Lucent NPS
The knowledge behind the network

Lucent Technologies
Bell Labs Innovations



Software Distribution in Windows 2000 Environments

By Homer Wu, Network Systems Engineer

Contents

CONTENTS 1

INTRODUCTION 3

WHAT IS INTELLIMIRROR? 4

SOFTWARE DISTRIBUTION VIA INTELLIMIRROR 4

- SELECTING TARGETS 5
- ADVERTISING AND SCHEDULING DISTRIBUTION 6
- DISTRIBUTING PACKAGES 8
- PACKAGING APPLICATIONS 9
- INSTALLING ENGINE: THE WINDOWS INSTALLER SERVICE 10
- REPORTING DISTRIBUTION STATUS 12
- MAINTAINING APPLICATION: POST SOFTWARE DISTRIBUTION 12

SOFTWARE DISTRIBUTION VIA SMS 2.0 12

- SELECTING TARGETS 13
- ADVERTISING AND SCHEDULING DISTRIBUTION 13
- DISTRIBUTING PACKAGES 14
- PACKAGING APPLICATIONS 16
- INSTALLING ENGINE: SMS CLIENT AGENTS 16
- REPORTING DISTRIBUTION STATUS 17
- MAINTAINING APPLICATION: POST SOFTWARE DISTRIBUTION 17

HOW TO CHOOSE BETWEEN INTELLIMIRROR AND SMS FOR SOFTWARE DISTRIBUTION IN A WINDOWS 2000 ENVIRONMENT 18

- SCALABILITY AND FLEXIBILITY 19
- RELIABILITY AND PERFORMANCE 20
- INTEROPERABILITY 20

EASE OF IMPLEMENTATION 21
MANAGEABILITY 21

SEVERAL SCENARIOS FOR SOFTWARE DISTRIBUTION IN WINDOWS 2000 ENVIRONMENTS 21

SCENARIO 1: A SMALL-SIZED ORGANIZATION 22
SCENARIO 2: A MEDIUM-SIZED ORGANIZATION 23
SCENARIO 3: ANOTHER MEDIUM-SIZED ORGANIZATION 25
SCENARIO 4:A LARGE-SIZED ORGANIZATION 27

SUMMARY 29

Introduction

IT managers have always been challenged with rolling out new applications or application updates to a large number of users. It is an expensive and time-consuming task regardless of the size of the organization. The traditional software deployment performed by support engineers on each desktop often results in non-standard configurations and also disrupts the users' regular activity. Organizations have long desired technologies that could automate large-scale software distribution without disrupting user's activity and at the same time, automate application maintenance on the users' desktops after installation.

Microsoft has provided customers with the System Management Server (SMS) to automate the software distribution in the Windows NT environment. Besides software distribution, SMS has also been employed successfully by many organizations for software/hardware inventory, remote diagnostic and control of desktops, network tracing and monitoring, and software metering.

With Windows 2000, IntelliMirror has been introduced to further ease the task of software distribution and maintenance for IT managers. IntelliMirror consists of a set of powerful features built into the Microsoft Windows 2000 operating system that are designed to increase availability and reduce the overall cost of supporting Windows users. Software installation and maintenance is the core feature of IntelliMirror. This native software distribution capability built into the Windows 2000 operating system has caused confusion among organizations that are searching for Microsoft software distribution solutions when they plan to upgrade their Windows NT systems to Windows 2000 systems.

The following pages provide an in-depth analysis to better understand how IntelliMirror and SMS achieve software distribution in Windows 2000 environments:

- Selecting targets
- Advertising and scheduling distribution
- Distributing packages
- Packaging applications
- Installing engine
- Reporting distribution status
- Maintaining applications—post software distribution

Based on this analysis, this paper:

- Suggests some guidelines on choosing between IntelliMirror and SMS for software distribution
- Discusses the scalability, reliability, manageability, ease of implementation, and interoperability of both systems.

Finally, this paper presents four scenarios are for your reference on how to choose or integrate IntelliMirror or SMS to achieve software distribution solutions in a specific Windows 2000 environment. We also discuss scenarios of a small-sized company, two middle-sized companies, and a large-sized company.

What Is IntelliMirror?

The IntelliMirror management technologies are a set of powerful features designed for desktop *Change and Configuration Management* and built into the Microsoft Windows 2000 operating system. IntelliMirror uses features in both Windows 2000 Server and Windows 2000 Professional to allow users' data, software, and settings to follow them.

