-of-compliance systems; they will simply be reconfigured (on a regular interval or at system restart, depending on the specific policy) to use the Group Policy–defined settings.
- **Baselines**—GPOs themselves act as baselines, containing the desired configuration.
- **Analysis**—GPOs offer a form of continuous configuration management through constant redeployment and reapplication; however, they have no analysis capabilities. In other words, *you* need to make sure that GPOs are applying to the appropriate computers. For example, suppose you create a GPO meant to deploy a patch to servers running IIS. If you forget to link the GPO to the right domain, site, or OU to actually hit all of your IIS servers, then not all of your IIS servers will be patched.

Further, IntelliMirror itself doesn't re-deploy. Thus, if you use IntelliMirror to deploy a patch (many patches aren't packaged in an MSI file that IntelliMirror can use), IntelliMirror will do so *once*. It doesn't analyze a system to determine whether a patch has been removed or overwritten.

- **Deployment**—As I've mentioned, software deployment of a limited type is available through the IntelliMirror features of Group Policy. However, this aspect of Group Policy cannot be regarded as continuous configuration management because the deployments take place only one time.

Thus, GPOs seem to solve a portion of the overall configuration management problem, at least with regard to deploying configuration settings. But GPOs are far from a complete solution, even for configuration settings: A vast number of configuration settings are not available through GPOs. For example, GPOs can neither alert you to a blank SQL Server sa account password nor do anything about it; GPOs are limited primarily to registry- and domain-based configuration settings. A GPO can't close an open mail relay on an Exchange Server computer nor even tell you that one exists. Thus, GPOs represent a limited configuration management solution with definite boundaries of functionality.

## Microsoft Baseline Security Analyzer

The Microsoft Baseline Security Analyzer (MBSA) is a graphical tool (that can also be executed from the command-line) designed to scan computers for missing or out-of-date security updates, as well as for critical, security-related configuration settings. As Figure 3.2 shows, MBSA can be used to scan an entire domain of computers, or entire IP address ranges, making it a reasonable tool for automating a portion of configuration management.

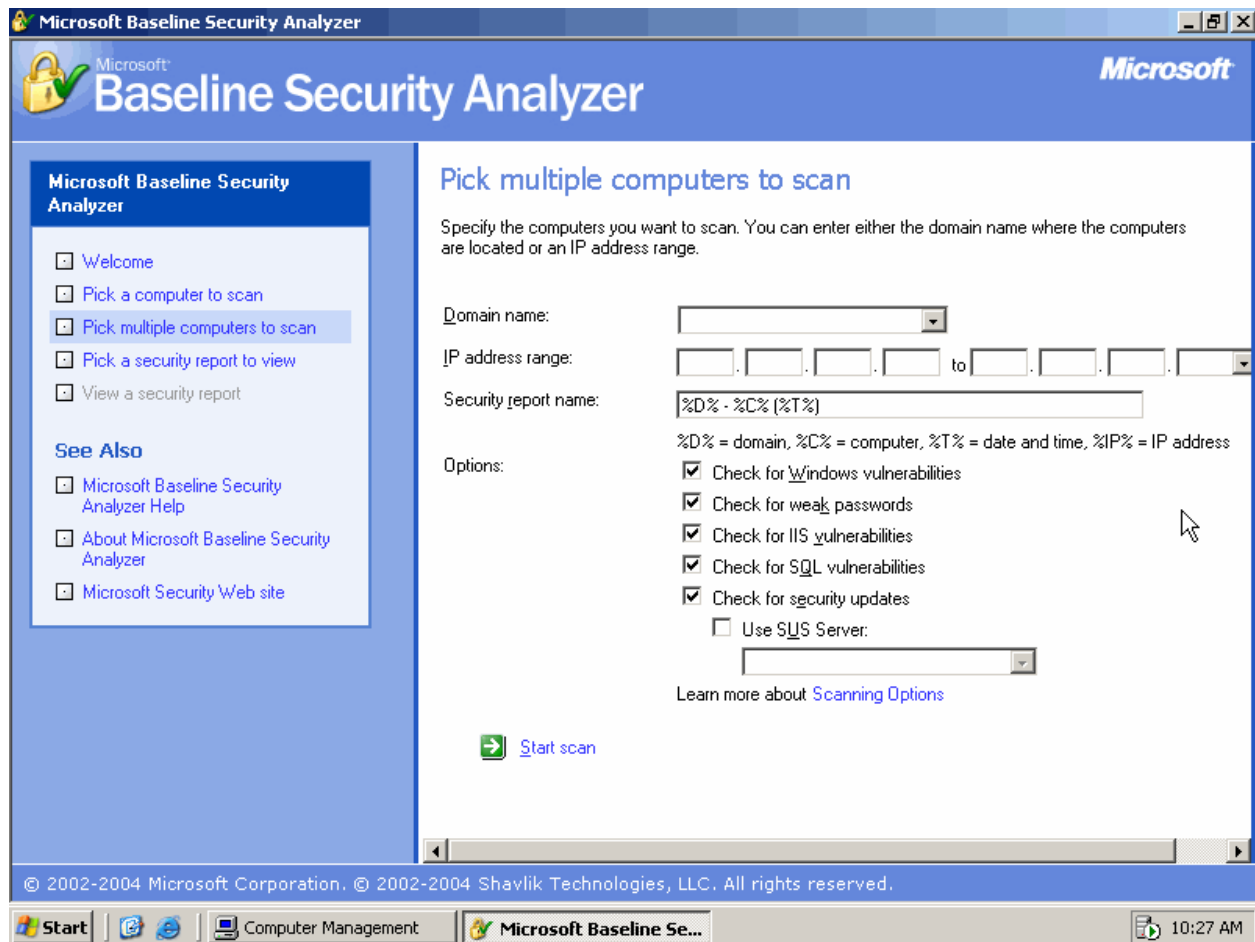


Figure 3.2: Using MBSA to scan multiple computers at once.

