SCCM

Patch Management

Ed Roth • John Savill • Orin Thomas
CONTENTS

The 4 Pillars of System Center Configuration Manager 3

Built on 4 Pillars 3
Core SCCM Features 4
What You Need to Know 5
Installation Considerations 6
Client-Deployment Considerations 7
Other Caveats 7
Worthwhile Investment 7

Solve 4 Common Patch Management Problems 8

Determine Which Updates Have Been Deployed 8
Prevent Update Traffic from Saturating WAN Links 9
Prevent Updates from Interrupting End Users 9
Test Updates 10
Acceptance Is the Key 10

SCCM Patch Management FAQs 11
Microsoft has christened System Center Configuration Manager (SCCM) 2007 as the new incarnation of its vaunted System Management Server (SMS). The System Center moniker acts as an umbrella that covers Microsoft's family of manageability tools. Along with Configuration Manager, the current list of System Center solutions includes Operations Manager, Data Protection Manager, Reporting Manager, Essentials, Virtual Machine Manager, and Capacity Planner. The company also recently announced a new Help desk offering called System Center Service Manager (SCSM). But SCCM is the senior member of the System Center lineup, and it’s arguably the anchor component.

Let’s take a look at SCCM’s architecture and the solid set of tools it provides for managing your entire Windows infrastructure, highlighting some of the new and exciting features of SCCM 2007. Then, let’s drill down into what you need to know about putting the new generation of Microsoft systems management software to work in your environment.

**Built on 4 Pillars**

SCCM is a major retooling of previous SMS technologies and capabilities. In its introduction of the new product, the Microsoft product team uses an analogy of four pillars upon which the new system is built. The pillars are *simplicity, deployment, security,* and *configuration.*

**Simplicity.** The simplicity pillar represents a worthwhile goal for a product with so many capabilities. Toward this end, Microsoft has rolled feature packs and add-ons into the core product so that administrators no longer need to find, download, and integrate such tools individually. A new setup routine tracks and displays setup tasks as they occur and builds a management point so that the SCCM installation is ready to begin client deployment following setup. Microsoft has also introduced the notion of maintenance windows and integrated Wake on LAN (WOL) capabilities, both of which let SCCM administrators more easily control when and how the tool’s operations occur on managed systems. The Microsoft Management Console (MMC) 3.0-based UI, which Figure 1 shows, gets some terrific enhancements, including drag-and-drop and search folders. Microsoft has streamlined many administrative tasks with dynamic wizards to reduce the complexity of operations. Another great new feature—Volume Shadow Copy Service (VSS)—enabled backups for SCCM site systems—further simplifies administrators’ lives.

**Deployment.** The deployment pillar focuses on making SCCM a complete solution for deploying both server and desktop OSs throughout the enterprise, in addition to applications and updates. These capabilities have existed in some fashion in SMS 2003, but Microsoft has redesigned them to integrate the latest Windows OS deployment technologies—such as Windows Pre-installation Environment (PE), Windows Imaging Format (WIM), and User State Migration Tool (USMT)—into an unattended OS deployment process. The product uses a task-sequencing engine during the deployment process to ensure that necessary steps (e.g., installing drivers and applications, restoring user documents and settings) occur.

**Security.** The security pillar is primarily composed of two security initiatives that make SCCM a better tool for managing security updates for your enterprise and make the SCCM infrastructure more secure than previous SMS versions. The first initiative involves enhanced vulnerability assessment and remediation technology, and the second initiative involves seamless, end-to-end, mutual authentication between SCCM systems and managed clients—whether they’re connected via the Internet or on the LAN or roaming between the two.
Configuration. The configuration pillar entails giving IT organizations the ability to model and manage a desired configuration for a given system type. SCCM administrators can create management policies to establish a baseline for system-configuration items, including hardware configuration, installed software, system load, and specific settings. The system can report on compliance with the baseline configuration and can take knowledge-driven actions based on particular out-of-compliance conditions.

Core SCCM Features
Total cost of ownership (TCO) was once a huge driver for promoting tools to better manage IT systems, but the term TCO seems to have fallen out of vogue. However, we should never underestimate the necessity of keeping the cost of managing desktop and server systems in check. IT organizations are responsible for maintaining a healthy TCO bottom line.