The features of IntelliMirror increase the availability of a user's data, personal computer settings, and computing environment by intelligent management of user's data, settings, and software. Based on policy definitions, IntelliMirror is able to deploy, recover, restore, or replace user's data, software, and personal settings in a Windows 2000 environment.

Three features are at the core of IntelliMirror:

- ***User data management:*** User's data will follow them; they can access their data whether they are online or offline and from any machine.
- ***Software installation and maintenance*:** The software distribution feature of IntelliMirror allows applications to follow users. This capability is achieved by automated deployment, configuration, repair, and removal of applications, service packs, and software upgrades without on-site technician support.
- ***User settings management*:** The customization of desktops and applications will follow users. So, a user's personal preference and setting for his or her desktop and applications are available wherever the user logs on.

The technologies behind IntelliMirror include, among others, Active Directory, Group Policy, Windows Installer, Offline Folders, Synchronization Manger, and Folder Replication. In the following sections, we will focus on the software distribution and maintenance feature of IntelliMirror. We will not cover the user data management and user settings management features of IntelliMirror in this paper. Keep in mind that these three features of IntelliMirror will only work on Windows 2000-based systems. They are not applicable to down-level windows operating systems, such as Windows 95/98, and Windows NT 3.51/4.0.

Software Distribution via IntelliMirror

IntelliMirror allows users' applications to follow them so they can access their applications on any machine in a Windows 2000 environment. IntelliMirror is also capable of distributing an application to a group of computers, so that any user who logs on to these computers can access the application. The capability of the automated software distribution within IntelliMirror provides these two features. In addition, IntelliMirror can make automatic repairs to any damaged applications that have been distributed to user's computers by IntelliMirror, whether they are caused by missing or damaged files, or even if accidental removal of the whole application from the user's computers occurs. The technologies behind these features, including Active Directory, Group Policy, Windows Installer, and the new MSI packaging mechanism for Windows Installer, are built into the Windows 2000 operating system.

Following is a brief description of how these Windows 2000 technologies work together to accomplish a software distribution via IntelliMirror. First, Active Directory is the database for software distribution targets. No matter whether the targets are a group of users or a group of computers, they must be the objects stored in Active Directory. Second, Group Policy actually defines the targets, and assigns or publishes an application to the targeted groups of users or

computers. Finally, Windows Installer service on user's Windows 2000 systems will install automatically, and further maintain, the applications assigned to users or computers. Before a software distribution cycle, the MSI packaging mechanism must be used to author application packages that are required by Windows Install service.

It is not the intention of this paper to give an in-depth exploration of how each of these new technologies functions; you can find White Papers or other documentation regarding each of these new technologies on the Windows 2000 operating system. Rather, we will focus on:

- How these technologies work with each other to accomplish the automated software distribution feature in a Windows 2000 environment
- Discuss a comparison of this feature with Microsoft's System Management Server Version 2.0 in a Windows 2000 environment.

This paper will help you decide how to choose between IntelliMirror and SMS 2.0 for automated software distribution, and how to integrate one with another in your Windows 2000 environments. To better analyze the software distribution feature of IntelliMirror and SMS, and to make a more intuitive comparison between IntelliMirror and SMS, a full software distribution cycle via both IntelliMirror and SMS is divided into the following stages:

- Selecting targets
- Advertising and scheduling distribution
- Distributing packages
- Packaging applications
- Installing engine – the Windows Installer Service
- Reporting distribution status
- Maintaining application—post software distribution

Note: Packaging applications should happen at the first stage of a cycle. The reason that we put it just before the installing stage is to better display how packaging application and installing packages work closely with each other.

In the following sections, we will explain in detail how software distribution functions at each of the above stages when IntelliMirror is employed to deliver applications to a group of users or computers in a pure Windows 2000 environment. We assume that you already have a good understanding of Active Directory, Group Policy, DFS, and Windows Installer.

Selecting Targets

The targets for software distribution via IntelliMirror can be groups of users or computer objects in the Active Directory, but not a specific user or computer. IntelliMirror uses the Active Directory and Group Policy together to define the targets for software distribution in a Windows 2000 environment. The targets that group policy can define include sites, domains, and organizational units (OU), because they are the containers in Active Directory to which Group Policy can be applied. If an organization has very clear sets of applications for a site, a domain, or an organizational unit, it will be very easy to select the targets for software distribution through IntelliMirror. For example, if application A will be used and only used by users in OU 1, and

application B will be used and only used by users in OU2, then one Group Policy object for each application can be created and applied to OU1 and OU 2 with its application respectively.

