

realtimepublishers.com®

# *The Definitive Guide<sup>™</sup> To*

# Enterprise Network Configuration and Change Management

**VOYENCE™**

*Don Jones*

Chapter 4: The Scope of Change Management .....	65
Network Infrastructure Devices .....	65
Routers .....	65
Switches .....	68
Firewalls.....	70
Load Balancers.....	71
VPN Concentrators .....	72
Intrusion Detection Systems and Intrusion Prevention Systems .....	73
Wireless Access Points .....	74
Servers and Clients .....	75
UNIX- and Linux-Based Clients and Servers.....	75
Windows-Based Clients and Servers .....	76
Portable Client Devices.....	77
Tool Integration .....	78
Developing Appropriate Processes .....	79
Processes for Readily Managed Devices .....	79
Processes for Less Readily Managed Devices.....	81
Managing Without Help .....	82
Summary .....	84

## Copyright Statement

© 2004 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.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 Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

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.

Realtimepublishers.com and the Realtimepublishers 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 Realtimepublishers.com, please contact us via e-mail at [info@realtimepublishers.com](mailto:info@realtimepublishers.com).



## Chapter 4: The Scope of Change Management

Where can change and configuration management be applied in your environment? Given the broad range of devices and computers on the typical network, the ability to manage them all more effectively would be useful. However, various devices have differing levels of manageability. Thus, systems and network administrators must face the realities of change and configuration management. To help you do so, we'll explore several types of devices and computers, looking at how their configuration information is stored and managed and how change and configuration management can help you better administer each type of device.

### Network Infrastructure Devices

You probably know what a router is, what a switch does, and how firewalls work; you're also probably more than familiar with the management and configuration techniques for these devices. But to more effectively manage each device, we must look carefully at how these devices physically store their configuration data. The physical storage plays an important role in how a network configuration management solution can help manage these devices, and plays into the various technologies—which I'll cover in the next chapter—that make network configuration management possible.

 In Chapter 5, we'll explore the underlying technologies that make change management possible.

#### Routers

Almost all routers are implemented as standalone devices. Although it is possible to use a Linux or Windows server as a router, doing so is usually much less efficient than using a dedicated router from, for example, Cisco, Nortel, or 3Com.

Dedicated routers store their configuration information in what amounts to a simple text file, which is usually contained in the device's Flash Memory. This file is read when the device is started, and re-read when necessary (or on command) to pick up a configuration change. Listing 4.1 shows an example router configuration file.



```

!
version 12.1
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Lab-A-Atlanta
!
enable secret 5 $14Fmu45FsUOVz.WI5pFVqI.SHHnC/
enable password 7 030752180500
!
!
ip subnet-zero
no ip domain-lookup
ip host lab-b 201.100.11.2
ip host lab-c 199.6.13.2 199.7.14.2
ip host lab-d 204.204.7.2
ip host lab-e 210.93.105.2
ip host switch-a 205.7.5.9
ip host switch-d 210.93.105.9
!
ipx routing 0001.960e.f3c0
!
!
interface Ethernet0/0
description connected to Lab-Hub-A-Atlanta
ip address 192.5.5.1 255.255.255.0
full-duplex
ipx network 1A encapsulation SNAP
no mop enabled
!
interface Serial0/0
description connected to Lab-B-Boston
ip address 201.100.11.1 255.255.255.0
ipx network 1C
clockrate 2000000
!
interface Ethernet0/1
description connected to Switch-A-Atlanta
ip address 205.7.5.1 255.255.255.0
full-duplex
ipx network 1E encapsulation SNAP
no mop enabled
!
interface Serial0/1
no ip address
shutdown
!
router rip
version 2
redistribute connected
network 192.5.5.0
network 201.100.11.0
network 205.7.5.0
no auto-summary
!
```



```

ip classless
ip http server
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
tftp-server flash:
snmp-server community public RO
snmp-server location Lab-Atlanta
snmp-server contact Joe,555-1212,joe@company.com

banner motd #Welcome to Lab-A-Atlanta#
!
line con 0
exec-timeout 0 0
password 7 094F471A1A0A
login
transport input none

line aux 0
password 7 070C285F4D06
login

line vty 0 4
password 7 01100F175804
login
!
End

```

***Listing 4.1: An example router configuration file.***

This file is easy to read because it is simply a string of text commands; it isn't stored in some proprietary binary format. It also contains useful information. For example, the lines

```

!
hostname Lab-A-Atlanta
!
```

tell you that the router's hostname is Lab-A-Atlanta. A network configuration management solution can parse this information and use it to populate a detailed inventory list. In addition, information such as the following lines

```

tftp-server flash:
snmp-server community public RO
snmp-server location Lab-Atlanta
snmp-server contact Joe,555-1212,joe@company.com
```

play a crucial security role in your network. These lines define the SNMP community string, which, in this case, is set to the default "public." The router is configured only to allow reading of SNMP information—designated by the RO; however, the use of the default community string enables an attacker to easily read information about your network from the router, thereby making subsequent attacks more effective.



 One benefit of a network change management solution is to detect security vulnerabilities and either call them to your attention or disallow them completely.

Although this information isn't stored in a proprietary format, it can be difficult to quickly ascertain such security vulnerabilities simply by looking at the device's long configuration file—the sheer length of the file can make details like this easy to miss. One of the values offered by an enterprise network configuration solution is that the tool is automated and tireless, so it can spot these problems for you and alert you to them.

Most routers are managed via Telnet (or SSH) sessions and include a feature that writes the router's configuration to a Trivial FTP (TFTP) server. Network configuration management solutions can easily log on via Telnet (or SSH) and issue the TFTP commands; management solutions can even host a TFTP server so that the router writes its configuration directly to the management solution for analysis. Because router administration is so simplistic, it is easy for a network configuration management solution to help automate and control that administration.