That’s where SCCM comes in. SCCM is geared toward increasing the overall effectiveness of IT organizations, streamlining provisioning, and managing computing resources while minimizing the overhead of doing so. The following core SCCM features all contribute in the effort of accomplishing these lofty goals: software distribution, inventory and reporting, device management, OS deployment, software update management, remote tools, desired configuration management, network access protection, and Internet-based client management.

Software distribution and updates. Software distribution is a huge part of SCCM and has been since the first version of SMS. Software distribution is the ability to remotely deploy software—typically an application—to one or more client systems. That summation sounds simple enough, but modern businesses’ software-deployment needs reach far beyond simply installing a given software package onto a group of desktop computers. Attention must be paid to a target system’s connection type, system type, and usage pattern, as well as the overall bandwidth of the network you’re using for delivery. Furthermore, once you’ve installed a software package, it will likely need updates over the course of its service life. You can use collection machine variables—which help you categorize computers based on certain parameters (e.g., OS, memory, disk)—to ensure that SCCM targets only appropriate systems for certain software. Background Intelligent Transfer Service (BITS) and maintenance windows ensure that software installation doesn’t hamper a user’s productivity. If an uncooperative user insists on powering off his or her system each night, you can use WOL to power it on for software maintenance. SCCM uses binary deltas—with DFS replication (DFSR) hashing—to minimize the bandwidth impact of application updates for sites and distribution points across your network. (A binary delta copies only changed bits of an application update. For example, if you have a 700MB Microsoft Office package and you need to change one file, only the differences in that file will need to be transferred for the entire package to be current—as opposed to the entire 700MB package.)

Inventory and reporting. Even small IT shops can have trouble getting a clear picture of the hardware and software assets that comprise their fleet. SCCM’s inventory and reporting features help with this challenge. You can configure the inventory component to collect hardware and software information from client systems at a prescribed interval. The reporting component then assembles appropriate pieces of the collected data into meaningful reports. These reports can be quite simple (e.g., a breakdown of desktop computer platforms) or quite complex (e.g., HP laptops in the accounting department with a specific BIOS version and video driver version, running Microsoft Internet Explorer—IE—7.0 on Windows XP SP2). Software-inventory and software-metering reports can also help you get a firm grasp on license management.

Device management. Device management—which Microsoft really should call mobile device management—originated as a feature pack add-on to SMS 2003. The company has enhanced the feature and incorporated it into SCCM. Device management lets you perform on mobile devices management functions similar to those available to traditional clients. For example, you can perform hardware and software inventory, file collection, software distribution, settings control, and password management. Current SCCM-manageable devices include those running Windows Mobile software on Pocket PC, or smart phones. The SCCM documentation—accessible from the Learning Path—contains an exhaustive list.

OS deployment. SCCM’s OS-deployment capabilities add up to a dramatically enhanced version of the SMS 2003 feature pack add-on and solution accelerator. These new core functions are based on OS deployment technologies in Windows Server 2008 and Windows Vista. Using the OS deployment tools, you can build a
reference machine and capture a single image of it for deployment to an entire enterprise. SCCM supports such deployment scenarios as bare-metal installations, in-place upgrades, and machine-to-machine migrations.

**Software update management.** SCCM leverages Windows Server Update Services (WSUS) as the underlying technology for updates and patches. However, you’ll use the SCCM interface to wield enhanced control over the approval and application of updates. Additionally, SCCM’s update-management features give you a means with which to deploy updates from third-party and internal software providers and—for the purpose of compliance—allow for tracking and reporting of updates applied throughout your enterprise.

**Remote tools.** The ability to remotely control managed systems has been a long-standing, useful SMS feature for troubleshooting and providing end-user support. Microsoft has revamped SCCM’s remote tools so that, by using Vista’s RDP protocol, they realize the benefits of improved performance, security, and richer collaboration technologies. SCCM also still supports Remote Desktop and Remote Assistance.

**Desired configuration management.** Every IT organization recognizes the benefits of standardizing systems and configurations. SCCM’s desired configuration management component—previously an SMS 2003 solution accelerator, now enhanced and integrated into SCCM—lets you define a model for the configuration of a certain class of system. SCCM will then monitor managed systems for compliance according to that definition.