The smallest container that Group Policy can be applied to is an organizational unit in the Active Directory. If an application must be deployed only to a specific group of users or computers within a single organization unit, defining targets for software distribution through IntelliMirror will become complicated. However, it is possible to select a specific group as the distributing targets by using group policy filtering—special security properties on a Group Policy Object (GPO). To do so, that specific group will be assigned the read and run rights on the GPO defined for the software distribution, while the read and run rights on this GPO for all other groups in the same Organization Unit must be disabled manually or by disabling the read and run rights for the authenticated users. With this practice, it can become extremely complex to maintain and troubleshoot the GPOs, especially in a very large Windows 2000 environment where hundreds of groups exist in one Organization Unit.

To use IntelliMirror as the primary application distribution mechanism in an organization, the OU structure of the Active Directory should be designed to match the application distributions in this organization. For instance, two applications, Microsoft Visual Studio 6.0 and Microsoft Office 2000, must be distributed to two groups of users in this organization—developers and salespersons. Visual Studio 6.0 only needs to be deployed to developers, and Office 2000 to sales persons. To simplify the Group Policy structure, two organization units, one for developers and one for salespersons, should be created. So, one Group Policy object for each OU can be created to assign these two applications to the prospective OUs.

Advertising and Scheduling Distribution

Windows 2000 Group Policy is the foundation of IntelliMirror and the key technology behind the built-in software distribution feature of Windows 2000. It runs on all authentication servers (Domain Controllers). In addition to its role in selecting targets for software distribution, Group Policy is also responsible for advertising applications to targeted groups of users or computers.

Advertising Applications

Two software distribution methods via IntelliMirror, assigning and publishing applications, are defined by configuring Group Policy on Windows 2000 Domain Controllers. You can either assign or publish applications to a group of users, but you can only assign applications to a group of computers.

When you assign an application to a group of users, Group Policy ensures that the application will always be available to the users. The Group Policy object that defines the software distribution keeps a short description about the software, including the shortcut, name, locations of the software, etc. During the logon process, the *winlogon.exe* service running on user's computers will communicate with the Group Policy on the domain controllers to obtain this information if they have been assigned to this application.

After these users log on to their Windows 2000 systems, the application shortcut will appear in the *Start* menu. In addition, the registry will be updated with information about the application assigned to them, including the location of the application package and the location of the source files. With this advertisement information downloaded to users' computers, the application will be

installed automatically the first time users activate the application from the *Start* menu or open a file associated with this application.

Although users can remove assigned applications using the Add/Remove Programs feature in the Control Panel, they only remove it for the duration of a log-on session. The next time they log on to their computers, the application icon will reappear and will be reinstalled if users start to use it. The Windows Installer service built in Windows 2000 systems accomplishes this auto-repair feature. We will address the details of Windows Installer service in the following sections.

An application can also be assigned to a group of computers. During their first boot-up process after the assignment, the application assigned to these computers will be installed before users log on. Afterwards, any users who log in to these computers can access that application.

With the Windows 2000 Group Policy, applications can also be published to groups of users

Note: This is only applicable to user groups, not for computer groups.

Users can install published applications from the Add/Remove Programs feature in the Control Panel. When an application is published to a group of users, no shortcuts of the application appear on users' desktops, and no registry entries changes are made. The advertisement information for applications is stored in the Group Policy object, and the *mshta.exe* process is responsible for communicating with the Group Policy object to get the advertisement. To install a published application, users can use the Add/Remove Programs feature in the Control Panel, which will include a list of all published applications that are available to the users; or, users can open a document file associated with a published application. For example, double clicking an *.xls* file will start the installation of Microsoft Excel on users' computers, if Excel has been published to these users.

Organizations will find this feature very useful and efficient for distributing applications to a wide range of users when the user population for each application is not defined clearly. By publishing an application through IntelliMirror, users can decide whether to install this application and if so, when. This is called on-demand installation and we will discuss this capability in more detail later in this paper. As an illustrative example, an organization has purchased 50 licenses of Visio software for use by the entire staff. Ten departments claim that some of their users will need the software but the IT department cannot determine which users will use the software in these ten departments. In this situation, publishing Visio to all these ten departments can be accomplished easily via IntelliMirror. Of course, some schemes, such as SMS software metering, should be in place to control the licensing issue.

