

realtimepublishers.com<sup>tm</sup>

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



**Configuration  
Management**

*for the Windows Enterprise*



*Don Jones*

---

Chapter 4: Automated Continuous Configuration Management .....	57
The Goals of Continuous Configuration Management .....	57
Initial Provisioning.....	59
Configuration Monitoring.....	61
Configuration Enforcement .....	63
Patch Management.....	65
Software Deployment .....	66
Tools and Systems .....	67
ConfigureSoft Enterprise Configuration Manager.....	67
Ecora Patch Manager .....	71
Microsoft AD and Group Policy.....	73
Microsoft SMS.....	73
Microsoft Windows Update.....	73
Microsoft WUS and SUS.....	74
Tripwire for Servers .....	74
Summary .....	75

## 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.

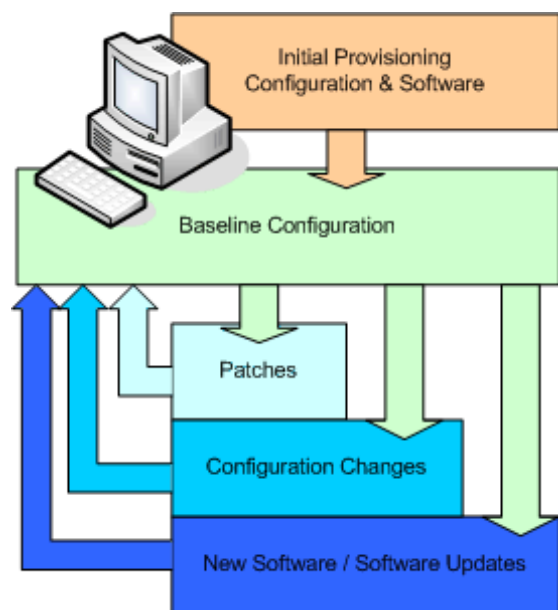
If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at [info@realtimedpublishers.com](mailto:info@realtimedpublishers.com).

## Chapter 4: Automated Continuous Configuration Management

In the previous chapter, I showed you how to create a very basic, home-grown configuration management system. I also pointed out some of the shortcomings associated with a home-grown system—not the least of which is a fairly high level of labor overhead. Such a system—no matter how much time you put into it—would be hard-pressed to offer all of the features provided by a good configuration management system. Thus, this chapter focuses on commercial tools, effective processes, and industry methodologies for automated *continuous* configuration management.

### The Goals of Continuous Configuration Management

Before we delve into which tools and systems help provide continuous configuration management, I want to lay out the business goals that these tools and systems are meant to address. One way to think about these goals—at the risk of using an already overused term—is to think about the *life cycle* of a computer. Figure 4.1 illustrates this life cycle.



**Figure 4.1:** Continuous configuration management life cycle.

Every computer starts with an initial deployment, and your continuous configuration management process should include tools to make that initial provisioning more convenient and consistent. As I pointed out in previous chapters, provisioning isn't critical from a management standpoint because regardless of the level of consistency of your initial provisioning, computers' configurations will still drift over time. Your overall continuous configuration management process must accept and accommodate this fact, making initial provisioning an unnecessary step from a purely management point of view. However, provisioning should offer a low-effort means of deploying highly consistent configurations to both reduce administrative overhead and to improve the end-user experience.

After initial provisioning is completed, the computer should be in a state representative of a baseline configuration. At this point, continuous configuration management really kicks in, ensuring that the computer *remains* at the baseline configuration, even if the baseline is redefined at some point. For example, patches, desired configuration changes, and even new software will all change the baseline (or desired configuration state) for your computers, and a continuous configuration management process should provide tools to help you bring computers into compliance with the baseline. Your continuous configuration management process should also help repair computers that are misconfigured, bringing them back into compliance with your baseline.

Thus, a good set of continuous configuration management tools should incorporate the following functionality:

- Tools for rapidly deploying new computers—Such tools might include imaging applications such as Symantec Ghost and deployment tools such as Windows Remote Installation Services or Automated Deployment Services. Although these tools don't contribute to continuous configuration management directly, they help improve the initial provisioning process, reducing administrative overhead and improving consistency for end users.
- Configuration monitoring—Your tools should provide alerts and notifications whenever managed systems drift from your desired configuration state.
- Configuration enforcement—In some situations, you might want your tools to reconfigure systems to bring them back into compliance with your baselines.
- Patch deployment—Because patches—particularly security-related patches—are such an important part of the configuration management landscape, you should invest in tools that provide solid patch management and deployment.
- Software deployment—You'll need some means of deploying new and revised applications to your computers; regardless of whether that means is the basic deployment provided by Group Policy or the more powerful features offered by a tool such as SMS, make sure software deployment is in your continuous configuration management toolkit.

You should have a few key features that you look for in each of these continuous configuration management areas.

### Patches vs. Software

Why should patch deployment be considered something different than normal software deployment? Software deployment—deploying new applications or new versions of applications—is typically project-related. You don't normally find enterprises rolling out new versions of software on a daily basis. Software deployments generally involve license purchases, deployments to specified computers, and so forth through a fairly managed, structured process. You also don't typically worry about computers that haven't received an application; users will usually make sure that you hear about any missing applications! All told, applications tend to be a higher-profile, less-frequent configuration item.

Patch deployment, however, is a daily concern. Patches may be released within your organization on a scheduled basis, but emergency patches do come along, requiring immediate testing and deployment. You also need to ensure that patches *stay* deployed, and that any managed system that is missing a patch receives that patch quickly. Patches tend to be lower-profile, meaning users won't usually realize a patch is missing, and patch deployment tends to be more frequent.

Thus, tools used to deploy patches and those used to deploy new applications need slightly different feature sets in order to be the most effective. Software deployment tools, for example, may include license-tracking features to help manage license purchases. Patch management tools may run daily scans of managed systems to look for missing patches. Although these requirements differ, you can use a software deployment tool to deploy patches. Windows Group Policy, for example, can be used to deploy patches if they're packaged as Windows Installer packages. However, Group Policy doesn't provide the features needed to really *manage* patches; it can't scan for missing patches, it can't push out patches to a machine that suddenly needs them because it's filling a different role, and so forth.