MBSA doesn't just scan for patches; it also scans for weak passwords, known configuration vulnerabilities (which an administrator can correct without applying a patch), and more. Figure 3.3 shows the results of a scan, which can be output to a file (a useful feature when you're scanning multiple computers).



**Microsoft Baseline Security Analyzer**

View security report

Sort Order:

**Computer name:** BRAINCORE\BRAINCORENET  
**IP address:** 192.168.131.65  
**Security report name:** BRAINCORE - BRAINCORENET (6-14-2004 10:34 AM)  
**Scan date:** 6/14/2004 10:34 AM  
**Scanned with MBSA version:** 1.2.3316.1  
**Security update database version:** 2004.6.8.0  
**Office update database version:** 11.0.0.6608  
**Security assessment:** Severe Risk (One or more critical checks failed.)

**Security Update Scan Results**

Score	Issue	Result
✖	Windows Security Updates	6 security updates are missing or could not be confirmed. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>
✖	SQL Server/MSDE Security Updates	Instance (default): 1 critical security updates are missing. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>
✘	MSXML Security Updates	1 security updates are out-of-date. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>

Previous security report      Next security report

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

Start | Computer Management | Microsoft Baseline Se... | Microsoft Software Upda... | 10:34 AM

**Figure 3.3: MBSA scan results.**

MBSA 1.2 implements a new vulnerability scanning engine, one that is positioned to become Microsoft's standard scanning engine for future versions of MBSA, Software Update Services (SUS), SMS, and more. In the past, scanning the same Windows computer with different Microsoft tools could produce remarkably different results; by standardizing on one scanning engine, Microsoft hopes to produce more consistent and accurate results.


How well does MBSA fit into a configuration management system?

- **Inventory**—MBSA collects information about the systems it scans. It does not, however, collect this information over time or persist it in a database. Scans occur in real-time, which can create a significant performance impact when scanning a large number of computers at once. Additionally, because the information isn't persisted, future scans will take the same amount of time, because MBSA must start the scan from scratch, rather than simply starting with the previous scan's data and looking for the changes (or *delta*) since the last scan.
- **Baselines**—MBSA does, as its name implies, use a baseline. However, it's a baseline defined primarily by Microsoft, and not by you. If your server baseline requires IIS to be uninstalled, for example, MBSA won't report on servers that have IIS installed; it will simply report on IIS installations that are missing security patches. This inability to tell MBSA what *you* (rather than Microsoft) consider to be secure severely limits the value MBSA can have in many environments. MBSA analyzes for compliance with Microsoft's definition of secure.
- **Analysis**—MBSA performs a simple analysis of your systems, comparing them with Microsoft's baseline for security. However, the analysis stops at the report produced by MBSA; there is no feedback into something like Group Policy so that misconfigured systems can be automatically corrected. If MBSA reports that several computers are missing a patch, you're on your own to deploy that patch to those computers; if MBSA reports that several computers are configured in a less-than-secure fashion, you'll need to come up with your own means for correcting that configuration.
- **Deployment**—MBSA is a read-only tool, meaning it only scans and reports; there's no built-in corrective capability or ability to deploy patches based on what MBSA finds.


MBSA provides valuable insight into how your Windows systems are configured. In fact, it's the only freely available scanning and analysis tool from Microsoft; the rest of Microsoft's configuration management tools (such as Windows Update Services—WUS—and Group Policy) are not only push-only but they provide no feedback to tell you which computers have received an update or configuration setting. Only MBSA can touch computers directly and tell you what vulnerabilities exist. Unfortunately, MBSA only cares about what *Microsoft* considers to be a vulnerability. Also, because MBSA scans only in real-time, it can be difficult to use it against a large number of computers on a regular basis. Although scanning one computer only takes a minute or two, scanning a couple thousand client computers will create significant network traffic and likely take several hours (if not days).

## SUS and WUS

Microsoft's Windows Update Web site, originally introduced back in the Win9x days, automatically scans a system to determine what patches are needed, then offers the ability to download and install those patches. SUS and its successor WUS provide a centrally administered means of controlling user access to Windows Update.

 WUS is version 2.0 of SUS, with a new name and a greatly expanded feature set. As of this writing, WUS is expected to release sometime in the latter half of 2004 or early 2005.

Essentially, SUS/WUS downloads every available update from Windows Update. As an administrator, you can then test those updates and approve them for use on your network, as Figure 3.4 shows.

 SUS only handles updates for Windows and bundled components, such as IE and Media Player. WUS will handle the gamut of Microsoft applications addressed by MBSA 1.2, including SQL Server, Exchange Server, Microsoft Office, and more.