 Telnet, SSH, and TFTP all present potential security concerns, particularly from a "defense in depth" viewpoint. I'll discuss these concerns along with the protocols themselves in Chapter 5.

## Switches

Managed switches are similar to routers in the way that they work. For example, Listing 4.2 shows a sample configuration file from a Cisco 2924XL switch.

 Network configuration management solutions can't usually be employed for unmanaged switches because unmanaged switches contain no user-definable configuration settings.

```
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch_A
!
enable secret 5 $1$8rc/$4h8lW2f1F3mPpo2LTDAD41
enable password xxxxxxxxxxxx
!
!
ip subnet-zero
!
!
interface FastEthernet0/1
description VLAN 1 Central Station
duplex full
!
interface FastEthernet0/2
description VLAN 1 to Linksys Switch
duplex full
```



```

!
interface FastEthernet0/3
description VLAN 1 to Router Atlanta
duplex full
!
interface FastEthernet0/4
description VLAN 2
duplex full
port group 2
switchport access vlan 2
!
interface FastEthernet0/5
description VLAN 2
duplex full
switchport access vlan 2
!
interface VLAN1
ip address 192.1.1.4 255.255.255.0
no ip directed-broadcast
no ip route-cache
!
ip default-gateway 192.1.1.1
snmp-server engineID local 0000000902000002FDDDB700
snmp-server community private RW
snmp-server community public RO
snmp-server chassis-id 0x0E

banner motd #Welcome to Switch_A#
!
line con 0
password xxxxxx
login
transport input none
stopbits 1

line vty 0 4
password xxxxxx
login

line vty 5 15
password xxxxxx
login
!
End

```

**Listing 4.2:** A sample configuration file from a Cisco 2924XL switch.



This configuration file isn't meant to be complete or usable; I'm showing it as an example of how straightforward and easy-to-understand a switch's configuration file usually is. You can see how such a simply-presented set of information can be more easily handled by third-party management solutions, which can readily analyze them for configuration inconsistencies, maintain a version history of the configuration, and much more.



Although similar in appearance to a router's configuration file (largely as a result of the fact that both are from Cisco devices), the configuration file for a switch has a number of unique settings. Thus, although the switch's configuration is just as easy to retrieve as a router's configuration, by using Telnet and TFTP, a configuration management solution must be able to specifically recognize switches in order to be able to parse the configuration file. The file itself contains key data that allows this recognition to occur, such as the switch's OS version number and the switch's manufacturer and model number. The file can also reveal security vulnerabilities, which can be difficult to spot. Can you see any?

As a hint, let's take a closer look at the SNMP strings:

```
snmp-server engineID local 0000000902000002FDDDB700
snmp-server community private RW
snmp-server community public RO
snmp-server chassis-id 0x0E
```

This switch defines two strings, the default “public” and a not-too-imaginative “private,” which is probably the first string an attacker will try after “public.” These long, detailed files make it difficult to catch these problems—as with routers, the files are simply so large that, even though they are easy to interpret, it's easy to overlook the details. In addition, it is pretty rare for administrators to even scan through these files on a regular basis; most administrators tend to just modify one or two settings at a time. This situation reinforces the value of a network configuration management solution—it will never miss these problems.

The configuration of a switch is stored in a basic text file, making it easy to retrieve, easy to analyze, and easy to store in a database for change-management purposes. Text files offer additional advantages for comprehensive configuration management solutions. For example, the solution can provide its own text editor for editing or creating configuration files, then push those files out to devices. If the configuration files were in a more complex format, it would be more difficult for a configuration management solution to offer an in-solution configuration editor.

## **Firewalls**

Firewalls fall into one of three categories: their configuration files look a *lot* like the ones I've shown you for a router and a switch, their configuration files look *very* different than those we've already explored, or they don't have configuration files at all.

Standalone firewalls—so-called “black box” firewalls (such as Cisco's PIX products)—are managed devices similar in nature to routers. In fact, they *are* routers with a lot of extra port-filtering and security capabilities. Using these firewalls with a configuration management system is typically straightforward because these firewalls have the same Telnet management interface and TFTP capabilities that a router or switch has.

Other firewalls are built on a more general-purpose platform, such as a UNIX, Windows, or Linux server. These firewalls—such as Microsoft's Internet Security and Acceleration (ISA) Server or Checkpoint's Firewall1 product—are software-based and sometimes offer capabilities beyond those offered by standalone, hardware firewalls such as a PIX box. However, the complexity of software firewalls makes it very difficult, if not impossible, to incorporate them into a configuration management system.



ISA Server, for example, stores much of its configuration information in the Windows registry, and additional information in Microsoft's Active Directory (AD). Although it is quite possible to remotely read and change this information—AD, for example, supports the standard Lightweight Directory Access Protocol (LDAP) for doing so—as a practical matter, there are no tools for managing ISA Server apart from those that come with it. You are unlikely to find a third-party solution that can manage both your routers *and* a software firewall.

The ability to be incorporated into an enterprise network configuration management solution should be part of the selection criteria for you when considering products for your network. Software firewalls such as those from Checkpoint and Microsoft might be less desirable if your intention is to have a network that can be centrally managed from a single configuration management solution.

 Software-based firewalls aren't inherently unmanageable; it is just that the technologies required to do so are much more complex than the basic Telnet-and-TFTP combination used by so many hardware-based network devices. Those technologies are continuing to evolve, in fact; although Microsoft, for example, has long supported Remote Procedure Calls (RPCs) for remote registry manipulation, the company is now beginning to move to Windows Management Instrumentation (WMI) for such activities. The continuing evolution of these technologies make it difficult for third-party vendors to incorporate them into their management products, especially when compared with protocols such as Telnet and TFTP, which have existed virtually unchanged for nearly two decades.