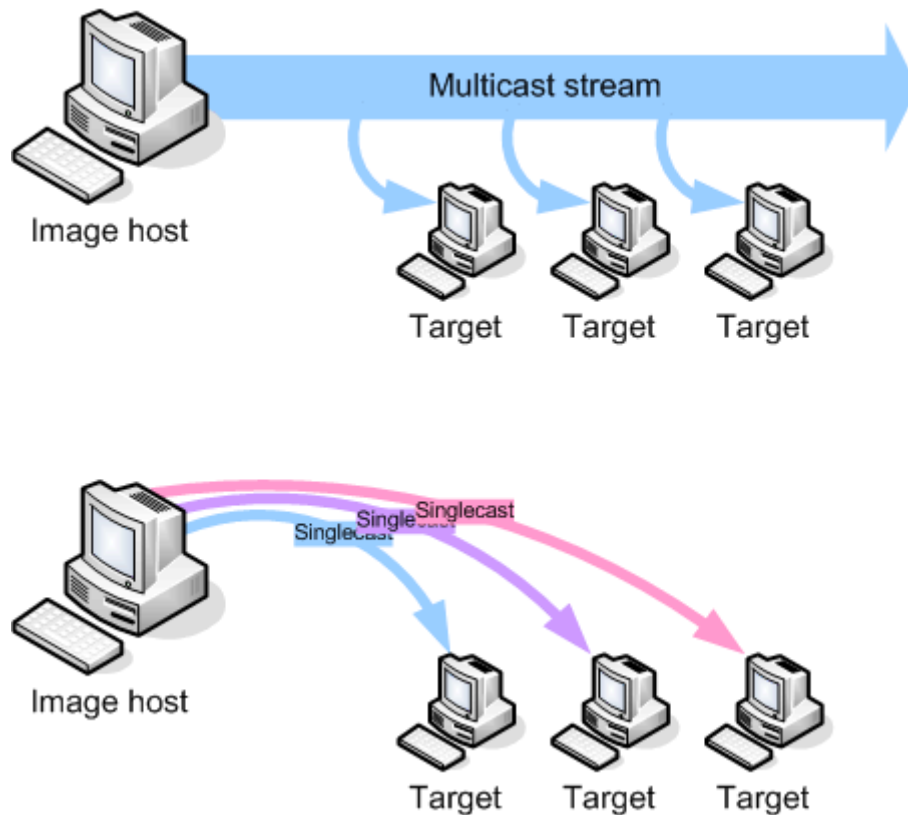
### Initial Provisioning

Initial provisioning best practices are all about convenience and consistency, allowing you to deploy properly configured, usable systems without much administrative effort. Any tool that meets these goals is a candidate for your continuous configuration management toolset.


Depending upon your specific environment and business needs, the following list highlights criteria you might look for in a useful toolset:

- An easy way to update images—Traditional imaging software requires you to have a “model” computer configured exactly the way you want it; this model is then imaged, and the image is applied to new computers as you provision them. Updating the image can be painful if you must reconfigure a new computer and re-image it; some imaging applications allow you to directly edit the image to add applications, update files, and more.
- Patch applications—Images are out of date from practically the moment they're made. It's not efficient to create a new image each time a new patch is issued that affects the software or OS on the image; rather, most administrators apply the image, then use their regular patch management tools to bring the imaged system up to date with the latest patches. However, some higher-end imaging applications can incorporate the installation of initial patches to bring the image up to date. This feature is strictly a convenience, but it can reduce the time necessary to deploy new, properly patched systems within your environment.

- Multicast imaging—Deploying a dozen new computers is easier if you can deploy an image to all of them at once. Newer imaging applications use multicast traffic, meaning they don't use any additional network bandwidth above and beyond what it would take to deploy the image to one computer; a single transmission is received by all computers that you're imaging. Figure 4.2 illustrates this process and compares it with older, singlecast image deployment.



**Figure 4.2:** Multicast imaging can provision several computers at once.

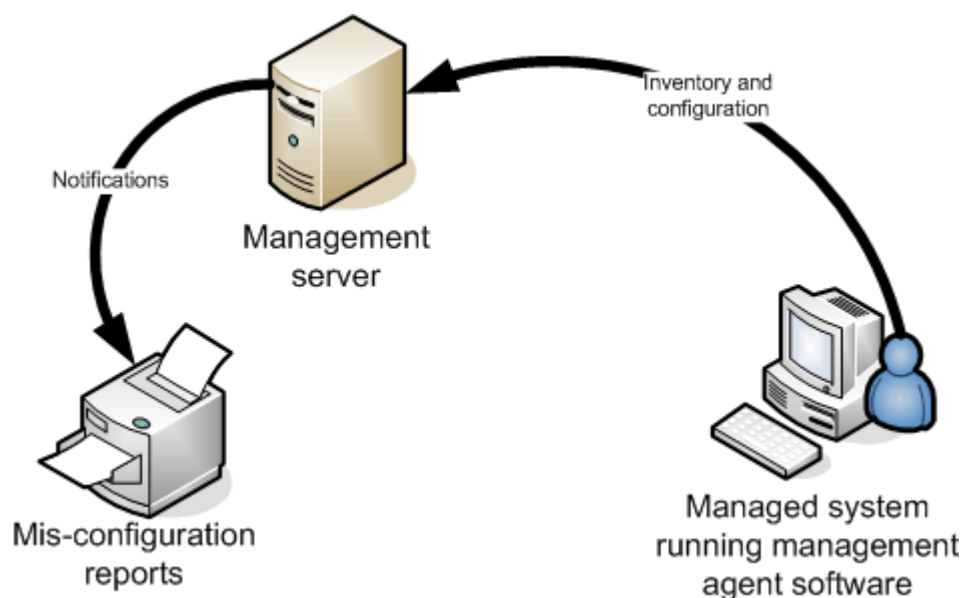
 Typically, you'll receive the maximum benefit from multicast imaging when all of the target computers are on a single network segment and not separated by a switch (use a hub) or router (unless you're creating a VLAN using a switch, which achieves a similar effect to a hub and helps restrict the multicast traffic to just those areas of the network that need to see it). Although multicast traffic can place additional overhead on switches and especially on routers, the overhead of singlecast traffic (where the image is sent independently to each target computer) is still much higher.

- Centralized control—Some systems provide a Portable Execution Environment (PXE) server, allowing newer computers with PXE-compliant network adapters to “boot from the network,” contacting an imaging server and even automatically selecting the proper OS image for that computer (this selection is often based on the network adapter's MAC address, meaning you must preconfigure the server with a list of addresses and images). This server-based imaging allows new computers to be connected to the network, switched on, and automatically provisioned with the desired image.

These features are all designed to provide faster and more efficient provisioning. As I've stressed, you shouldn't rely on an imaging system to provide a baseline configuration that is sufficient for continuous configuration management. Instead, the other tools in your continuous configuration management toolset should continuously evaluate managed systems for compliance with a set of defined standards and baselines, regardless of how those systems were initially deployed.

### **Configuration Monitoring**

Configuration monitoring is an easy task in theory. As Figure 4.3 shows, a management agent (a small software program running on each managed system) reports inventory and configuration data to a central server. This server compares the agent's information with a set of standards that you have defined; for any area in which the agent does not report information complying with your standard, a notification (or complete report) is generated.