**Network access protection.** Microsoft’s Network Access Protection (NAP) is an entirely new feature in SCCM. In simple terms, NAP is a tool for monitoring your network for noncompliant, potentially vulnerable systems, and proactively correcting any potential compliance problems before permitting such systems network access. However, NAP implementation requires Windows Server 2008 to be running Network Policy Server. NPS policies measure system compliance, and SCCM’s NAP performs any required remediation.

**Internet-based client management.** Although SMS has traditionally managed many types of clients—including desktops, laptops, and servers—the ability to manage portions of the client population connected via the Internet has been lacking. SCCM has incorporated secure Internet-based management capabilities into the core feature set. Using public key infrastructure (PKI), clients can securely participate in traditional software deployments, inventory schedules, and other SCCM functions while connected only via the Internet.

**What You Need to Know**

Now, you’re probably wondering what else you need to know before taking the SCCM plunge—either as a new deployment or as an upgrade to an existing SMS installation. For new deployments, the first thing you need to consider is the size and complexity of your environment, and whether you require and can benefit from SCCM’s extensive management capabilities. If you read my beta review of System Center Essentials (see the Learning Path), you might remember that tool’s limit of 30 servers and 500 client systems. Those numbers also serve as a reasonable point at which implementing SCCM starts to make sense: If you have fewer than 500 systems, you might not benefit from the robust, complex beast that is SCCM. If you have an existing SMS implementation, an upgrade to SCCM should be on your radar at release time. After you make the decision to move to SCCM, you’ll want to spend some time on two preparatory steps, involving PKI and site system roles.

**PKI.** Of primary concern, if you don’t have an existing PKI implementation, you’ll need to learn about the technology and deploy PKI to support SCCM’s advanced security features. PKI is a requirement for native-mode deployments (i.e., full deployments of SCCM clients and required servers) because the system uses a site server signing certificate to sign all SCCM policies. Through this infrastructure, site systems and managed clients establish mutual trust.

**Site system roles.** Your next area of study is site system roles. SCCM offers numerous new roles and dispenses with or renames a few old ones. Although adding new roles might seem to contradict the goals of the simplicity pillar, Microsoft has designed the roles to help you better manage and maintain your SCCM infrastructure and managed systems.

As you see in Figure 2, the SCCM 2007 system roles are primary site server, site database server, Configuration Manager console, branch office distribution point, fallback status point, management point,
PXE service point, reporting point, server locator point, software update point, state migration point, and system health validator. Note that not all roles are necessary, and each role doesn’t need to reside on a dedicated server. In fact, for very small implementations, it’s feasible—but not recommended—that all required roles reside on one server. Your determination of appropriate roles and supporting hardware will be a factor of your environment’s workload and security requirements. You can find many planning aids for SCCM deployment in Microsoft’s Configuration Manager Documentation Library (see the Learning Path), which can help you come up with the right mix of roles and hardware.

Two new roles of note are branch office distribution point and fallback status point. A branch office distribution point (which replaces the old secondary site role) can be a Vista or XP system. This system can hold software applications and updates for distribution to a branch office. SCCM utilizes BITS technology to initially populate and apply delta changes to software on branch office distribution points. SCCM uses the fallback status point as a catchall for communications from managed systems that have somehow become orphaned from their intended management point. This system role is instrumental in discovering and fixing client-reporting problems in your fleet.

Installation Considerations
You’ll want to become familiar with the various installation options available to you, depending on your current situation.

If you’re installing a brand-new SCCM 2007 site, you have two options—simple setup and custom setup—although the simple setup isn’t very useful unless you’re deploying for test purposes. SCCM’s Setup Wizard checks for prerequisites (as Figure 2 shows), helps you mitigate any software deficiencies, then walks you through the process of specifying site and managed system parameters. If you already know exactly how you want to deploy SCCM, you can streamline this process by using the scripted installation option.

![Figure 2: The SCCM 2007 system roles](image)

If you’re upgrading an existing SMS 2003 site, you have a number of options, decisions, and prerequisites to consider. First, before you can add SCCM to the mix, your SMS 2003 site must be running SMS 2003 SP2. Second, SCCM doesn’t support Windows 2000 servers, so you'll need to upgrade any SMS systems running on that OS. Third, you need to decide whether you’ll use a side-by-side or in-place upgrade strategy.