Scheduling Distribution

The software advertisement via IntelliMirror is available immediately to users or computers after their next logon or reset, provided that all the domain controllers in a domain are synchronized with the new Group Policy object. In a large environment with many Windows 2000 Domain Controllers, especially when they cross multiple sites over a Wide Area Network (WAN), it may take some time to replicate the Group Policy objects to all the domain controllers. Users authenticated to an unsynchronized domain controller will not notice the advertisement until the domain controller is synchronized.

Group Policy information is stored in two locations: a Group Policy container in Active Directory and a Group Policy template in the domain controller's *Sysvol*. You need to understand how Active

Directory and *Sysvol* replicate in your environment before you can anticipate when users will see the advertisement after an application is assigned or published to them through IntelliMirror.

Unlike SMS, there is no built-in scheduling scheme to control the software distribution process via IntelliMirror. Once an application is assigned or published to a group of users, these users will be able to download the advertised software immediately after they next log on (assuming all domain controllers are synchronized). If the application is assigned to the computers, the application will be installed immediately after these computers are restarted. Therefore, it is the users' call to determine if and when to deploy the application by clicking the icon on their desktop shortcut, installing it from the Add/Remove Programs feature in the Control Panel, or by rebooting their computers. The system administrator has no control over the timing of the software distribution.

System administrators must be aware of the impact that IntelliMirror traffic will have on their network when it is used to deploy an application to a large number of users or computers. For example, an organization decided to deploy Microsoft Office 2000 to all of its users via IntelliMirror. During the night, the Office 2000 package was loaded on to one file server, and a Group Policy object was created to assign Office 2000 to all the users in this organization. The next morning (from 8 to 10 A.M.) when hundreds of users logged on to their computers and tried to open Word or any Office 2000 applications, *all* of the Office 2000 applications were downloaded and installed immediately on all these computers. Because the download and installation occurred within a 2-hour time frame, the corporate network was overloaded and the file server was hammered by the download processing. All other regular activities were also disrupted during that period of time. This is an important lesson for an organization to learn if it is going to embrace IntelliMirror for its software distribution solution. Later in this paper, we will discuss how to avoid this type of reckless deployment by employing a manual schedule for software distribution via IntelliMirror.

It is important to note that assigning or publishing an application through IntelliMirror is persistent. That is, this application will always show up on the user's desktop, or in the Add/Remove Programs feature in the Control Panel until the assigning or publishing is removed specifically from the Group Policy.

Distributing Packages

A major concern for an organization in designing a good automated software distribution system is how this distribution system will impact the organization's LAN and WAN. The key is to limit the impact by setting up more distribution points and placing them close to the location of targeted users. Unlike SMS, IntelliMirror does not provide any built-in mechanism to place packages to distribution points in close proximity to the users. It must be designed manually and implemented before any applications can be assigned or published to users or computers.

To optimize the performance of software distribution through IntelliMirror and limit its impact on your network, establish a method to place application packages close to the targeted users, and to enable load balancing and fault tolerance for large deployment. The Microsoft Distributed File System (DFS), a built-in feature in Windows 2000, is a good solution for setting up software distribution points to enable load balancing and fault tolerance in a Windows 2000 environment.

With Microsoft DFS, multiple copies of shares on separate disk drives or servers can be mounted under the same logical DFS volume, thus permitting load balancing between servers. The application package for one IntelliMirror application assignment can then be placed on several file share servers that are mounted under a single logical DFS volume. Later, as users try to install the applications assigned or published to them, the downloading request will be referred transparently

and randomly to one of the network shares comprising the DFS volume. To distribute users randomly and evenly between several servers with the same application package will enable load balancing of the software distribution process via IntelliMirror, and enhance the distribution performance.

In the example given in the previous section, if the Office 2000 package was placed on several servers, and mounted under one DFS volume name across its network, it would mitigate the impact of the Office 2000 deployment via IntelliMirror on their network and server.

It should be noted that the load balancing through Microsoft DFS is limited. It may not work well on very large and complex network environments where software distribution must cross multiple sites linked via slow and complex WANs.

Packaging Applications

Packaging applications is always one of the most critical keys of large-scale deployment, no matter which method you use to deploy the applications. A rollout a faulty application package to hundreds or thousands of users can be extremely painful because it is difficult and time consuming to remedy a massive deployment and it disrupts regular business operation. For this reason, many skilled engineers dedicate time to develop and test packages (for applications and their updates) in organizations where regular deployment of applications to large user groups occurs via a software distribution system. We cannot overemphasize how critical a good package is to a successful deployment.