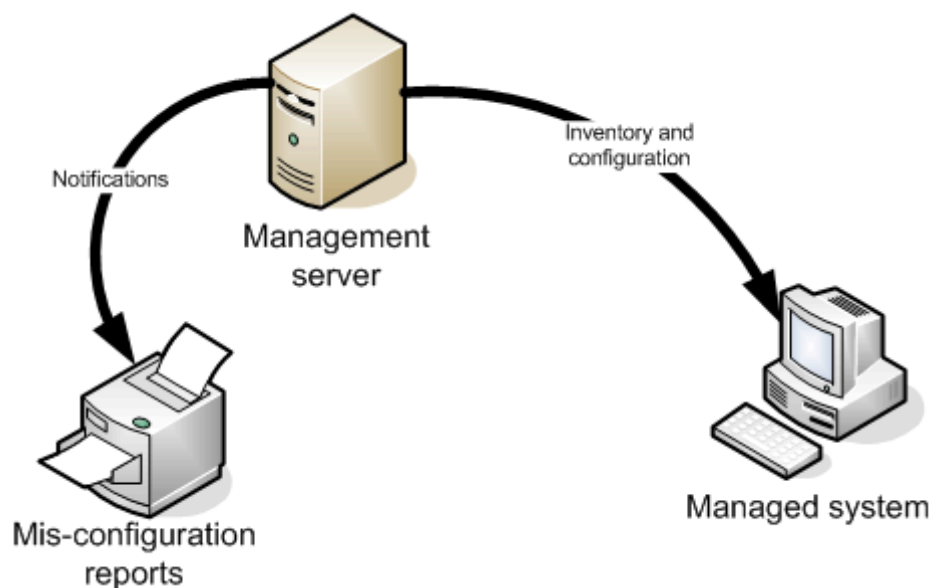
**Figure 4.3: Configuration monitoring architecture.**

In reality, as I discussed in the previous chapter, configuration notification is a difficult job. Windows contains literally thousands of configuration items, and that number grows with every installed application. Agent software must be capable of tapping into a wide variety of Windows Application Programming Interfaces (APIs) in order to retrieve inventory and configuration data, and management servers must typically compare and sort tens of thousands of pieces of data to produce notifications and reports.

This agent-server architecture is highly preferred, mainly because it results in lower overhead. Agents can stagger their inventory reports so that the management server isn't overwhelmed. Often, agents maintain a local copy of their inventory data, allowing them to send only changes to the server, which further reduces overhead on the server. The fact that the agent runs locally on each managed system helps provide the agent with access to an increased amount of configuration data. The agent's data-gathering phases can often run in the background, while the computer performs other useful tasks, and the agent generally consumes no network bandwidth until its report is transmitted to the server.



Figure 4.4 shows a less desirable way of collecting data—having the central server reach out and gather the information from remote managed systems. In this scheme, the server is doing most of the work. In order to determine what has changed on each managed system, the *entire* data collection process must be performed from scratch during each inventory, increasing overhead on the server and on the network. The data gathering also tends to utilize more of the managed system’s resources, meaning the system may be less responsive while inventory is being collected. Also, because Windows doesn’t expose all of its configuration information through a single API, some configuration information might not be accessible remotely.



**Figure 4.4:** Remote, agentless inventory gathering.

The Microsoft Baseline Security Analyzer (MBSA) uses this basic architecture. Although MBSA is useful for scanning one or two servers for missing Microsoft-provided security patches, it’s less suitable (as I discussed in the previous chapter) for working with larger numbers of computers.

You might wonder why, with configuration *enforcement*—which I’ll discuss next—you need to bother with configuration *notification*. After all, if something is *enforcing* your configuration standards, why should you need to be notified? Notification is necessary because any configuration change is a potential security or operational problem. If, for example, your enforcement tool is resetting a particular configuration setting on a daily basis, you need to look into why that setting is drifting every single day.

Configuration notifications may also alert you to *desirable* changes that need to be incorporated into your standards. For example, another administrator in your organization may have learned about a new registry setting that improves security on the TCP/IP stack. If that registry setting is in your baseline configuration, the administrator won’t be able to make their change “stick,” because your enforcement software might reset the registry back to your baseline standard. By reviewing your notifications, however, you’ll notice that the change was made, have an opportunity to investigate it, and consider it for inclusion in your enforced baseline.

## Configuration Enforcement

Configuration enforcement is a useful feature to ensure that computers remain configured as you want them. Configuration enforcement essentially combats configuration drift, keeping computers firmly fixed in your approved baseline configuration.

Windows Group Policy, a feature of Active Directory (AD), is a form of configuration enforcement: settings are pushed into the registry, overwriting any changes the user of the machine might have made. Settings are even refreshed on an ongoing basis, ensuring that the computer remains configured as desired. Drawbacks of Group Policy are that it is limited to registry-based configurations and its application isn't especially granular or dynamic.

 I'll review more about Group Policy later in this chapter.

One thing you need to be clear on, however, is that configuration enforcement must be continuous and fairly assertive. For example, a strength of Group Policy is that it doesn't check to determine whether a computer *needs* a particular configuration; Group Policy simply pushes out its configuration. As I covered in the three previous chapters, you can't make assumptions about a computer's configuration. Even if you just deployed a computer's configuration yesterday, its configuration may have already drifted from your baseline. Your enforcement tools must constantly reapply your configuration settings, as necessary, without trying to guess whether reapplication is necessary. It is acceptable (even preferable, for performance reasons) for your solution to analyze managed systems to determine what configuration drift has occurred, then apply the settings to correct the drift; this technique works because there is an active analysis of the computer to see what its current condition looks like.

Another key element of good enforcement is *grouping*. Consider Figure 4.5, which shows five computers grouped according to their role as a Web server, database server, or both. These groups were manually created by an administrator.

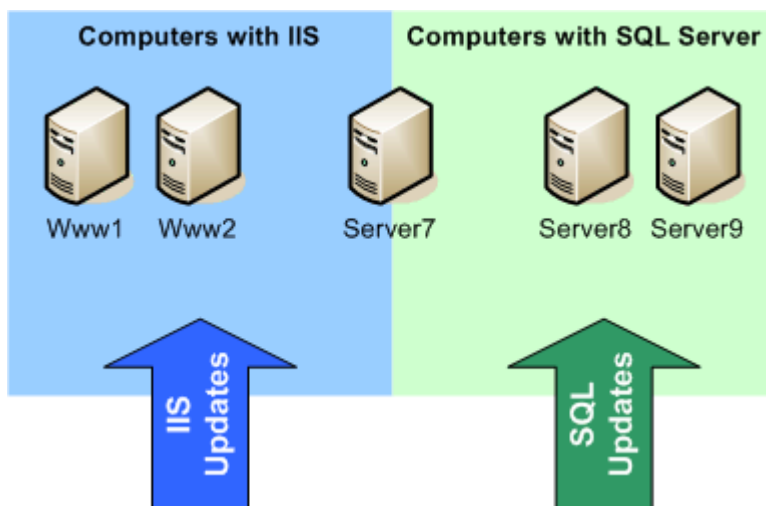


Figure 4.5: Manually grouping computers.

Notice that these two groups are used to deploy security updates (which, in this example, would be configuration updates rather than patches). Computers running IIS receive the IIS updates, and computers running SQL Server receive updates for that product. This type of manual grouping is provided by tools such as AD (which allows targeting of Group Policy based on group membership) and by Windows Update Services (WUS).