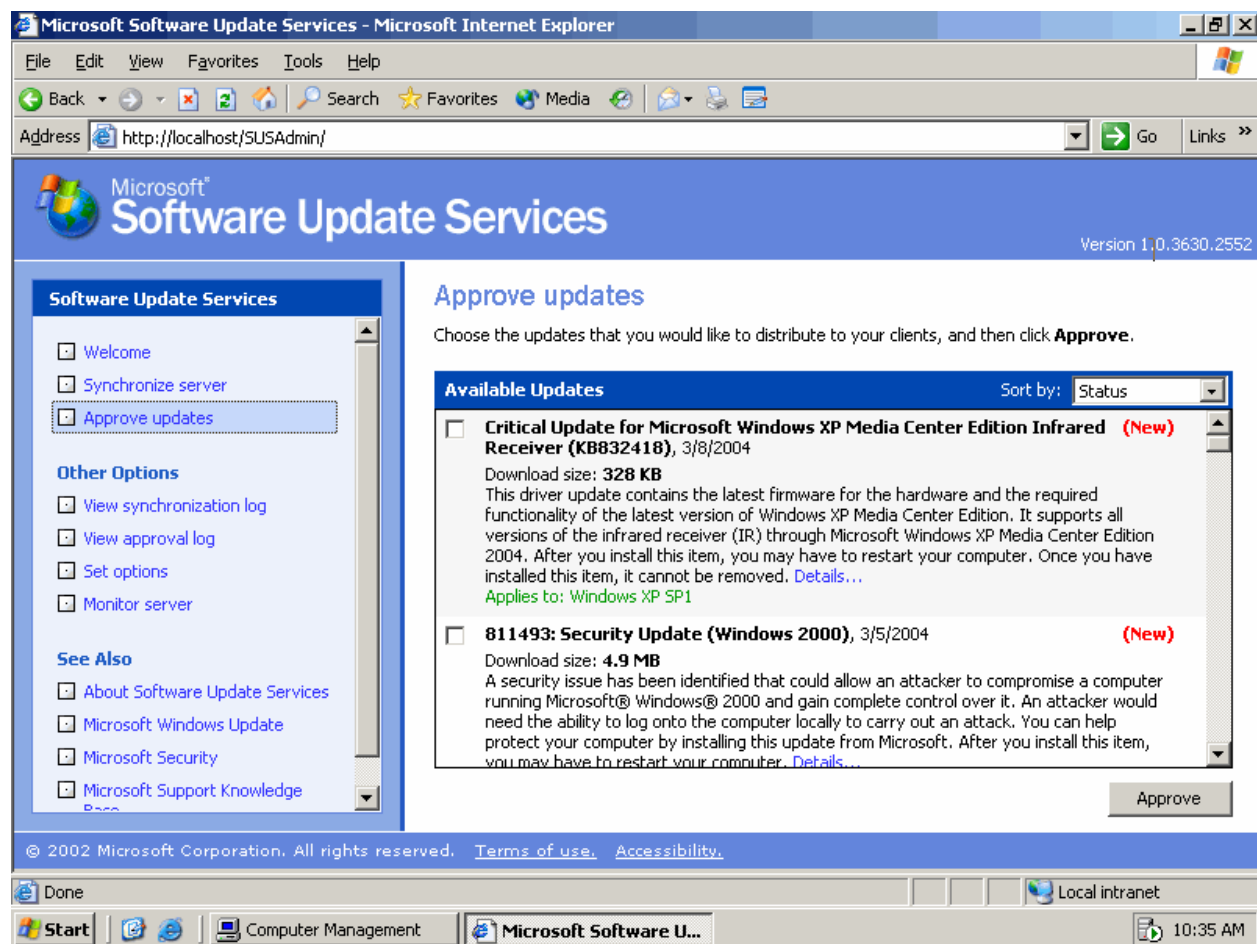


Figure 3.4: Approving updates in SUS.

Along with the ability to prevent users from accessing Windows Update directly, SUS/WUS gives you complete control over the updates that are deployed within your environment. So how does SUS or WUS address the overall goals of configuration management?

- **Inventory**—Both SUS and WUS collect inventory information to decide which updates need to be deployed. A client running on each targeted computer (Win2K and later) uses an MBSA-like engine (WUS will actually use the MBSA 1.2 engine) to look for missing patches, then installs the missing patches automatically from the SUS/WUS server.
- **Baselines**—As with MBSA, all scans are done according to Microsoft’s idea of a baseline. Because you, as the administrator, must approve each update for deployment, you can prevent unwanted updates from entering your environment. However, you can’t program SUS or WUS to look for vulnerabilities that you have identified. Further, SUS and WUS concern themselves only with patchable problems; configuration problems—such as blank account passwords or open mail relays—aren’t considered because SUS/WUS can’t fix those problems.

Another problem with SUS is the difficult of maintaining more than one approval list for your network. With SUS, updates are approved for all computers, meaning you can’t approve an update just for a small test group. WUS corrects that problem in part by allowing you to create computer groups and approve updates on a per-group basis. However, WUS doesn’t create those computer groups dynamically, which means you’ll need to make sure you keep on top of group membership. If you install IIS on a server and forget to include the server in the group that receives IIS updates, that server’s IIS installation will remain unpatched and vulnerable.

- **Analysis**—SUS/WUS does a good job of analysis. Because patch scans are conducted from scratch each time, SUS/WUS doesn’t make any assumptions about what has already been deployed to a computer; patches that are overwritten or uninstalled will be redeployed at the next scan.
- **Deployment**—SUS/WUS *is* a deployment tool and nothing more.

SUS/WUS, then, seems to represent a solid means of keeping systems up to date, at least with regard to patches. The caveats are that they use a Microsoft-defined baseline rather than your own and that they make it difficult (especially in the case of SUS) to manage updates for groups of computers.

#### Is There a Complete Solution?

Reading back through the past few tools, you might start to see the glimmerings of a complete solution: use Group Policy to deploy configuration settings, SUS or WUS to deploy patches, and MBSA to check up on the two of them and to watch for configuration problems that neither SUS/WUS nor Group Policy can address.

You *can*, in fact, do so, and that’s what Microsoft intends. You’ll even get a hint of continuous configuration management because both Group Policy and SUS/WUS work to continually deploy their respective configuration items. However, you’ll be operating from a set of Microsoft-defined baselines (especially with SUS/WUS and MBSA), and your solution will likely face serious scalability issues, particularly in the MBSA component, which simply isn’t designed for regular large-scale use.

You’ll also lack an inventory, which can be useful for performing non-real-time security analyses and for targeting deployment and configuration activities. You won’t, for example, have a clear idea of how many computers in your environment contain a specific model of processor that has a known security vulnerability; MBSA simply doesn’t collect that information, let alone store it.

## Security Templates

Security templates are a way to define a standardized set of configuration settings and apply them to groups of computers. A security template looks a lot like a GPO, as Figure 3.5 shows. However, in addition to the configuration settings present in Group Policy, a template can specify system services information, registry security, and even file system security. Once a template has been created, you can deploy it by simply importing the template's INF file into a GPO, then linking that GPO to the appropriate domain, site, or OU.