### **Load Balancers**

Load balancers, like firewalls, fall into categories: hardware-based and software-based. Software-based solutions such as Microsoft's Network Load Balancing (NLB) service, included with Windows Server 2003 (WS2K3) and other Microsoft products, are practically impossible to manage through third-party configuration management solutions. NLB stores its configuration in the Windows registry and doesn't lend itself as readily to external remote management as hardware-based load balancers. Although more established software-based load balancers such as those from Resonate are much more easily managed by their own or third-party management products.

Like routers and switches, hardware-based load balancers are much easier to incorporate into a centralized configuration management system. The F5 BIG-IP load balancer, for example, is a hardware-based unit that is managed primarily through Telnet and has TFTP capabilities similar to a traditional switch or router.

 Logging and notification are also important aspects of manageability. Most enterprise-class network devices—including routers, switches, firewalls, load balancers, and so forth—offer this functionality. These capabilities allow a device to write log entries—using technologies such as Syslog, TACACS+, or RADIUS—whenever something happens on the device. Devices may also be able to send SNMP traps notifying a central management station of certain events that have occurred on the device. Through these events—whether logged or sent via SNMP—configuration management solutions discover that something might have changed on a device, triggering a re-scan of the device's configuration and an analysis to determine what changed. Further actions might include notifications to administrators or even a complete backup of the device's configuration to capture the change.



## VPN Concentrators

VPN concentrators are often implemented as hardware devices, meaning they follow the fairly typical management methodology of Telnet-and-TFTP that you are, by now, familiar with. This methodology makes concentrators as easy to incorporate into a network configuration management solution as is incorporating routers and switches. Consider the partial configuration file for a Cisco VPN 5002 concentrator that Listing 4.3 shows.

```
[ General ]
Password          = hello
DeviceName        = vpndevice

[ Domain Name Server ]
PrimaryServer     = 209.165.201.29

[ Time Server ]
Enabled           = On
ServerAddress     = 209.165.201.30
BindTo            = WAN 0:0

[ Logging ]
Level             = Debug
LogToAuxPort      = On

[ Link Config WAN 0:0 ]
Mode              = FrameRelay

[ IKE Policy ]
Protection         = MD5_DES_G1

[ IP WAN 0:0 ]
# This interface connects to the Internet
Mode              = Routed
IPAddress         = 209.165.201.2
SubnetMask        = 255.255.255.224
RIPVersion        = V2
Numbered          = On
PointtoPointFrame = On
InterfaceDLCI    = 17

[ Radius ]
# This RADIUS server is for accounting for all users
PrimAddress       = acct.nsp.com
Secret            = accountingsecret
BindTo            = WAN 0:0
Accounting        = On
PrimVSAAcct      = On

[ IP Static ]
# Default route
0.0.0.0 0.0.0.0 209.165.201.3 1

[ Context List ]
flash://companyA.cfg
flash://companyB.cfg
```

***Listing 4.3: A partial configuration file for a Cisco VPN 5002 concentrator.***

Although this file is markedly different from the Cisco switch and router configuration I listed earlier, it is nonetheless a straightforward easy-to-handle text file that can be easily supported by a third-party configuration management solution. VPN concentrators also produce prodigious log files, primarily to account for user activity. The following example log file, which Listing 4.4 shows, illustrates how straightforward this information is in its presentation and formatting.

```
50235 10/23/2002 17:44:00.930 SEV=4 IKE/52 RPT=1217 201.214.18.178
Group [VPNUser] User [DOMAINX\userx]
User (DOMAINX\ userx) authenticated.

50236 10/23/2002 17:44:01.760 SEV=5 IKE/184 RPT=1215 201.214.18.178
Group [VPNUser] User [DOMAINX\userx]
Client OS: N/A
Client Application Version: 3.5 (Rel)

50238 10/23/2002 17:44:37.610 SEV=4 IKEDBG/65 RPT=397 201.214.18.178
Group [VPNUser] User [DOMAINX\userx]
IKE TM V6 FSM error history (struct &0x4c5db3c)
<state>, <event>:
TM_DONE, EV_ERROR
TM_WAIT_QM_MSG, EV_TIMEOUT
TM_WAIT_QM_MSG, NullEvent
TM SND_REPLY, EV SND_MSG
```

**Listing 4.4: An example log file.**

These logs can, of course, include information about when administrators log on and off of the concentrator. An administrator logging off is a clue that a configuration change might have occurred; by scanning these log files for changes, network configuration management solutions can use the logoff event as a trigger to pull the device's configuration and look for any changes the administrator might have made.

However, not *all* concentrators are hardware-based. Microsoft Windows, for example, includes Routing and Remote Access software that can serve as a VPN concentrator but is much less easy to manage by using external utilities. Also, some hardware-based concentrators aren't *managed* and are simply "dumb" terminals that accept incoming VPN connections and place them onto your network. These lower-end concentrators are typically found in smaller environments and might only support a handful of user connections. Be aware of the manageability of these devices when considering them for your network.

### **Intrusion Detection Systems and Intrusion Prevention Systems**

Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are among the newest types of devices being found on today's networks. Rather than acting as active components of the network, such as a firewall—which creates a more secure environment by actively limiting traffic—IDSs and IPSs typically act in a more passive role, sitting on the sidelines and scanning traffic, log files, and other data for evidence of an attack. For example, the popular Nokia IP330 or the Econet Sentinel both monitor network traffic. In the case of an IDS such as the Nokia, you receive warnings and notifications when suspicious activity is detected; in the case of an IPS such as the Sentinel, the device can actually block attackers' computers from accessing your network once an attack is detected.