Organizations that aren’t heavily invested in their current version of SMS will find the side-by-side upgrade acceptable. This upgrade amounts to bringing up the new SCCM site, then reassigning and upgrading existing managed systems to the new site. More probable though, is an in-place upgrade. An in-place upgrade migrates your existing data to the new database schema and lets you run in an interoperable mode while you convert to SCCM 2007. One caveat is that the upgrade process removes any unsupported feature packs—particularly those for...
OS deployment and device management. However, although the upgrade removes the legacy feature packs, their functionality is replaced natively in SCCM 2007, and the new SCCM-native features will use the settings previously configured for the feature packs.

When you upgrade, you should go from the top of your hierarchy down. One helpful tip is to consider placing a central SCCM 2007 site above your existing SMS 2003 primary site, then let your data flow up. Using this scenario, you can familiarize yourself with the new SCCM console while using your own data. From the SCCM 2007 console, you can view—but not edit—SMS 2003 site settings. You can upgrade secondary SMS 2003 sites to SCCM 2007 manually, by pushing them via SMS, or by installing them through remote control. You can assign SMS 2003 clients to SCCM 2007 sites, and SCCM 2007 clients—in mixed mode—can roam back to an SMS 2003 site for interoperability.

**Client-Deployment Considerations**

You can assign SCCM clients based on AD OUs so that the assignment strategy can be more aligned with the structure of your business than an SMS site structure. In addition to standard push-client installations and software distribution methods, there's a new way to perform client installation. Using the Software Update Point, you can piggyback on your WSUS implementation to overcome client-installation obstacles such as account permissions and unopened ports. When Microsoft releases SCCM to manufacturing, the company will provide an .adm template for distributing SCCM client settings via Group Policy.

Microsoft has also made notable improvements to the SCCM client-installation executable. The tool uses a single binary file—ccmsetup.exe—for all client installations. The new executable has bandwidth awareness through BITS, and it downloads a simple XML manifest first to determine which components are applicable to a given client, then downloads and installs only what is necessary.

**Other Caveats**

SCCM's native mode and the PKI infrastructure it requires are requirements for Internet-based client management. Also, you're going to have to modify your AD schema to use NAP, but that prospect isn't as scary as it might sound. If you're comfortable with it, you can run the ExtADSch.exe file (from \SMSSETUP\BIN\i386) on the SCCM 2007 installation media, or you can use a Microsoft-provided LDF file. The LDF file documents the classes and attributes added in the process of modifying the schema, as well as the SCCM features they're associated with. (NAP is one such feature that requires an update to the AD schema.)

**Worthwhile Investment**

Microsoft’s investment in its four-pillar strategy of simplicity, deployment, security, and configuration should pay dividends for IT organizations ranging in size from medium to huge. Existing SMS users will benefit greatly from an upgrade to Microsoft’s latest and greatest configuration management tool, and SCCM’s new capabilities and usability add up to a compelling argument for deployment in many IT organizations where previous versions of SMS might not have made the cut.
Update management is a task IT professionals approach with the same enthusiasm they usually reserve for a visit to the dentist. Ensuring that computers are up-to-date is tedious, and at the back of every IT professional’s mind is the fear that lurking somewhere deep within a newly released patch is code that will cause more problems than it solves. You need to balance thoroughly testing updates before deploying them with the knowledge that code that targets the vulnerabilities these updates address usually appears on the Internet within a week of the update’s release. Spend too long pontificating on the adverse impact of applying an update and you’ll become vulnerable to the exploits the update protects you against.

In this article, you’ll learn about several patch-management-related problems and the steps you can take to mitigate them. The particular annoyances this article covers are as follows:

- Determining which updates have already been deployed
- Preventing update traffic from saturating WAN links
- Preventing update installation from interrupting end users’ computer use
- Testing updates before deployment

This article focuses primarily on problems related to managing updates for Microsoft OSs and applications. Managing updates for third-party products without using a tool such as Microsoft System Center Configuration Manager (SCCM) 2007 poses even more challenges.