The drawback of this setup: What if IIS is later installed on Server8? Or what if the SQL Server Desktop Engine is later installed on server Www2? Unless an administrator remembers to update the group memberships, these two servers wouldn't receive the correct configuration updates and would be vulnerable to exploits in IIS and SQL Server. Because these groups are static, the configuration enforcement doesn't look at Server8 for IIS issues, and doesn't look at Www2 for SQL Server issues.

A feature offered by third-party tools, dynamic grouping is more effective. Figure 4.6 illustrates how a configuration management server uses inventory data to determine what software is installed on each computer. Each computer is then dynamically assigned to the appropriate groups. For example, Server7, Server8, Server9, and now Www1 are all running the services associated with SQL Server; the configuration management system correctly places both of them in the dynamic group for SQL Server computers.

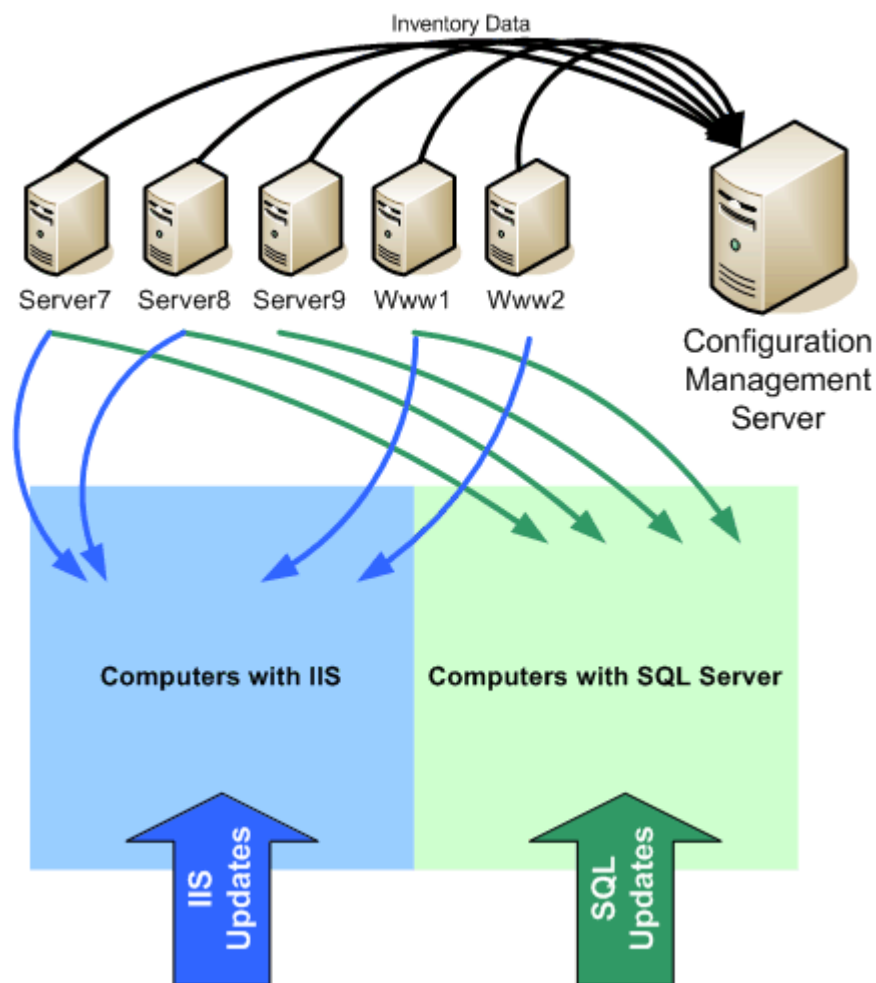



Figure 4.6: Dynamic grouping.

Dynamic grouping is an effective technique for enforcing configurations only on those computers that need the specific configuration, while eliminating the need to manually maintain those groups to ensure that computers always remain updated.

### **Patch Management**

Patch management is an essential part of any good continuous configuration management system. Patch management tools should be easy to use and, in a Windows environment, ideally integrate to at least some degree with Microsoft's XML-based database on available security patches (the same database used by Microsoft products such as WUS).

 Until recently, different Microsoft products—Windows Update, SUS, and MBSA—could scan the same computer and return different results. With MBSA 1.2, Microsoft introduced a new scanning engine, which will be included in Windows Update v5, WUS, and all other Microsoft patch-scanning services and products to improve consistency.

Security analysts are often confounded and frustrated by companies' attitude toward patches. Consider that three highly publicized viruses—SQL Slammer, MS Blaster, and Download.Ject—all targeted product vulnerabilities that had *already been patched*, often months in advance of the virus' release. Effective patch management would have made these viruses harmless; clearly, the industry isn't practicing effective patch management. One possible reason is the “do more with less” economy in which administrators are simply overwhelmed. The solution is automated patch management: a system that can obtain the necessary updates automatically, locate computers that need those patches, and deploy those patches automatically. Automation—taking the administrator out of the loop, essentially—helps to solve the problem of patch management.

#### **Agent vs. Agentless**

A big debate exists over agent and agentless patch and configuration management. As I've already described, an agent-based architecture places a small software program—the agent—on each managed system. The agent communicates with a central server to pass inventory and configuration information, and receives configuration instructions from the server. Agentless systems work without software installed on the managed systems.

The benefits of an agent-based system include reduced server overhead and better scalability; of course, the system must support the OSs in your environment in order to work. Agentless systems sometimes claim better cross-platform support, but that's rarely true; they must still understand the structure and layout of each managed OS, whether an agent is in use or not. Agentless systems tend to be less scalable because the central server takes on most of the work.

The biggest difference between the two is deployment: agentless systems obviously have no deployment, while agent-based systems have an agent that needs to be deployed. Better agent-based systems will perform agent deployment for you, pushing their agent out over the network rather than requiring you to perform a separate install or software deployment.

## Software Deployment

Software deployment isn't usually considered a maintenance function (except perhaps for service packs, which fall somewhere between "software" and "patches" in many cases); rather, software deployment is often considered a project-based task, occurring only when new applications are ready to be deployed in the environment. Still, software deployment changes the configuration of target systems and should therefore be a part of a comprehensive continuous configuration management process.

Most software deployment systems that deal with Windows utilize Windows Installer packages, because the Windows Installer technology is available for Windows 95 and later and is built-in to Win2K and later. Windows Installer handles the packaging side of the deployment, bundling applications' files and installation instructions into a single MSI file for more convenient management. Deployment systems, then, must only handle the task of getting the MSI package to the target systems and having Windows Installer take over and install the package.

Most software deployment systems—even the IntelliMirror features built-in to Windows—offer at least basic targeting capabilities. IntelliMirror, for example, allows packages to be targeted to machines contained in a particular site, domain, or organizational unit (OU) within AD. Higher-end systems, such as SMS, offer greater targeting capabilities, such as targeting a package to the results of a SQL query against the SMS database. This functionality allows you to deploy software to all machines with sufficient RAM, free disk space, and other requirements, if desired.