These systems are so new that few standards have emerged for management and maintenance. Almost all support SNMP and logging capabilities (that being one of their primary functions), and most enterprise-class systems support management through Telnet. However, not all of them have configurations simple enough to support TFTP transfers of a text-based configuration file, although some do. Enterprise network configuration management products are still evolving to support IDSs and IPSs, just as the IDSs and IPSs themselves are evolving to better protect networks from attack.



Complete configuration management via SNMP is certainly possible, particularly when a device comes with an SNMP Management Information Base (MIB) that describes the device's capabilities. However, retrieving or restoring a configuration via SNMP—which essentially requires configuration parameters to be queried or changed one at a time—is rarely practical for complex devices such as IDSs and IPSs. I'll describe this limitation in more detail when I describe SNMP in the next chapter.

### Wireless Access Points

Wireless access points (APs) are one of the newest types of network devices on today's enterprise networks. There are two distinct classes of devices found even on very large networks: enterprise-class APs, which are *intended* for corporate use, and consumer-class APs, which are definitely *not* intended for corporate use. Although most consumer APs sport attractive, easy-to-use configuration menus from an embedded Web browser, these APs are not manageable in the enterprise sense of the term, and cannot be integrated into typical network configuration management solutions.

Enterprise-class APs, however, are quite manageable. Many offer a Web-based configuration interface. In addition, many support more traditional device management through Telnet and provide TFTP capabilities for uploading and downloading configuration files to and from the device.



Wireless APs are one of the most important classes of devices, aside perhaps from firewalls, to ensure that you have under centralized configuration management as quickly as possible. Other devices, such as routers and switches, don't provide the easy access to your network that a wireless AP does. Certainly, a misconfigured router can open IP ports and expose your network to attack over the Internet; a misconfigured AP, however, can make it possible for nearly anyone, even relative computer amateurs, to access your network backbone. Because centralized configuration management solutions offer more consistent device configuration and can detect unauthorized configuration changes, they make APs a safer part of your enterprise network.



You should attempt to always work with wireless APs' configurations over their wired connection, addressing them by their wired interface IP address, if necessary. Protocols such as TFTP and Telnet present sufficient security disadvantages on their own, without throwing in the additional security risks of wireless networking, which is basically broadcasting data to anyone who can pick it up and decrypt it. Although wireless security is definitely improving (especially with standards such as Wi-Fi Protected Access—WPA—and Advanced Encryption Standard—AES), there is no reason to take chances. Keep your configuration information traveling over inherently more secure physical wires whenever possible.



## Servers and Clients

Servers and clients are becoming a more important part of the enterprise network. Servers, in particular, offer key services to users that make servers every bit as important as infrastructure devices such as routers. In fact, if you run DHCP, DNS, and other infrastructure services on a server, that server is very much a part of your network's foundation. Even clients play an important role, particularly from a security standpoint, because they can be a launching point for Trojan horse-based attacks on other network components. Yet many organizations don't consider servers and clients when creating a network configuration management strategy.

There's a good reason, of course: Clients and servers can't be as easily managed as routers and switches, as I'll discuss in the following sections. But just because they can't be easily managed doesn't mean they should be ignored when it comes to configuration management.

 The following sections explore the difficulty in managing clients and servers. Later in the chapter, I'll discuss network change and configuration management strategies for even these difficult-to-manage devices.

### ***UNIX- and Linux-Based Clients and Servers***

Apache, for example, is probably the most popular Web server software run on the UNIX or Linux platforms. It uses normal text files for its configuration. The main configuration file, httpd.conf, looks something like the one that Listing 4.5 shows.

```
# server settings
Port 80
User httpd
Group httpd
ServerType standalone
ServerRoot /usr/local/apache
ServerAdmin user@domain.com
ServerName linux
ServerSignature EMail

## house-keeping files
LockFile /var/run/httpd.lock
PidFile /var/run/httpd.pid
ScoreBoardFile /var/run/httpd.scoreboard

## logs

LogLevel warn
LogLevel debug
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined"
LogFormat "%h %l %u %{[%Y%m%d %H%M%S %Z]}t %v \"%r\" %>s %b
 \"%{Referer}i\" \"%{User-Agent}i\" common"
CustomLog /var/log/apache/access.log common
ErrorLog /var/log/apache/error.log

## connection settings

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
```



```

KeepAliveTimeout 15
MinSpareServers 1
MaxSpareServers 10
StartServers 1
MaxClients 150
MaxRequestsPerChild 30

```

**Listing 4.5: Httpd.conf, the main configuration file for Apache.**

Unfortunately, Apache can have several other files: module.conf, host.conf, srm.conf, access.conf, and more. In addition, Apache add-ons can have their *own* configuration files. For example, the popular JServ add-on uses jserv.properties, jserv.share.properties, and more. It's almost impossible for a network configuration management solution to know which files to look for without administrator intervention, and any software not specifically called out to the solution won't be covered by configuration management. What's more, the OS doesn't provide any kind of real-time notification when these files are changed, making it tougher to detect unauthorized configuration changes. In fact, you're pretty much limited to frequent, automated scans of the files to detect changes. Even scanning the files can be difficult, as the OS doesn't make the configuration files available over the network by default; you must configure your servers with a Telnet or other utility in order to make the configuration files available. Worse yet, no two administrators configure their servers alike, making it even more unlikely that an automated network configuration management solution would be able to find the information it needed.