### Determine Which Updates Have Been Deployed

As more computers in organizations become mobile, administrators have more difficulty determining whether a particular update has been deployed on every computer or just on some of them. Back when I worked in first-level support, it was easy to keep track of which updates had been installed because we had to install them manually and would cross computers off a central list as each computer was done. When updates deploy automatically over the network, it’s more difficult to track whether updates have deployed successfully, unless you use solutions such as SCCM 2007.

Most organizations use Windows Server Update Services (WSUS) to manage the deployment of OS updates, as well as updates for Microsoft applications. When a configured computer contacts the WSUS server to obtain and download updates, the WSUS server records which updates computers have obtained. Computers can contact the WSUS server according to a schedule or the connection can be initiated manually. The drawback to WSUS is that although it records which updates computers have obtained, it doesn’t actually check the client to see if any updates are missing, and it can be hazy on whether the update that was obtained has actually installed correctly.

WSUS knows only about updates it has provided; it has no way to know if an update has been installed in another way. For example, what if a laptop user spends a few weeks away from the office and uses Windows Update through the Internet, rather than WSUS, to keep her computer up-to-date? WSUS doesn’t know about these updates because it’s aware only of updates it distributes, not updates obtained from other locations. WSUS also knows only about computers that have reported to it. It’s possible for WSUS to be completely unaware of computers on your network because, for some reason, those computers have never been able to successfully contact the WSUS server.

There are two free solutions to help with the problem of knowing which updates are installed, so you don’t have to manually check each computer to see whether a specific update is installed and you don’t have to deploy a solution such as SCCM 2007 that can be a drain on your IT department budget. The first free tool you can use to check computers for missing updates is the Microsoft Baseline Security Analyzer (MBSA), which you can download from Microsoft’s website. The latest version, MBSA 2.1.1, supports scanning Windows 7 and Windows Server 2008 R2 computers. You can use MBSA to check for all missing updates from a list published by Microsoft or against the list of updates that you’ve approved on a WSUS server. It would be nice if the
MBSA tool’s functionality were included with WSUS so that update deployment and checking could occur from a single console, but Microsoft currently has no plans to combine these two products.

Instead of the MBSA tool, you can use Get-Hotfix, a new cmdlet included with Windows PowerShell 2.0. Get-Hotfix lets administrators query computers locally or remotely to determine which hotfixes are installed. Using PowerShell scripting, you can query a list of computers to determine whether an update or a specific list of updates is missing. For example, the following script scans all computers listed in the file computers.txt and adds the names of computers missing the hotfix indicated by the identifier KB974332 to a text file named Missing-KB974332.txt:

```powershell
get-content computers.txt | foreach
    { if (!(get-hotfix -id KB974332 -computername $_ -ea 0))
    { add-content $_ -path Missing-KB974332.txt }}
```

Although the code appears on multiple lines here, you would enter it all on one line. The bit of code -ea 0 sets the error action to silent, preventing the command from producing error text during execution.

**Prevent Update Traffic from Saturating WAN Links**

Organizations deploy solutions such as WSUS not only to centralize the deployment of updates but also to minimize the amount of update traffic downloaded from the Internet. Rather than 1,000 computers downloading a 100MB update, one WSUS server can download the update and then distribute it to all computers on the network. This process works fine until you take into account branch offices and saturated WAN links. Just as you generally don’t want to have all your computers downloading the same large update from the Internet, you don’t want to have 100 computers at a branch office all downloading the same update from the head office WSUS server across a low-bandwidth WAN link.

You can configure WSUS so that it hosts only the update approval list and not the update files, forcing WSUS clients to download update files from the Internet, but this means that all WSUS clients, not just those in branch offices, source their update files from Microsoft’s update servers. In many organizations, the solution to this problem has been to configure separate WSUS servers at each branch office location, with branch office clients obtaining updates from their local WSUS server. However, adding WSUS servers adds to administrative overhead. It’s possible to configure WSUS servers in an upstream/downstream relationship so that updates approved on one server are automatically approved on another, but every server you add to your infrastructure will increase costs in some way.