Higher-end systems—particularly agent-based ones—generally provide tracking and feedback. SMS, for example, can tell you which computers have yet to install a package you've deployed. Lower-end systems, such as IntelliMirror, don't provide this feedback capability—a shortcoming that can create management difficulties as you try to determine how complete a software deployment is within your environment.

### Confirmation for Patch Deployments?

You'll notice that I didn't include a tracking and feedback feature as desirable for patch management. Although some systems do offer this capability, it's uncertain how useful the feature is. Keep in mind that patches can become uninstalled and overwritten; unlike applications, it's often critical to security that patches *remain* properly deployed.

For this reason, I think it's more important that a patch management system continually scan managed systems to determine which patches are needed, then deploy them. There is no real need for confirmation in a system such as this, because if a target computer hasn't yet installed a patch, the system will continue trying until the patch is in place.

What can be helpful for patch management systems is a pre-deployment assessment—a report of which managed systems will need a particular patch. If you're deploying a patch that might have operational or support issues, it's good to know ahead of time how widespread those issues might be—particularly because a patch might apply to more computers than you realize.

## Tools and Systems

Before delving into specific tools, it is important to understand that no single tool provides every aspect of the continuous configuration management life cycle. Microsoft SMS, for example, is a useful software deployment tool but doesn't provide any features for configuration setting management; ConfigureSoft Enterprise Configuration Manager (ECM) addresses configuration control, but doesn't seek to be a software deployment solution. Different tools, then, fit different portions of the overall life cycle. In the next several sections, I'll introduce you to tools that are well-suited for various aspects of configuration management.

### ConfigureSoft Enterprise Configuration Manager

ConfigureSoft Enterprise Configuration Manager (ECM) is an agent-based system for configuration enforcement and patch management. The product consists of two basic, integrated components—a compliance feature and the Security Updates Manager (SUM), both managed from a Web-based administration console.

SUM provides patch management capabilities. As Figure 4.7 shows, SUM uses Microsoft security bulletins and knowledge base articles as the basis for its patch management feature.

The screenshot shows the 'SUM Bulletins By Bulletin' page in the ECM portal. The page includes a navigation menu on the left with options like Console, Compliance, Reports, and SUM. The main content area displays a table of bulletins with the following data:

Row	Bulletin	Revised	
1	KB870669	7/6/2004	How to disable the ADODB.Str
2	MS04-017	6/10/2004	Vulnerability in Crystal Reports
3	MS04-016	6/8/2004	Vulnerability in DirectPlay Cou
4	MS04-015	5/11/2004	Vulnerability in Help and Supp
5	MS04-014	5/11/2004	Vulnerability in the Microsoft Je
6	MS01-052	5/11/2004	Invalid RDP Data Can Cause T
7	MS04-011	4/21/2004	Security Update for Microsoft v
8	MS04-013	4/13/2004	Cumulative Security Update fo
9	MS04-012	4/13/2004	Cumulative Update for Microso
10	MS04-010	3/9/2004	Vulnerability in MSN Messenge
11	MS04-009	3/9/2004	Vulnerability in Microsoft Outlo
12	MS04-008	3/9/2004	Vulnerability in Windows Media
13	MS04-007 (MS04-011)	2/13/2004	ASN .1 Vulnerability Could Allow
14	MS04-006	2/13/2004	Vulnerability in the Windows Ir
15	MS04-004	2/3/2004	Cumulative Security Update fo

Figure 4.7: Managing patches based on Microsoft bulletins.




Notice the Create Template button in the right-pane toolbar. This button creates an Assessment Template, allowing SUM to check the systems it's managing to determine how many are affected by the bulletin. Once a machine is under SUM management, patches are automatically deployed to it. SUM is also capable of pushing out its agent to managed systems right through the Web console, meaning you don't need to undertake a specific software deployment. Alerting options allow you to be notified by email when new bulletins are added to the database and if a patch fails to deploy properly; this feature keeps you firmly in the loop regarding your enterprise patch management.


As I mentioned, SUM is only part of ECM's features. Another part of the base console allows you to view a number of important information categories, including:

- Change management—The tool provides automatically-generated log files that display all changes that have occurred to managed systems. These changes are detected by the local agent software on each managed system, and reported to the central ECM server.
- Aggregate data—This data provides information such as free disk space, processes, IP information, service configuration, and more across your enterprise.
- Security aggregate data—Similar to the regular aggregate data feature, this section allows you to browse security-related information, such as NTFS audit settings, registry key permissions, hotfixes, and more, from any machine in your enterprise.
- Enterprise applications—This section allows you to browse specific enterprise-wide applications, such as IIS and SQL Server. You can look at each machine running this application and browse various configuration information, all from the central console.

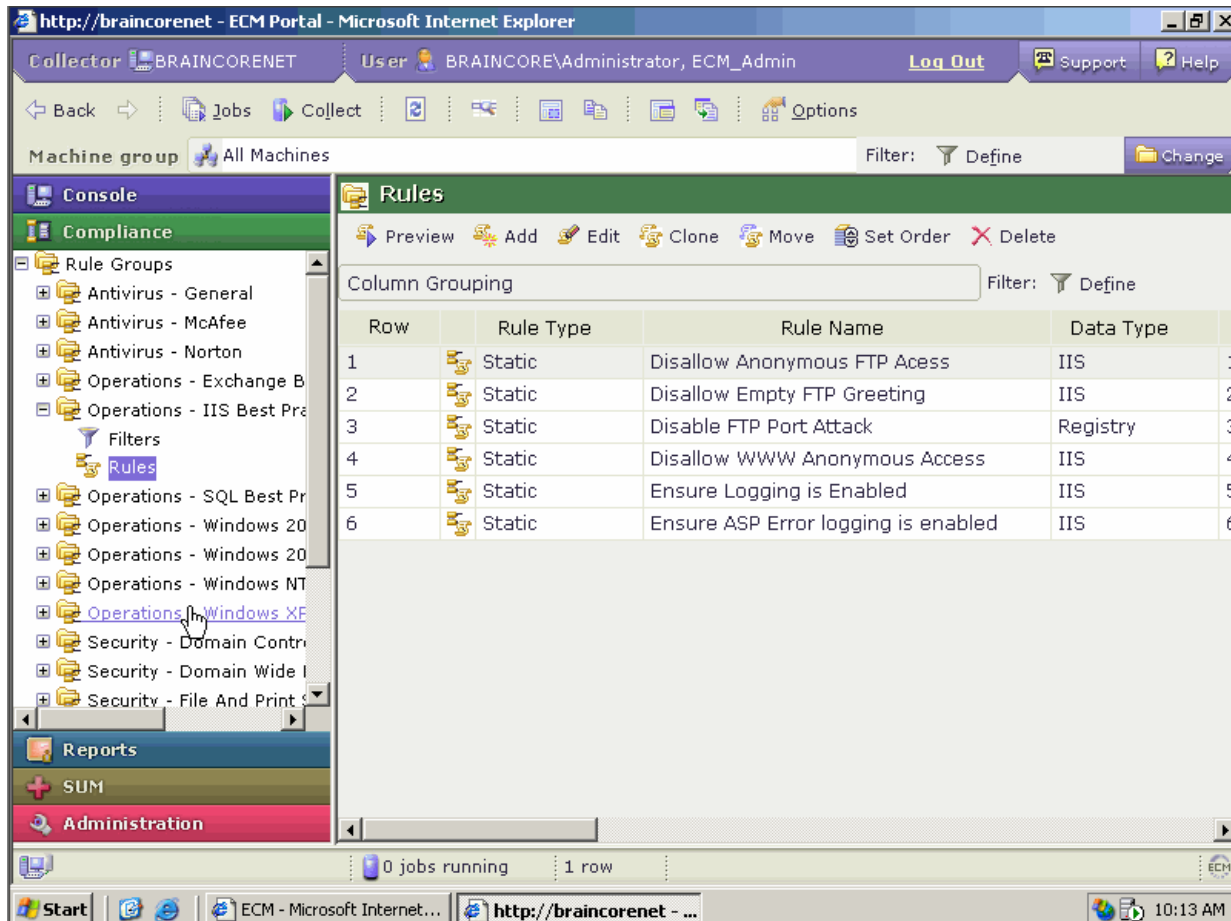
ECM's console also provides dashboards, which offer a quick view of key information such as OSs, antivirus status, and more.