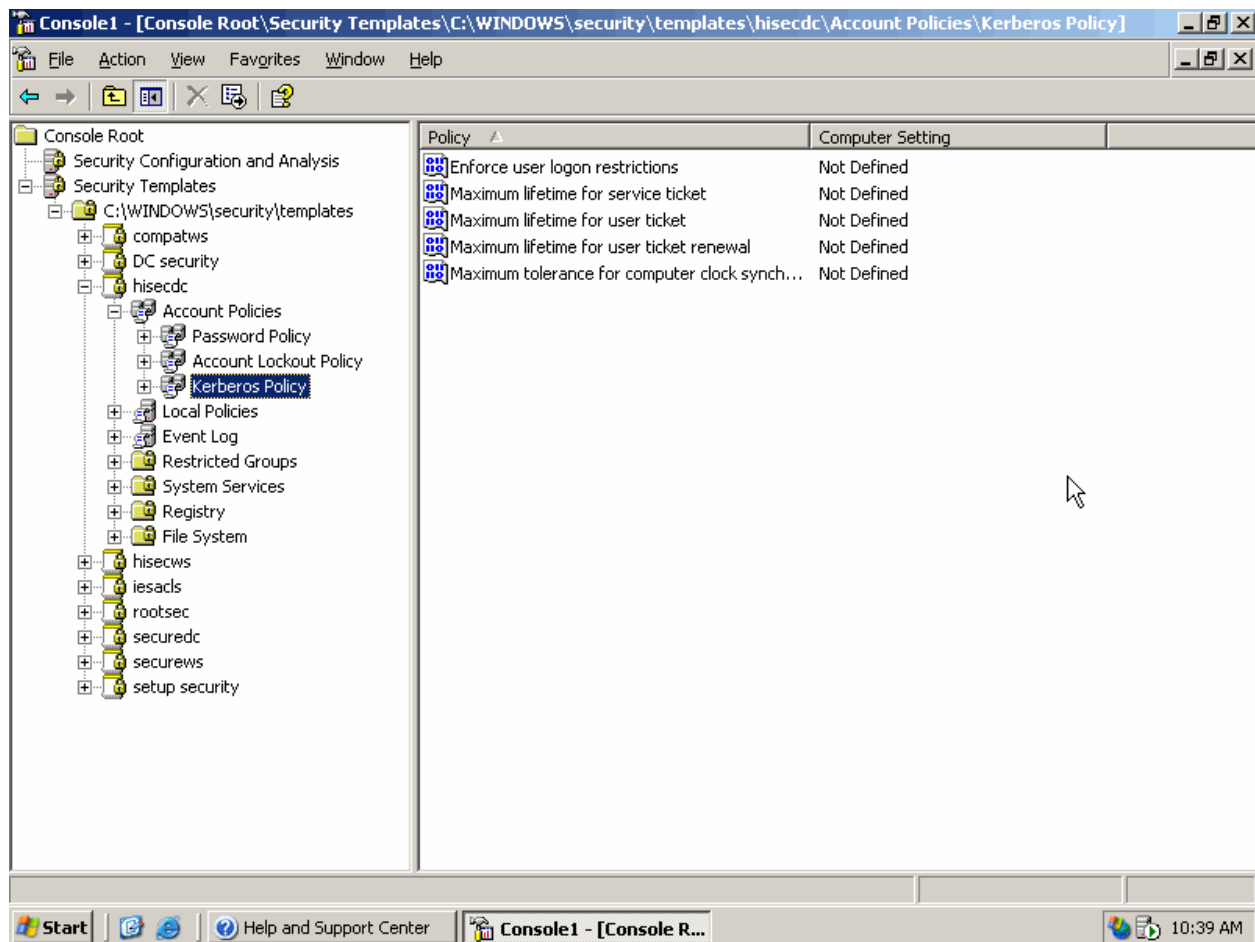


Figure 3.5: Modifying a security template.

You can also deploy templates without a domain by using the command-line tool Secedit.exe. Secedit works on the concept of a *security database*, which is a standalone database used to contain security settings. For example:

```
Secedit /analyze /db MyDatabase.sdb /cfg MyTemplate.inf
```

will create a new security database named MyDatabase.sdb, and import the MyTemplate.inf security template into this database. You can specify UNC paths to import a template from a file server, but each computer needs to use its own security database. After importing a template, you can execute

```
Secedit /configure /db MyDatabase.sdb
```

to modify the computer's current configuration so that it matches the database. Databases can, by the way, contain *multiple* templates; in the event of a conflict in their settings, the last template imported into the database takes precedence.

Secedit can be called from a traditional batch file assigned as a logon script, and it can be called multiple times. Each time, the settings in the database are applied to the computer's active configuration. In this way, the security template acts as a sort of baseline, and is continually reapplied to your computers, solving the problem of users possibly modifying their computers' configurations. However, Secedit doesn't necessarily notify you that a template *needs* to be applied; it simply applies the template without first analyzing the computer's current condition (that analysis is something you can do manually with the graphical Security Configuration and Analysis console).



Secedit works on Win2K, Windows XP, and WS2K3.

### Resource Kit Tools

The various Windows resource kits—versions have been produced for every major Windows version since Windows NT 3.1—come with a variety of tools that can help you roll our own configuration management strategy. For example, one of the tools in the WS2K3 resource kit is Spcheck.exe. Combined with a downloadable INI file from Microsoft, this tool checks the service pack version on any Windows system version NT 4.0 and later. The tool checks several files and drivers to identify the latest service pack (and hotfix) version, helping identify which are installed and which are missing.

The resource kits are a valuable source for scripts, command-line tools, and even a few graphical tools that can become part of a configuration management process. Generally, these tools supplement functionality built-in to Windows or provide alternative means of performing tasks (such as providing a scriptable command-line tool that performs a task that could otherwise only be performed manually through the GUI).



Check out the resource kits online at <http://www.microsoft.com/windows/reskits/default.asp>.

There are some cautions to be observed regarding resource kit tools, however. First, these tools, although provided by Microsoft, are unsupported by Microsoft. This fact is unfortunate because many of the tools provide the best possible way to accomplish certain tasks (if not the only way, in a number of cases). Any problems you have using resource kit tools are *your* problems, and Microsoft's product support folks generally won't help you with them. The lack of support makes a lot of businesses understandably gun-shy about relying on resource kit tools for critical tasks such as configuration management; they are fine using the tools as a shortcut for administrative tasks, but the business can't *rely* on the tools because they're unsupported.