With the introduction of Windows Installer, Windows 2000 provides many new features for developing more efficient and consistent application packages for large corporate deployment.

Application packages developed for the Windows Installer contain three hierarchy components: products, features, and components. Unlike the script-driven package written for SMS, the database style package files for the Windows Installer applications can be optimized for installation performance. These files describe, among other things, the relationships between features, components, and resources for a given product. A Windows Installer package file has the extension of MSI and typically appears at the root of the product's CD or network share, along side the product's files. However, it can also contain the product's files in internal compressed cabinet files (.cab).

Microsoft, along with other third parties, provides rich sets of packaging tools for Windows Installer applications. Most of these vendors also have corresponding tools for packaging SMS applications. These MSI packaging tools include InstallerShield for Windows Installer, Wise for Windows Installer, Visual Studio Installer from Microsoft Visual Studio, and Veritas WinINSTALL LE.

Some new features of Windows Installer packages, compared with packages designed for other existing installers, include:

- Database formatting of package files, rather than procedural scripts
- Hierarchy structure that consists of products, features and components, rather than a single and flat product
- Managing applications at the component level, rather than files or registry entries
- Universal identification for each component enables component sharing between multiple applications

- Refcount on components level enables much cleaner uninstalling.

To leverage IntelliMirror fully as a software distribution method in a Windows 2000 environment, much investment is required on how to:

- Develop good Windows Installer packages
- Customize commercial Windows Installer packages, such as Microsoft Office 2000, for specific usage.

Installing Engine: The Windows Installer Service

The Windows Installer, a built-in component in the Windows 2000 operating system, is the key component of the software distribution feature within IntelliMirror. The Windows Installer is the installing engine that achieves the final goal of a software distribution—installing the software on user's computers without an on-site support technician. Beyond this general goal, the Windows Installer has also added many features for software maintenance after installation on users' computers.

The Windows Installer, a new built-in installation service in Windows 2000 systems, consists of:

- An operating system-resident installation service
- A standard format for component management
- A management API for applications and tools

The operating system-resident installation service, appearing as "Windows Installer" in the services and application console, runs under the security context of a local system account that has the administrative privilege over users' computers. This is why the Windows Installer can distribute software on lock-down computers. We have already discussed the standard format for component management in the previous section of this paper. The management API for applications can be used to manage the applications after their installation, or to trigger an on-demand installation.

These three components integrate with each other to accomplish many powerful features of software installation and management in a Windows 2000 environment. System administrators, who are responsible for large-scope application deployment, have long desired these features.

On-Demand Install, Both Feature and Product Level

The Windows Installer can install any features on the fly within an application when users need them. Users do not have to exit the application and run the setup program again. In other words, all features of an application are available for use even if they are not yet installed on users' computers.

As mentioned in the previous section, an application can be published to a group of users without actually installing it on their computers. When these users need this application, the Windows Installer will install the application automatically and open it for the users, either through the Add/Remove Programs feature or through file extension association.

Transacted Installation (Rollback)

The Windows Installer providers the Rollback feature when an installation process encounters a fatal error during setup. By maintaining an undo operation for every operation, including removal or any other configuration change that it performs during an installation, the Windows Installer can

return the computer precisely to its previous state. This includes restoration of deleted or overwritten files, registry keys, and other resources. This feature can help to keep the damage at a minimum level for a massive software distribution in unexpected situations, such as a network glitch, or a power failure.

Upgrades and Patching

This feature is accomplished by developing applications that compile with the MSI packaging format for the Windows Installer. The Windows Installer can determine whether older versions of files should be removed, or whether an application file should not be installed if a newer version is already installed. This is possible because the Windows Installer keeps a database of information about all the products installed on the computer, including all their features and components. It also provides built-in patching technology to apply a patching package to an existing Windows Installer-installed application.

Customization Transforms

In the past, if system administrators wanted to customize the behavior of an installation, such as unattended installation (silent installation), they had to modify the setup script of each application to achieve the desired results. If similar changes were required to many different setup scripts, system administrators needed to repeat those efforts for each script. With Windows Installer, generic modifications can be built into a single transform and applied to many different packages, as long as those modifications are *legal* for the package files in question.

Operation in Lockdown Environment

To reduce support costs, many organizations have *locked down* their desktops by controlling people's ability to write to the system directory and registry. Since the Windows Installer operates as a system service on Windows 2000 systems, the installation of distributed software by the Windows Installer has the ability to run in one of two contexts:

- As the Local System account that has greater privileges than the user
- As the user that is the default behavior

Due to the security issues, to install an application under the Local System account via the Windows Installer service on a computer, a system administrator's approval is required for that application, as well as for that computer. This can be done by enabling the Group Policy setting of "Always install with elevated privileges" under both computer and user configuration.