 These dashboards utilize Java; thus, if you are running the ECM console on system with a later-version of the Microsoft Windows OS (such as WS2K3) that doesn't include a Java Virtual Machine (JVM), make sure you first install a compatible JVM. You can download and install the Sun Microsystems' JVM from <http://www.java.com>. Click the Download link at the top of the page to enter the automated download manager.

The other part of ECM is the Compliance section. The purpose of this section is to define *rules*, which describe your desired baseline configurations. You then create and run *templates* to see how well your rules are being enforced throughout your enterprise. Several templates come with the product, including a variety of Best Practices templates, a set of templates describing the SANS Institute's Securing Windows Guidelines, and more; you can immediately run these templates to see how well you're doing within these categories.

 ECM doesn't query managed systems in real-time when analyzing templates. Instead, it installs an agent on each managed system; that agent reports back thousands of configuration items to the central server's database. Running a template is, therefore, a very fast operation because the server simply analyzes the information in its local database.

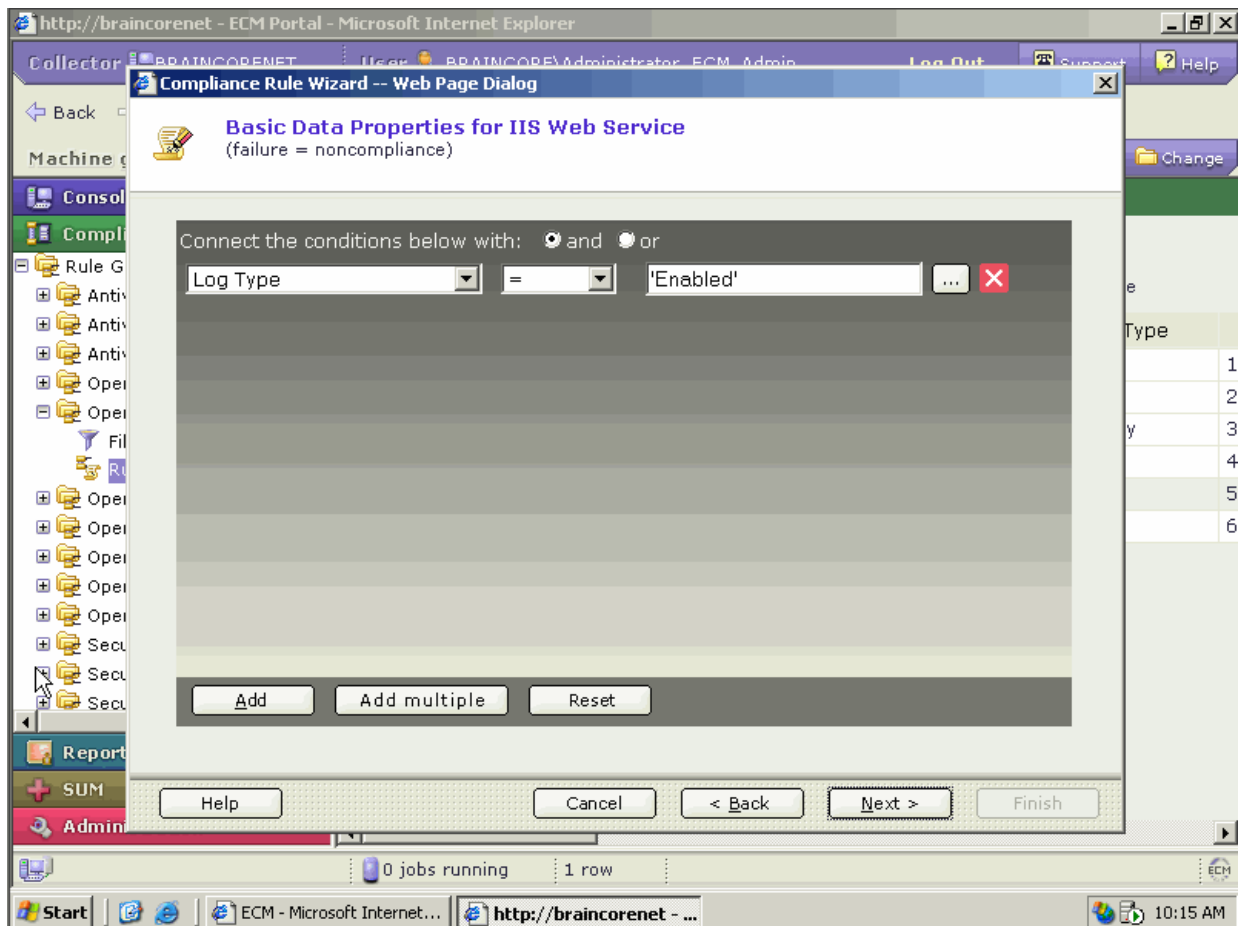
A huge variety of preconfigured rules also exist to help get you started. For example, Microsoft recommends that, for security reasons, logging be enabled on all IIS Web servers. As Figure 4.8 shows, ECM provides a rule for this best practice under its Operations—IIS Best Practices rule category.