### Scripting Your Own Tools

Scripting has become a popular tool for many administrators, enabling them to automate time-consuming tasks and to perform repetitive tasks more consistently and efficiently. For that reason, scripting—particularly with VBScript and WMI—is often seen as a potential way to create a homegrown configuration management system. WMI, in particular, excels at querying certain types of configuration data from Windows OSs.

For example, the VBScript that Listing 3.1 shows uses WMI to query each computer in the domain for its current service pack level. This information is written to a file, C:\Inventory.txt, which can be imported in Microsoft Excel.

```
'connect to the root of AD
Dim rootDSE, domainObject
Set rootDSE=GetObject("LDAP://RootDSE")
domainContainer = rootDSE.Get("defaultNamingContext")
Set oDomain = GetObject("LDAP://" & domainContainer)

'start with the domain root
WorkWithObject(oDomain)

'open output text file
Dim oFSO, oTS
Set oFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set oTS = oFSO.CreateTextFile("c:\inventory.txt",True)

Sub WorkWithObject(oContainer)
  Dim oADObject
  For Each oADObject in oContainer

    Select Case oADObject.Class
      Case "computer"

        'oADObject represents a COMPUTER object;
        'inventory it
        sComputer = oADObject.Name

        On Error Resume Next

        'connect to WMI on the remote machine
        Set oWMIService = GetObject("winmgmts:" _
          & "{impersonationLevel=impersonate}!\\" & _
          sComputer & "\root\cimv2")

        'connected OK?
        If Err = 0 Then
```



```

`query operating systems
Set cOperatingSystems = oWMIService.ExecQuery _
    ("Select * from Win32_OperatingSystem")

`for each operating system...
For Each oOperatingSystem in cOperatingSystems

    `write out the SP version
    oTS.WriteLine sComputer & ", " & _
        oOperatingSystem.ServicePackMajorVersion _
        & "." & oOperatingSystem.ServicePackMinorVersion

Next

End If
On Error Goto 0

Case "organizationalUnit" , "container"
    `oADObject is an OU or container...
    `go through its objects
    WorkWithObject(oADObject)

End Select

Next

End Sub

`finish up
oTS.Close
WScript.Echo "Inventory complete"

```

**Listing 3.1: VBScript that uses WMI to query for the service pack level of each computer in the domain.**

You can probably see how this script could be expanded to inventory almost anything available through WMI, which is a great deal of information (WMI as implemented in WS2K3, for example, contains hundreds of classes, each of which represents various aspects of the OS, computer hardware, performance counters, and more). Unfortunately, the output of a script like this example script is useful only for manual correction. In other words, you can use this script's output to manually install service packs on machines that are out of date, but there is no way to feed this script's inventory list into a Group Policy for an automated, targeted deployment of the service pack.

WMI itself—and thus, VBScript—is somewhat limited in its ability to modify settings, even though it can report on a great many different settings. WMI can be used to perform basic actions such as restarting a computer or setting a computer to use the Dynamic Host Configuration Protocol (DHCP) for its IP address; it can't be easily used to change file security settings or registry security settings, and it can't be used to perform actions such as change a SQL Server sa account password. And, although the breadth of WMI improves with every new Windows OS, it still isn't all-inclusive: basic services such as WINS and DHCP aren't exposed through WMI, and many older Windows Server system products (such as SQL Server 2000) don't interface directly with WMI.



### The Hidden Cost of Roll-Your-Own

Some administrators elect to create their own configuration management tools simply to save money. Perhaps their company has no budget for new tools or perhaps there is no time to evaluate available tools and select one. Either way, creating your own tools—by using scripts or combining existing tools such as security templates and Group Policy—seems like a cost-effective way to get configuration management for free.

Everything costs, though. Scripts cost time—and therefore, money—to develop. Piecemeal solutions incorporating a half-dozen other tools often require more time to set up and maintain than a single integrated solution would cost.

Long-term maintenance is another concern, at least for the company as a whole, if not for the individual administrator. Although the company's current staff might be able to create scripts and piece together solutions from a variety of disparate tools, those individuals will *eventually* leave or be promoted. Their replacements might not be able to maintain their customized, homegrown solution, meaning the company might have to start from scratch. A commercial configuration management system, however, comes with documentation, customer and technical support, and other resources so that the solution could remain in use through several staff changes.

Although it is certainly possible to “roll your own” configuration management system—or at least, pieces of one—examine the overall, long-term costs of doing so. A homegrown solution might seem cheaper in the short term, but in the long term, a more beneficial approach might be to simply purchase a packaged solution that can be implemented faster and more easily maintained over time.

Although scripting can definitely play a role in a homegrown configuration management solution, you should be aware of its limitations. Scripts have obvious scalability issues; running the script that Listing 3.1 shows in a domain with 20,000 computers will probably take several days to complete.

## Closing the Gaps in Configuration Management

At this point, you have probably gathered that you can definitely “roll your own” configuration management system, but there will be some gaps in it, it might not be perfect, and it might involve a little bit more effort than you thought. The next few sections describe the various features that your homegrown solution will probably be missing, and explain what you can do to fill those gaps. This arena is, by the way, where commercial software comes into play. Some of the problems a homegrown solution will have simply can't be solved by more homegrown solutions, unless you're setting out to “home grow” a commercial configuration management system. The specific areas I'll discuss are:

- Scalability—Although solutions such as scripts *can* solve portions of the configuration management problem, they don't often scale well for large organizations. Running a script against 20,000 computers is, as I mentioned, unlikely to solve any real needs.
- Dynamically grouping computers—One problem with using WUS/SUS or other solutions for configuration management is that they can't dynamically group computers. In other words, WUS will allow you to group all of your IIS servers together for patch management purposes, but *you* must tell WUS which servers are running IIS. A dynamic group, in contrast, would *look* for servers running IIS and place them into that patch management group automatically.
- Periodic deployments—This area of configuration management is overlooked by many solutions: the periodic deployment of new or updated software. Group Policy's IntelliMirror features handle this task fairly neatly, but lack flexibility.