### **Windows-Based Clients and Servers**

Windows-based computers aren't any easier to centrally manage than UNIX- or Linux-based servers, at least with regard to configuration management systems. Windows stores most of its information not in a text file or collection of text files, but in a proprietary database known as the Windows registry. The registry contains most of the configuration information for Windows itself, as well as for any installed applications, such as Web servers.

For example, consider the following excerpt from a Windows registry file:

```

Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EventSystem]
"Configured"=dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EventSystem\{26c409cc-
ae86-11d1-b616-00805fc79216}]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EventSystem\{26c409cc-
ae86-11d1-b616-00805fc79216}\EventClasses]

```

This file *looks* like it would be fairly straightforward for a configuration management system to work with. After all, it's a normal text file, not unlike the configuration files of many routers, switches, and other devices. However, this format isn't the native storage format for the registry; that format is a proprietary binary format that isn't especially easy to read, and the file itself is even less easy to actually access. To create this easy-to-use text file, you have to use the Windows' Registry Editor tool and export the registry to a text file. This process is manual and can't easily be automated.



Windows also stores information in AD, Microsoft's LDAP-based enterprise directory. Many Windows-based applications, such as Microsoft Exchange Server, also store configuration information there. Information in AD is easier to access than information in the Windows registry—via LDAP—but complicated in nature and organization, making it very difficult to perform point-in-time comparisons for change management purposes.

Other configuration information—such as the settings for IIS or Microsoft SQL Server—is contained in separate databases. The sheer variety of ways that Windows applications and the OS itself use to store configuration information makes it nearly impossible to select a single solution that offers configuration and change management for the entire platform. Most applications do, of course, come with console software that can be used to centrally manage multiple servers. For example, SQL Server's Enterprise Manager application makes it relatively easy to manage multiple servers from one workstation. However, these consoles rarely provide any proactive configuration management, version control, change management, or other higher-level management functions.

Windows does provide a rudimentary means of centralized configuration for Win2K and newer computers in an AD domain. Called Group Policy, this technology allows centralized specification and control of many client and server settings. However, Group Policy doesn't, by default, provide any change management, workflow capability for planning changes, and so on. Third-party products are available that can provide a degree of change management over Group Policy and Microsoft's own Systems Management Server (SMS) product provides a moderate amount of configuration tracking and notification, although it provides nothing in the way of proactive configuration control.

### ***Portable Client Devices***

The proliferation of portable devices—Palms, Windows-based digital assistants, smart cell phones, tablet computers, and so on—presents an interesting new configuration management problem. Although these devices rarely act in any sort of network infrastructure mode, they are nearly always wireless-capable, meaning they offer a potential point of entry into your network. Managing the security settings of these devices is critical, but these devices are the least suited for configuration management of any kind. They provide no standardized central control, management, or configuration.



### Servers and Clients: Is This Network Management?

Traditionally, systems and network administrators haven't considered servers and clients part of an enterprise's network configuration management strategies. I disagree with this philosophy primarily because the network and servers and clients work together to form a useful business tool. The network by itself is useless; similarly, servers and clients aren't much fun without a network to tie them together. Having a highly managed network that never goes down is great, but if your servers are constantly down as a result of poorly managed change, that robust network isn't doing you much good. Thus, configuration management is important at *all* levels.

Servers and clients are not inherently unmanageable. They're certainly more complex in terms of configuration than most network devices, but specialized software packages exist to help make server and client configuration management possible. However, because servers and clients tend to be so complex, their configuration management solutions tend to be platform-specific, which contrasts with the latest generation of network configuration management solutions, which often allow management of multiple vendors' network devices within a single solution.

In addition, there is no technical reason why server and client management can't be made less complex. Software could be written that outputs the Windows registry and other configuration repositories into a text file (perhaps formatted in XML), which could then be sent via TFTP to a configuration management solution—much like network devices are able to do. That same software could read in a text file to make configuration changes, enabling features such as configuration rollback and so forth.

## Tool Integration

Consider how a configuration management solution can fit into the overall scope of your enterprise network management framework. Many configuration management tools can, for example, work with the same SNMP traps, Syslog files, and other technologies that enterprise frameworks, such as Hewlett-Packard OpenView, support.

The scope of configuration management can be greatly increased by integrating configuration management tools with existing network management frameworks. Few vendor-provided tools—such as Microsoft Management Consoles (MMCs)—offer any “big picture” integration features that address the entire network; third-party tools may offer the ability to forward SNMP traps to a management workstation (or receive forwarded traps from a workstation), automatically create alerts when changes are detected, and so forth. These additional features often give third-party solutions a broader reach than vendor-provided, product-specific solutions.

Configuration management tools that *don't* offer some kind of integration with a larger management framework might actually impair network management. For example, if you have to make a decision between sending SNMP traps to a configuration management tool or a network management workstation, you'll be limiting the scope of one or the other. Integrated tools eliminate the need to make these decisions and compromises.



## Developing Appropriate Processes

As I've mentioned earlier in this guide, processes are the key to configuration management. A well thought-out process that can help reduce downtime, increase maintainability, and so forth. Tools such as network configuration management solutions help enforce these processes and provide automation capabilities that make your processes reasonable to implement on an ongoing basis. For example, most administrators would agree that backing up device configurations on a nightly basis is a useful practice; tools make it possible to do so without an undue administrative burden. Network managers generally agree that a set of configuration templates is a good idea for ensuring consistency across devices; tools make templates physically possible, even in large environments.