The solution for branch office computers is to leverage a new technology called BranchCache in conjunction with Windows’ existing Background Intelligent Transfer Service (BITS) peer caching functionality. BranchCache is a new feature for computers running Windows Server 2008 R2 and Windows 7 (Enterprise or Ultimate editions). BranchCache lets clients at branch offices share content automatically with each other when they obtain that content from an appropriately configured remote server. BITS peer caching is an existing Windows networking technology that can work in concert with BranchCache to make update transfer across the network more efficient.

You configure BITS and BranchCache through Group Policy. BITS and BranchCache policies are found in the Computer Configuration\Administrative Templates\Network node of a Group Policy Object. You can leverage BranchCache with WSUS only if the WSUS role is installed on a computer running Windows Server 2008 R2 and the client computers are running Windows 7 Enterprise or Ultimate edition.

The advantage of using BranchCache and BITS with WSUS is that organizations can use a single WSUS server to deploy updates to head office and branch office networks without saturating branch office WAN links with update traffic. Updates are retrieved across the link by one branch office client, then shared with the other clients at that location. This has the advantage of a local branch office WSUS server without the additional administrative overhead. You can find out more about BranchCache in the Microsoft article "Server Configuration."

**Prevent Updates from Interrupting End Users**

When it comes to scheduling the deployment of updates, you want to avoid the scenario in which a user who has a document open leaves his computer for a short amount of time and returns
to find the computer has restarted itself due to the installation of an automatic update. Users generally want their computers to restart only if they initiate the restart themselves. They really dislike computers that, from their perspective, seem to require restarts on an arbitrary basis.

I once had to spend several hours updating a manager’s computer because a previous administrator had allowed the manager to choose whether to accept or reject updates after the manager had lost several hours work due to an unexpected restart caused by the installation of an automatically scheduled update. Needless to say, the manager had declined all future updates, so the computer was several service packs behind where it should have been at that point in time!

Some of the randomness of update installation can be mitigated through configuring update-related Group Policies. The Enabling Windows Update Power Management to automatically wake up the system to install scheduled updates policy, in conjunction with the Configure Automatic Updates policy, lets administrators configure computers to wake from hibernation at a preconfigured time for the installation of updates. This system lets computers wake themselves for the purposes of update installation at 3:00 a.m., for example, when no sane user should have a document open. This method requires that the computer has a BIOS that supports waking from hibernation.

If you choose this method, you should also configure policy so that the default shutdown action is to hibernate the computer rather than to power off the computer. You can accomplish this by blocking end users from being able to power off the computer, then configuring the power settings policy to automatically hibernate the computer after a reasonable period of inactivity.

Test Updates
Every administrator fears applying the update that breaks something. An update that causes so many problems that it necessitates a complete OS reinstall are rare. Today, most updates that fail don’t do so in a spectacular and obvious manner. Failures, when they occur, are subtle. Administrators are unlikely to find a fault soon after installing the update on a test computer. People who use the OS or applications in day-to-day situations are more likely to find faults than those who only have a passing familiarity with them.

This situation makes it difficult for administrators to know whether deploying an update will cause a problem. Just because a problem isn’t immediately obvious doesn’t mean it isn’t serious. Vendors have released updates that caused data corruption that wasn’t apparent to administrators through typical testing, but end users discovered the trouble two days after the update was rolled out to every computer in the organization.

Administrators need a way for typical end users to test updates without deploying the update to every user in the organization. One solution is to have a group of users that function as update testers and to deploy the update to these testers a week before generally deploying the update more widely across the organization. In theory, the testers will encounter problems before the update is introduced to everyone. Inconveniencing one or two testers is less problematic than inconveniencing everyone. If testers can’t find a problem in a week’s worth of typical computer use, any problems that the update causes probably won’t be serious.

The main difficulty in recruiting testers is that testers need only one bad experience and they might be unwilling to test anymore: Someone who loses a day’s work is less likely to volunteer to be a guinea pig in future. Users in the IT department don’t make good testers because they rarely use applications in the same way that other employees in the organization do. When assembling a test group, you might need to find a way to reward the users, which probably requires the support of your management. Explain to management why you need a group of reliable testers but that also, from time to time, these testers will lose working time because of something unforeseen happening because of updates on their computers. Better, though, that a small number of users lose time than the whole organization suffers downtime because an update that causes a problem gets deployed to everyone without undergoing any local testing.