- Continuous management and enforcement—The ability to constantly monitor computers' compliance to specific configuration baselines and to either alert you or automatically correct compliance problems is crucial to a complete configuration management solution.
- Centralization—Many of the homegrown solutions I've described don't offer centralized management, which is a critical capability in today's "do more with less" economy, when one administrator is expected to have more and more control over growing networks.
- Analysis and reporting tools—Very few of the homegrown solutions I've described offer any kind of analysis or reporting capabilities. Some of them—such as WUS—offer very basic reports, but may not offer enough to help you manage your network the way you need to.
- Ease of use—Above all, configuration management systems must be easy to use or you'll simply wind up *not* using them. Homegrown solutions aren't always easy to use because they're patching together so many disparate tools, rather than offering an integrated solution that's designed to work as a system.

### **Scalability**

The very idea of having MBSA scan all 10,000 computers in your domain (or however many you have) should make you cringe. Having MBSA scan *ten* computers takes a long time; I can't even imagine what a whole network full of computers would require in terms of time and network bandwidth. This type of situation is what I mean when I say that most homegrown solutions aren't *scalable*: They may work great for one or two computers, but by the time you hit a hundred, they're difficult to use and at a thousand computers, they're useless.

Most scalable solutions take the form of a client/server architecture. Microsoft SMS does so, in fact, by installing agent software on each managed computer. ConfigureSoft Enterprise Configuration Manager does the same, as does the configuration management solution offered by TripWire. Even some forthcoming, free solutions from Microsoft, such as the Microsoft Audit Collection Server (MACS) use a robust client/server architecture. In this type of architecture, a client lives on each managed computer, gathering inventory information, accepting configuration instructions, and so forth; this information is passed to a server on which it can be aggregated and analyzed appropriately, often using high-end database technologies such as Microsoft SQL Server.

This client/server architecture takes advantage of *distributed processing*, meaning each computer on the network invests a little bit of time in making the solution work: Managed computers conduct their own inventories locally, then pass that information—often passing only what changed since the last inventory—to the server. This way, the server doesn't need to reach out to each managed computer for a full inventory every single time; the server simply accepts incoming data and processes it, allowing the server to focus on that task.

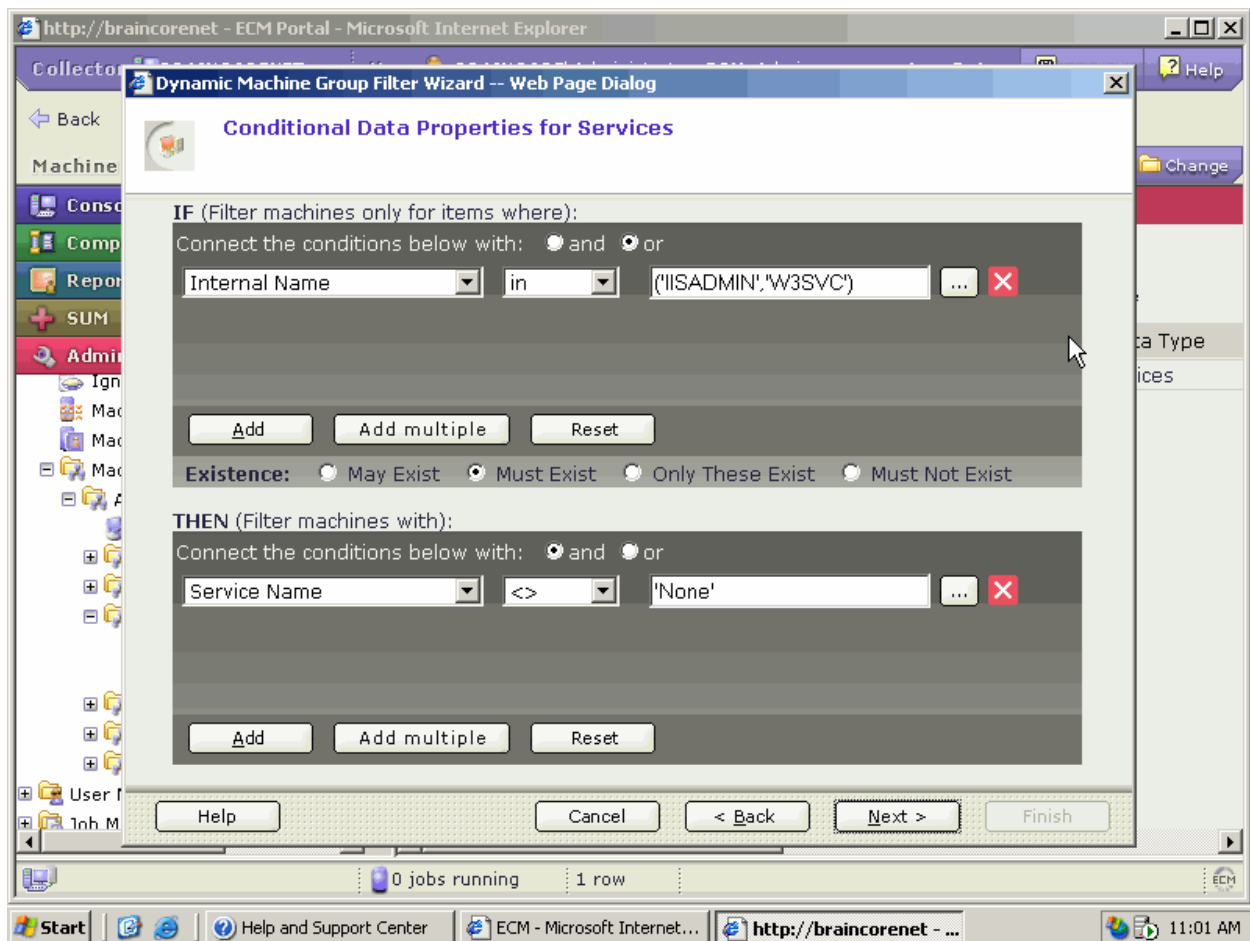
Any worthwhile commercial solution will use this client/server architecture; any solution that doesn't probably doesn't offer scalability for larger networks—and that, unfortunately, includes most homegrown solutions, which generally—like MBSA and scripts—rely on one computer to do most of the work.