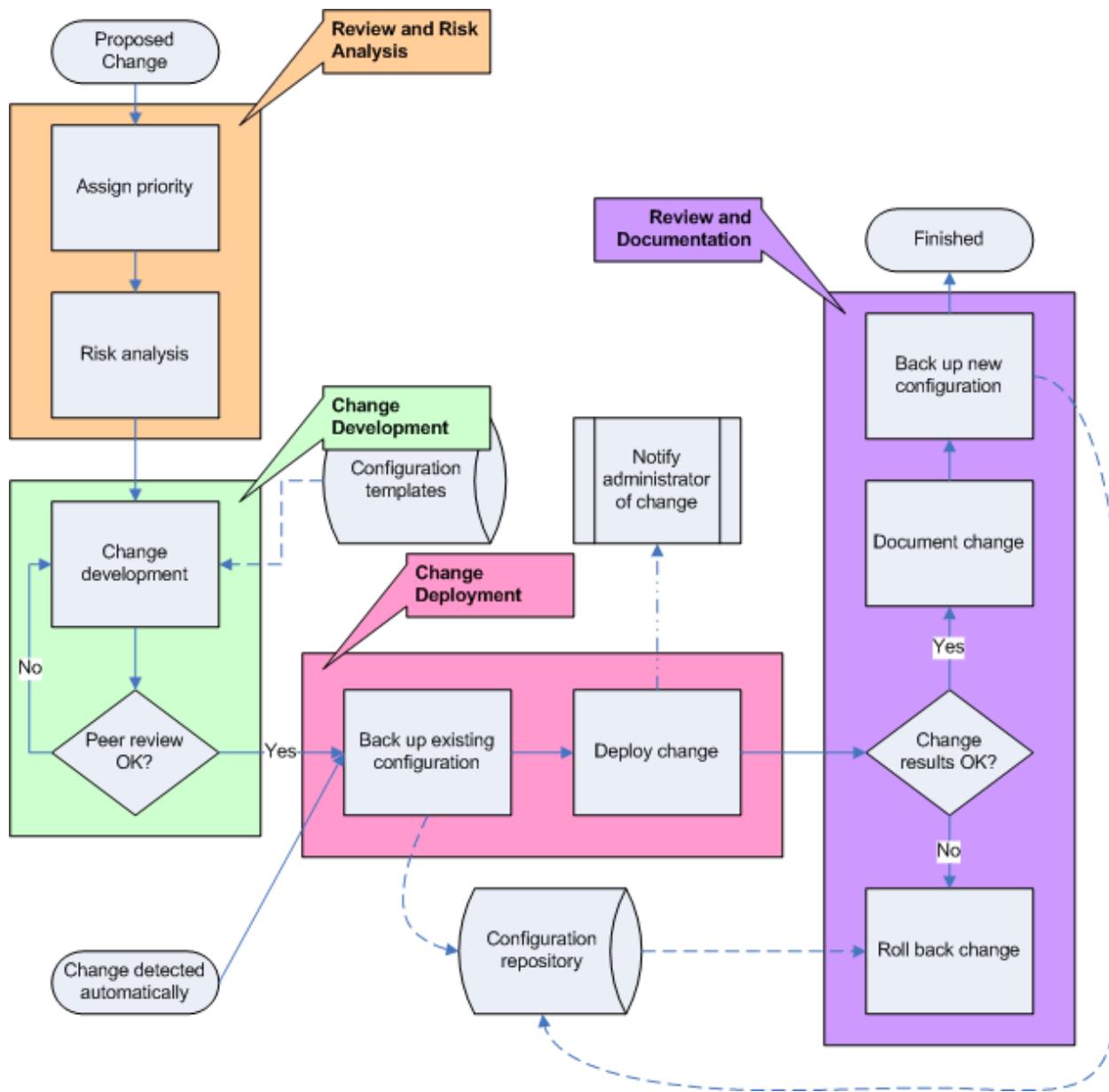
Throughout this chapter, I've described several instances in which devices' nature easily accommodate managed change, such as routers and switches. I've also discussed aspects of your network that might not be so readily managed, such as Windows or Linux servers. You should still make an effort to manage configuration changes on *all* of your devices. Tools aren't available to help you manage *every* device, so you'll need to develop different processes for those devices that don't accommodate managed change.

In the next two sections, I'll provide some brief examples of processes that work well with devices that can be readily managed by a network configuration management solution. I'll then contrast that process with one intended for less easily-managed devices.

### **Processes for Readily Managed Devices**

Consider the basic configuration management process pictured in Figure 4.1. The illustrated process is fairly standard—similar to those I illustrated in previous chapters. In this figure, I've used colored blocks to group the distinct phases of the process.





**Figure 4.1: Process for readily managed devices.**

This process is fairly complex and relies on the fact that a tool—a network configuration management solution, to be specific—will handle much of the work. The first part of the process, Review and Risk Analysis, probably occurs outside the tool. The next part, Change Development, can probably occur *within* the tool—the right tool will include configuration templates and a workflow engine that enforces the process itself.

During change deployment, the tool can automatically ensure that the existing configuration is backed up and that the change is deployed on schedule. Administrators can be notified that the change has occurred, allowing them to follow up and check the results of the change. If there's a problem, the change can be rolled back and effectively undone; if the change is okay, it can be documented and backed up.