**Figure 4.8: Rules for IIS Best Practices.**

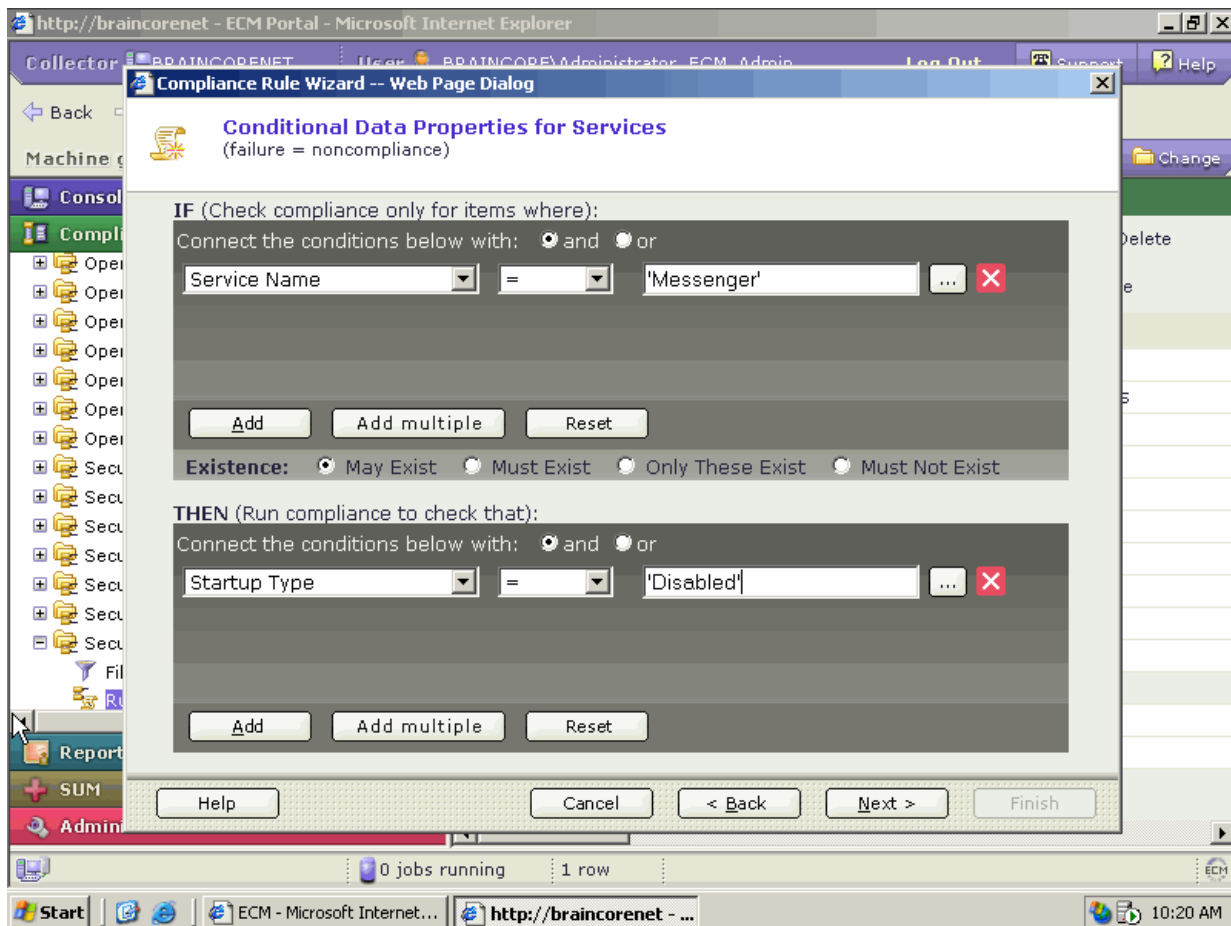
Examining this rule gives you an idea of the flexibility that the rules system incorporates. This particular rule uses a *basic* check, which means ECM can alert you to out-of-compliance machines but can't fix the problem for you. As Figure 4.9 shows, this basic check examines the Log Type and expects it to be Enabled.





**Figure 4.9: Examining a rule's configuration.**

Figure 4.10 shows how a *conditional* check can actually correct an incorrect configuration. In this case, the Messenger service is checked to ensure that its startup type is set to Disabled, ensuring that the service won't run on machines affected by this rule.



**Figure 4.10: Examining a conditional rule.**

Of course, ECM also provides fairly extensive reporting capabilities, allowing you to view change management reports, application information, security analyses, and more.

For more information about ConfigureSoft ECM, visit <http://www.configuresoft.com>.

### **Ecora Patch Manager**

Ecora Patch Manager is an agentless system designed to address enterprise patch management issues. An interesting feature of Patch Manager is its Test Center function, which allows you to deploy a patch to a single machine for testing prior to releasing the patch to your entire environment. Patch Manager also offers a patch rollback capability, making it possible to remove patches that are causing problems.

⚠ This feature is entirely dependent on the OS and patch to support uninstallation; many Microsoft patches do not support removal and Patch Manager won't make them removable.

Figure 4.11 shows Patch Manager's Missing Patches report, detailing the patches that are missing and the machines that are missing the patches. The primary purpose of this report is to alert you that patches haven't been fully deployed; the best action you can generally take from this report is to investigate the problem systems to determine why the patch won't install. Be aware that this report can contain a high number of false positives for some patches, often because the patch applies to software that isn't installed on the "problem" system.

Patch	Risk	Product Name	System	Note	OS Name	Scan Time
280380		SQL Server 2000 SQL Server 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
304404		Windows Media Player 6.4 for Windows 2000 Windows Media Player 6.4 for Windows 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
242479		Windows 2000 Advanced Server Windows 2000 Service Pack 4	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
823559	▲ MEDIUM	Windows 2000 Advanced Server Windows 2000 Service Pack 4	ECORAQA\BOSTON	ⓘ	Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
280419		Windows Media Player 6.4 for Windows 2000 Windows Media Player 6.4 for Windows 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
308567		Windows Media Player 6.4 for Windows 2000 Windows Media Player 6.4 for Windows 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
300635		MDAC 2.6 MDAC 2.6 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
823980	■ HIGH	Windows 2000 Advanced Server Windows 2000 Service Pack 4	ECORAQA\BOSTON	ⓘ	Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
298012	● LOW	SQL Server 2000 SQL Server 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
296138	▲ MEDIUM	Windows Media Player 6.4 for Windows 2000 Windows Media Player 6.4 for Windows 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
306908		SQL Server 2000 SQL Server 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
306908		SQL Server 2000 SQL Server 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
263968		SQL Server 2000 SQL Server 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
824105		Windows 2000 Advanced Server Windows 2000 Service Pack 4	ECORAQA\BOSTON	ⓘ	Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
299717	● LOW	SQL Server 2000 SQL Server 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
320920	▲ MEDIUM	Windows Media Player 6.4 for Windows 2000 Windows Media Player 6.4 for Windows 2000 Gold	ECORAQA\BOSTON		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
304404		Windows Media Player 6.4 for Windows 2000 Windows Media Player 6.4 for Windows 2000 Gold	ECORAQA\NEWYORK		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
242479		Windows 2000 Advanced Server Windows 2000 Service Pack 4	ECORAQA\NEWYORK		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
256052	● LOW	SQL Server 7.0 SQL Server 7.0 Gold	ECORAQA\NEWYORK		Windows 2000 Advanced Server	10/16/2003 4:35:17 PM
823559	▲ MEDIUM	Windows 2000 Advanced Server Windows 2000 Service Pack 4	ECORAQA\NEWYORK	ⓘ	Windows 2000 Advanced Server	10/16/2003 4:35:17 PM

**Figure 4.11: Ecora Patch Manager report.**

As with ConfigureSoft ECM, Patch Manager can alert you when new patches are released, or when a patch fails to deploy properly.