### ***Dynamically Grouping Computers***

Being able to target computers based on their current condition, rather than what *you* know about them (which may be out of date) is important. For example, if you want to deploy a patch to all of your IIS computers or ensure that all of your IIS computers have a particular configuration setting in place, you need to be able to accurately target all computers running IIS—even those that you may not be aware of. Most freely available solutions that could accomplish this task, including WUS and Group Policy, don't have an analysis component that can *find* IIS servers for you; you must tell them which computers to target, and if you miss a server, that server will remain untargeted.

SMS addresses this problem by allowing you to target software deployments based on an inventory query. For example, you might query all computers that have a particular service (such as IIS) installed, then target a software deployment to the results of that query. This method works great for deploying new software, which is a less-frequent task than, say, deploying security updates or enforcing configuration baselines.

Some third-party solutions, such as ConfigureSoft's Enterprise Configuration Manager (ECM), offer a similar capability. For example, Figure 3.6 shows a rule set, or filter, that defines a group. In this case, the group consists of all computers running the IISADMIN or W3SVC services. As the solution inventories your computers, any computers meeting this criterion will be automatically added to this dynamic group. Any configuration baselines you specify for this group will then be applied to all of those computers. If IIS is installed on a new server without your knowledge, it won't matter: the configuration management software will inventory the server, realize that IIS is installed, consider the server to be a part of this dynamic group, and treat it accordingly.



**Figure 3.6: Creating a dynamic machine group.**

This automation provides a value-added feature beyond the solutions you can create for yourself. You no longer need to worry about keeping track of your computers. Because the software automatically inventories computers on your network (once configured to do so), you won't "miss" computers that change conditions.

### **Periodic Deployments**

Group Policy is an excellent technology for periodic software deployment. However, it's missing any kind of real flexibility. Keep in mind that Group Policy can't be targeted to individual computers; it can only be targeted to a container of them—a domain, a site, or an OU. If you want to deploy a new software application to all Windows XP Professional computers by using Group Policy, you must make sure that all of those computers are in a place (such as an OU) where they will be targeted by a Group Policy.

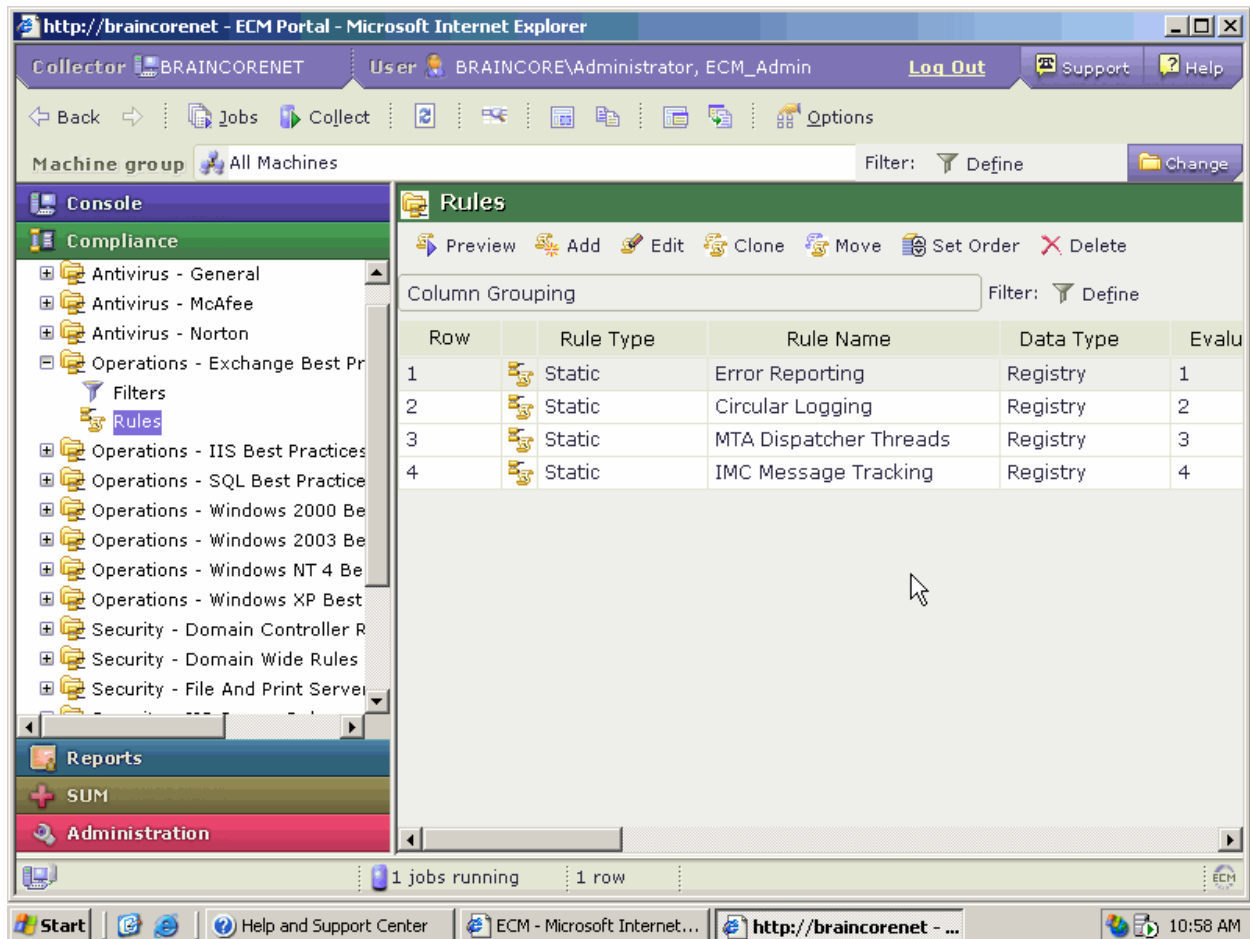
Group Policy in WS2K3 introduces a new capability called *WMI filtering*. This capability allows you to specify a WMI filter, and have the GPO apply only to computers that meet that criterion. So you could, for example, link a GPO to your entire domain, but specify a WMI filter so that only computers running WS2K3 with no service pack are affected. This feature is a great capability, and gives Group Policy some of the same features as SMS. However, there's a catch: Group Policy relies on the targeted computers to download the GPO, examine the WMI filter, and decide for whether they meet the criteria or not. Currently, only WS2K3 machines are capable of doing so; thus, the feature has limited applicability in most enterprises.