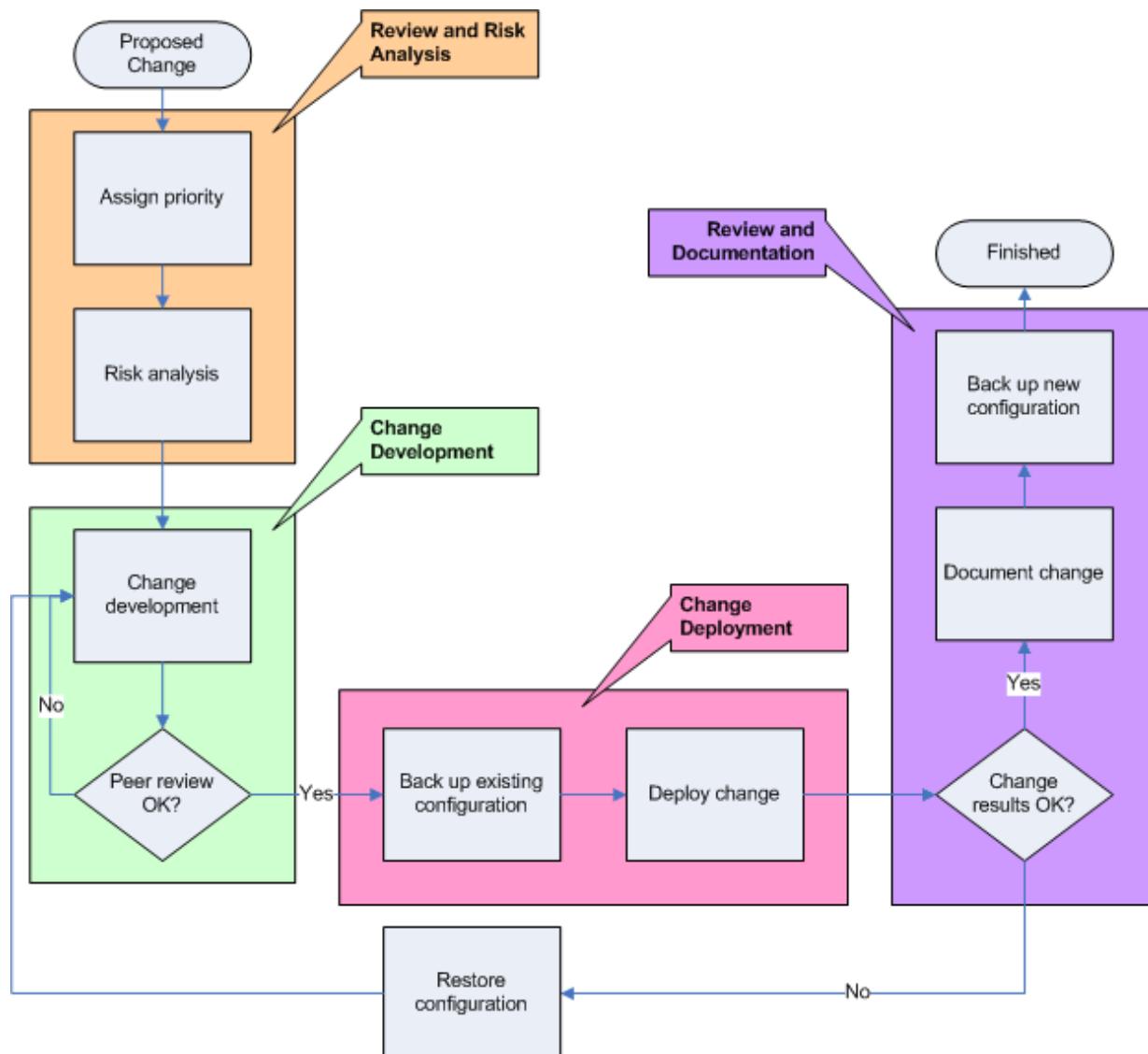


The tool can also provide another entry point into the process by automatically detecting changes. The detection of a change can result in an immediate backup and notification, allowing administrators to ensure that the change was authorized. Unauthorized changes can be readily rolled back.

This process involves many steps, but most are handled automatically (or made much easier) by the tool. This process is suitable for devices such as routers, switches, firewalls, and so forth—devices that lend themselves readily to a network configuration management solution.

### **Processes for Less Readily Managed Devices**

Devices that aren't as readily managed should, in theory, have the same process; after all, it's the process that is resulting in the benefits of reduced downtime and so forth. However, processes such as the one that Figure 4.1 illustrates are often too complex for administrators to actually follow in their daily routines; thus, it is wise to develop a process that is easier to work with when tools aren't available to handle the hard work. Consider the process in Figure 4.2.



**Figure 4.2: Process for less readily managed devices.**



The first section is simpler, as it's pretty much a human-based process. The second part appears similar to the process in Figure 4.1, but configuration templates aren't available. The peer review is an important step, but a significant risk is that there is nothing really enforcing it; in the previous process, the tool could *prevent* unreviewed changes from being deployed. In this instance, adherence to the process is more on the honor system. Change deployment is less sophisticated, too; although it is easy to back up the configuration of most servers and applications, there is no central, version-controlled repository in which to do so. Standard backups are the best alternative.

If a change is deployed that causes problems, the only option is to restore from a previous backup—a process often much less automated than a configuration rollback offered by network configuration management tools.

In addition, this process doesn't incorporate any notification for changes, whether authorized or not. There is no tool to detect unauthorized changes, and typically no means for delaying or scheduling the release of a change; changes occur immediately when they're made.

This process isn't *bad*; it just suffers from a lack of automation and enforcement. However, when dealing with devices that can't be readily incorporated into a central configuration management system, this type of process is often the best you can do. Complex devices such as Windows or Linux servers simply don't offer the features and technologies necessary for real configuration management.



Third-party software solutions exist that can help automate some or all of these steps. However, these solutions are almost by necessity platform-specific, working not only with a particular OS but generally with a specific *version* of that OS. Thus, completely automating configuration management in a heterogeneous environment could mean deploying half-dozen different configuration management solutions. This is one reason why the term "network management" is often held to include only network devices: They're simply easier to manage because they're more standardized and less complex.