📖 For more information about Ecora Patch Manager, visit <http://www.ecora.com>.

### **Microsoft AD and Group Policy**

As I discussed in the previous chapter, AD and Group Policy can be used as a rudimentary configuration enforcement and software deployment solution. Their biggest weaknesses, compared with full-features systems such as SMS and ECM, is that they provide virtually no back-end reporting. There is no way to browse machines' existing configurations or check for machines that are out of compliance or haven't installed a software package. In addition, Group Policy is also capable of managing only a small set of configuration settings—hundreds out of the thousands that are available.

### **Microsoft SMS**

Microsoft SMS is an agent-based, highly scalable system primarily designed for software deployment and software and hardware inventory; a good asset management system, in other words. SMS can be used to deploy software packages to the results of a database query, and populates that database with a large amount of hardware and software inventory information from managed systems. Deployments can be tracked so that failed systems can be manually dealt with. SMS can be integrated with Microsoft SUS and WUS, allowing SMS to also deploy patches based on Microsoft releases. SMS also incorporates other tools that are useful to Help desks, including integrated remote control of managed systems, license management features, and so forth. SMS isn't strictly designed for change management. Although its database can track and compare different versions of a managed system's configuration, SMS was not primarily designed to provide notifications of changes.

 For more information about Microsoft SMS, visit <http://www.microsoft.com/sms>.

### **Microsoft Windows Update**

Windows Update is a free, Web-based service for Windows 98 and later computers. It was designed to scan systems for missing software updates and deploy those updates over the Internet. Win2K and later systems include an Automatic Updates client that can interact with Windows Update without user intervention, providing basic, automated patch management.

The forthcoming Windows Update v5 will finally extend Windows Update beyond the basic Windows OS to include updates for Microsoft server products, Microsoft Office, and other Microsoft software.

 To see Windows Update in action, visit <http://www.windowsupdate.microsoft.com> from any Windows 98 or later computer.

## Microsoft WUS and SUS

SUS and its forthcoming successor, WUS, provide a corporation with a private Windows Update server. Designed to interact with the Automatic Updates client on Win2K and later computers, SUS and WUS download updates from Windows Update, then deploy them across the corporate network. SUS provides only basic features, essentially deploying all approved patches to all systems; WUS will feature somewhat more granular management capabilities that include static grouping. WUS will also feature pre-deployment assessment reports and a more extensive reporting mechanism to track patch deployment.

 To learn more about WUS/SUS, visit <http://www.microsoft.com/windowsserversystem/sus>.

## Tripwire for Servers

Tripwire for Servers (TfS) is a change management system that is compatible with Windows servers. TfS' main function is to detect configuration changes on your servers, and to alert you to them, if desired. TfS is agent-based, making it scalable and allowing configuration comparisons to be run quickly from the central server's database.

 The product isn't specifically designed for use with Windows clients.

Figure 4.12 shows the main TfS console. As you can see, TfS examines a wide variety of configuration settings, including documents, the registry, and much more, giving you the ability to browse these settings from the central console.

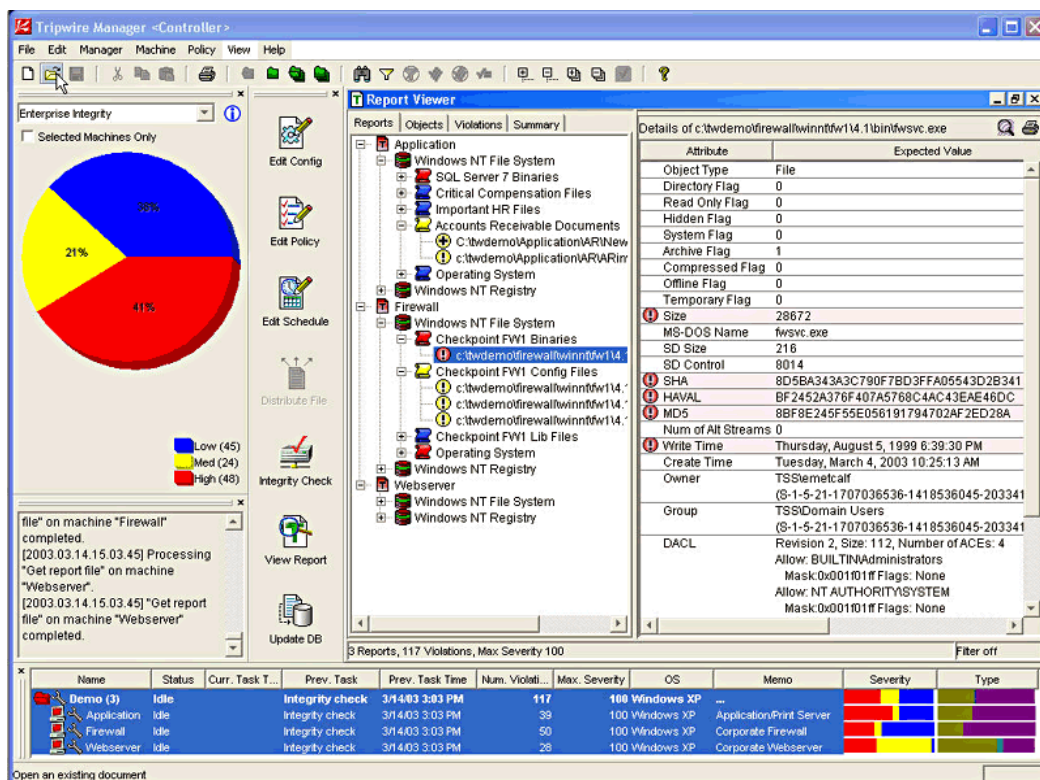



Figure 4.12: TfS console.

 To learn more about TfS, visit <http://www.tripwire.com>.

## Summary

In this chapter, I've laid out some clear goals for the tools that comprise your continuous configuration management solution. I've introduced you to some specific solutions from both Microsoft and third-party software vendors, and helped you understand what pieces of the continuous configuration management puzzle these solutions address. You should be able to better analyze your continuous configuration management requirements at this point, and select a set of tools that meets your needs.

Configuration management is becoming more important in today's enterprises. As I've mentioned, both patch management and configuration enforcement play vital security roles, and automated configuration management systems are quickly becoming the only realistic way to manage the growing complexity in IT environments. Configuration management is fast evolving from a "nice to have" feature for large enterprises into a mission-critical set of processes and procedures that every company with an IT investment needs to consider.