In this arena, solutions such as SMS come into play. As I've described, SMS actively inventories your computers, so SMS itself knows which OS the computers are running, what service pack level they're at, and so forth. You can create queries on these criteria and target only matching computers with an SMS software deployment. This functionality is step up from Group Policy and provides much more granular capabilities for periodic software deployments.

### ***Continuous Management and Enforcement***

Throughout this chapter, I've emphasize the importance of continuous configuration management. Effective configuration management is about more than just pushing out patches and configuration settings through Group Policy; it involves constantly scanning computers to ensure that those pushed-out configurations haven't changed and, if they have, that they are put back to where they're supposed to be.

For example, consider the console that Figure 3.7 shows. You see that a compliance category named Operations—Exchange Best Practices is opened. This category has several rules in effect, all of which target specific registry settings. The idea is simple: Any computer that is expected to maintain this particular compliance category must have these four registry keys set as desired. If the computer doesn't, it is out of compliance. This compliance category might be assigned to a dynamic group, which targets servers running the Exchange Server services. Thus, any server running Exchange must be configured in the fashion shown. If it isn't, it is out of compliance, and, at the very least, an alert will be generated.



**Figure 3.7: Defining compliance rules.**

This type of third-party solution isn't simply pushing out these rules; it is actively scanning computers. If a new Exchange Server computer is brought online, the software recognizes the system as belonging to the Exchange Servers dynamic group, and expects it to comply with these four compliance rules (and probably other rules from other categories). The administrator doesn't need to take any action or even realize that a new Exchange server was brought online. The solution dynamically picks up on this server's existence and configuration because the solution is *continually* monitoring computers for compliance with the defined rules.

This functionality is the ultimate in configuration management: Computers can be dynamically grouped based on the actual services they provide, and compliance rules can be monitored and enforced for each dynamic group. Although you can use security templates to enforce these registry settings, *you* would need to identify the servers running Exchange. If one was brought up while you were on vacation, or one was brought up without your knowledge, it wouldn't be compliant with your configuration baselines.

## **Centralization**

Of course, solutions such as Group Policy are, by their very nature, centralized. Most of the other homegrown solutions we've looked at, however, are not. MBSA maintains no database of any kind for the information it finds, much less a centralized database; scripts can be distributed across a network.

Commercial solutions provide a centralized management console, centralized database for configuration and inventory information, and so forth. This centralization is a benefit of their client/server architecture, in which all data is brought up to the centralized location by design. This centralization allows the entire enterprise configuration management to be administered by a single person, if necessary, which is a valuable feature for busy environments and overworked administrators.

## **Analysis and Reporting Tools**

We haven't really looked at any homegrown solutions that offer great reporting. MBSA certainly offers reports, but looking at an MBSA report on a thousand computers is an exercise in futility. The information simply isn't organized in a fashion that makes it useful for large numbers of computers. WUS offers good reporting for patch deployment, but that's about it.

Commercial tools offer the power of a relational database and well-designed reporting tools to provide better reports. These tools can, for example, provide a list of all computers that need a new service pack, or a list of computers containing a specific kind of processor. In addition, they can provide you with a list of all computers that don't meet a new configuration standard that your company wants to implement, enabling you to gauge the impact of deploying that new standard. These reports are provided almost instantly, from the software's inventory database, without having to reach out and query each computer on your network for new data. This type of report is what administrators need—nearly-instant, up-to-date reports that consolidate enterprise information into one place.

## **Ease of Use**

One major problem I've pointed out with homegrown solutions is their lack of ease of use. Scripts in particular are used by fewer than 32 percent of Windows administrators, according to a recent Microsoft survey on Windows automation. Thus, a homegrown solution involving scripts is unlikely to qualify as "easy to use" for most of the administrators in the company, and if the solution isn't easy to use, nobody will use it. Worse, if the administrator who wrote the script leaves his or her position or the company, there might be nobody left familiar enough with the script to continue using it.

Commercial solutions all offer easy-to-use graphical user interfaces (GUIs), professionally written documentation, and other tools designed to make them easy to use. A less-experienced administrator can be productive with these tools, allowing senior administrators to work on new projects rather than focusing on maintaining a library of scripts on which the company has come to depend entirely for its configuration management.



## Summary

In this chapter, we've explored the ways in which you can create your own configuration management system, or at least some of the pieces of one, using freely available tools, scripts, and so forth. We've also examined the gaps that these solutions contain, including problems with scalability, ease of use, and flexibility. The goal of this chapter is to provide you with a better feel for the technology that goes into configuration management solutions, and an idea of what you can accomplish without purchasing a commercial software package—and what you'll be missing.

In the next chapter, we'll focus on commercial solutions that provide a more complete continuous configuration management infrastructure. This chapter will identify how these solutions meet another portion of the continuous configuration management challenge: configuration analysis and enforcement, along with patch and update deployment. I'll also come back to built-in solutions such as Group Policy to show you how they can fit into an overall configuration management scheme.