### **Managing Without Help**

It is not a foregone conclusion that you'll use a network management solution. For example, smaller networks might rely on manually performing network configuration management tasks without a specific software solution. In such cases, IT staff will use basically the same process that Figure 4.2 shows except that they won't have any tools to help accomplish each step. Figure 4.3 expands on this process, noting the manual steps that must be taken to implement configuration management and points out potential risks that a network change management solution would catch automatically.



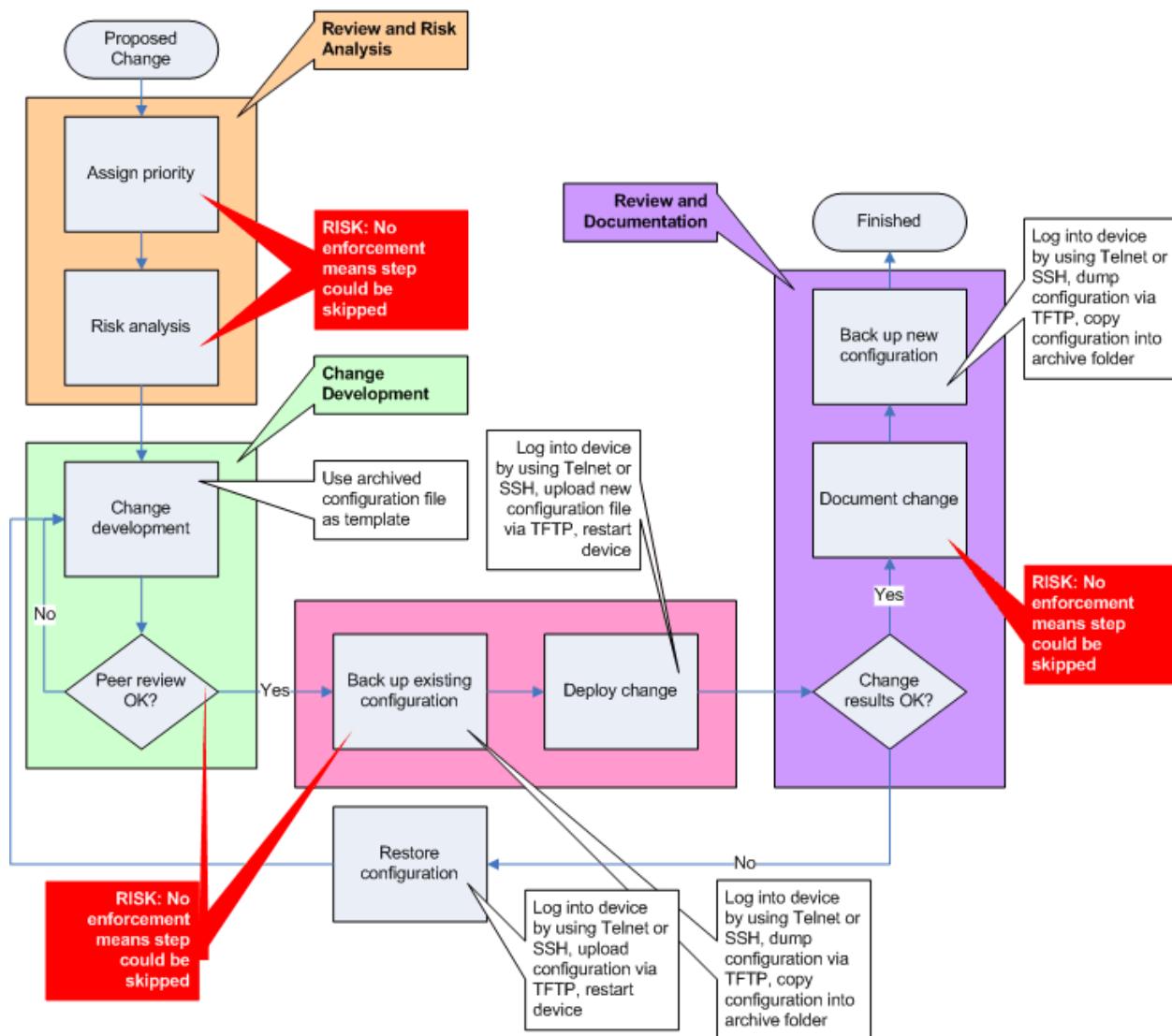


Figure 4.3: Manual configuration management process with procedures highlighted.

As you can see, performing network configuration management without a software solution is certainly possible, although it is more difficult and is not a feasible option in a larger environment with hundreds of devices. Significant business risks remain with a manual process because the very steps designed to help reduce downtime—peer review and change documentation, for example—are the steps most easily skipped by busy, overworked administrators who are simply trying to keep things working smoothly. The value of a configuration management system is twofold:

- Automation—A change management solution will obviously be a welcome tool for those overworked administrators.
- The enforcement of key business process steps—This benefit generates the business advantage of configuration management by helping to reduce or eliminate downtime as a result of improper changes, poor documentation, and so forth.



## **Summary**

In this chapter, I've covered how various devices and components on your network store their configuration information, and how that information is—or is not—accessible to external configuration management solutions. With a better understanding of which devices can be readily managed, you will be able, when possible, to select those types of devices for use on your network. I've also provided a sample process that illustrates how manageable and less-manageable components—such as servers—can be brought under a primitive form of configuration management by having administrators adhere to well-designed processes. Of course, nothing beats a solid configuration management tool for simplifying and automating those processes when the devices under management lend themselves to it.

In the next chapter, I'll dive into network configuration management technologies. I'll discuss the technologies that are available in detail, including under-the-hood information about how each one works and how it can contribute to an enterprise configuration management process.