Acceptance Is the Key
The main way to reduce the annoyance of the patch-management process is to accept that it will always be necessary and that the best way to deal with it is by being organized. Although patch management will never be something that IT professionals eagerly anticipate, following the advice in this article can reduce these specific annoyances to minor irritations.
SCCM Patch Management FAQs

**Q. Can I deploy updates to non-Microsoft applications using System Center Configuration Manager (SCCM)?**

**A.** Microsoft has released System Center Updates Publisher 2011, which lets you deploy and track third party and in-house application updates using SCCM 2007, 2012 Beta 2, or System Center Essentials 2007. Now you can use SCCM to manage updates for all applications within your organization, the same technologies that have enabled rich update management for Microsoft OSs and applications.

**Q. How do I enable Windows Server Core Automatic Updates?**

**A.** You can push patches to your computers by using solutions such as Microsoft System Center Configuration Manager (SCCM) 2007, or you can configure automatic updating through Group Policy. If you want to locally enable Server Core Automatic Updates, use the following command to enable Windows Update’s normal checks (you must be in the \Windows\System32 folder):

```
Windows\System32>cscript scregedit.wsf /au 4
```

The output would be:

Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Registry has been updated.

Then you would enter the command:

```
Windows\System32>net stop wuauserv
```

The output would be:

The Windows Update service is stopping.
The Windows Update service was stopped successfully.

Next, enter:

```
Windows\System32>net start wuauserv
```

The output would be:

The Windows Update service is starting.
The Windows Update service was started successfully.

You can also force Windows to perform an update check using the `wuauclt /detectnow` command. To check the state of automatic update, use the `cscript scregedit.wsf /au /v` command.

**Q. How do I create the Systems Management container in Active Directory (AD)?**

**A.** By default, when you extend the AD schema for Microsoft System Center Configuration Manager 2007 (SCCM), it doesn't actually create the Systems Management container or set permissions. Fortunately, the process to create the container isn't too complicated.

1. Log on as a domain administrator.
2. Launch ADSI Edit (adsiedit.msc).
3. Open the domain partition, expand the domain name, and right-click the CN=System container. Select New then Object in the context menu.
4. Select the container type, enter a name of `System Management`, and press Next then Finish.
5. Now you need to give the SCCM server full permissions on the System Management container and its child objects.
6. Still within ADSI Edit, right-click the System Management container and select Properties.
7. Select the Security tab.
8. Click Add, select the computer account of the SCCM server, and grant it Full Control.
9. Edit the new permissions. Under the Advanced section, click Edit for the SCCM account. Change the Apply onto field to `This object and all child objects`.

---

Brought to you by Windows IT Pro
10. Click OK.

Where can I get the SCCM Toolkit?

The address for the SCCM Toolkit will change over time as newer versions are released. As of this writing, the latest version for SCCM 2007 (version 2) is available here. The easiest way to find it is to use Bing to search for SCCM 2007 Toolkit.

If you use SCCM, it’s highly recommended that you use this toolkit. One tool you’ll use a lot is Trace32, which makes the SCCM logs easy to read. The first time you launch Trace32, it will prompt if it should be used as the default viewer for SCCM logs. I suggest you say Yes, as shown here.

I’m trying to deploy Forefront Endpoint Protection (FEP) using System Center Configuration Manager 2007 (SCCM) using a custom advertisement, but it isn’t working. Why not?

When the FEP server components that integrate with SCCM are installed, a package, a program, and an advertisement are automatically created to enable deployment of the FEP client, which can be set to download to a cache then execute. If you create your own advertisement that allows installation from the distribution point, the FEP client install will fail.

Because FEP is a security component, it must be installed from the local cache, so make sure you enable Download content from distribution point and run locally under the Distribution Points tab of the advertisement to distribute the FEP client in SCCM.

<table>
<thead>
<tr>
<th>General</th>
<th>Schedule</th>
<th>Distribution Points</th>
<th>Interaction</th>
<th>Security</th>
</tr>
</thead>
</table>

Each boundary in the Configuration Manager site is designated as a fast (LAN) network or a slow or unreliable network. Specify how to run the content for the advertised program, depending on the type of boundary the client is connected to.

When a client is connected within a fast (LAN) network boundary:

- [ ] Run program from distribution point
- [x] Download content from distribution point and run locally