Software Auto-Repair

The Windows Installer enables automated repair of an application in much the same way that it enables on-demand install. When a user opens an application, the Window Installer will perform a check to determine whether the components or features for that application are properly installed. If some files required by that a feature are missing or damaged, the Windows Installer will perform an on-demand installation to repair the application on the fly.

In conclusion, the Windows Installer is responsible for the installation of distributed software. Furthermore, it introduces many new technologies to centralize the management of application installation and maintenance in a Windows 2000 environment. The Windows Installer plays a core role in the software distribution processing via IntelliMirror. Understanding the technologies behind

the Windows Installer will surely help an organization to better leverage IntelliMirror as its software distribution solution.

Reporting Distribution Status

Windows 2000 IntelliMirror does not provide a centralized status reporting mechanism for software distribution, although the Windows Installer provides very rich installation information in a log file on individual computers. So, there is no feedback on how well a large-scope software deployment was undertaken by IntelliMirror. If IntelliMirror is employed for the software distribution in an organization, and it is critical for this organization to obtain the status report, the organization should develop a solution to gather status information from the log file in each targeted computer.

Maintaining Application: Post Software Distribution

With any conventional software distribution system, such as Microsoft SMS, once it distributes an application is distributed to users' computers and installs it successfully, the mission of software distribution is completed. The responsibility for maintenance and removal of the application belongs to the users and their support technicians

With Windows 2000 IntelliMirror, a full cycle of software distribution has expanded beyond the initial installation; it also includes the maintenance of all the applications installed on users' computers. IntelliMirror is capable of repairing, removing, upgrading, and patching user's applications automatically and transparently. The features of auto-repair and on-demand installation of the Windows Installer accomplish this goal.

Software Distribution via SMS 2.0

Microsoft System Management Server 2.0 is one of the most complex Windows management systems. With its flexible site hierarchy structure, SMS can be scaled to very large environment. It provides the following Windows management portfolio:

- Software and hardware inventory
- Software distribution
- Software metering
- Remote control and diagnostics of desktops
- Network monitoring and tracing
- Server health monitoring and status report

In the following discussions, we will only cover the software distribution feature of SMS 2.0 in a Windows 2000 environment. We will analyze the functionalities of the SMS 2.0 software distribution feature in each stage of a full distribution cycle as IntelliMirror.

Many of the software distribution features of SMS discussed here are only available in SMS 2.0. If there is an existing SMS 1.2 implemented on your Windows NT systems, you must upgrade to SMS 2.0 before you can upgrade the Windows NT systems to Windows 2000 systems. Windows 2000 systems require SMS 2.0 with service pack 2.

Selecting Targets

Systems Management Server 2.0 provides much richer options for selecting distribution targets than IntelliMirror in a Windows 2000 environment. With the detailed inventory of all the users and computers in an organization stored in its inventory database, SMS 2.0 can distribute software to any combination of the following:

- Users
- User groups
- TCP/IP network segments
- Computers

The SMS inventory information is stored in the SQL database dedicated to SMS. The SQL database contains information about users, user groups, network segments, NT domains, SMS sites, and most of all, detailed hardware and software information about user's computers. SMS software distribution uses sophisticated SQL queries to target specific users or computers based on their detailed information stored in the SQL database.

For instance, a query can select all the computers in an organization that satisfy Windows 2000 Professional's minimum hardware requirements, such as more than 64 MB physical RAM with a Pentium 166 or higher CPU, before a Windows 2000 Professional upgrade package is distributed to those computers. One can also use a query to select all the computers with certain IP segment numbers as distributing targets, so that a large package will not be distributed across slow and expensive WANs. Software can also be distributed to a group of users based on their group membership. The membership targeting method is dynamic in SMS 2.0. If a user's group membership has changed during a software distribution process, SMS will determine automatically whether the user is still qualified as a target. The software will be distributed to the user, or the software will be removed from the user by SMS depending on whether the user is joined to the group or removed from the group.

All other collections gathered by SQL queries for software distribution in SMS are also dynamic (like the group membership) and they can be combined together to define more granular targets. The flexibility in SMS 2.0 allows it to define ideal groups of targets, especially in a large organization with a very complex network structure. By carefully selecting the targets, system administrators can optimize the performance of a large deployment and control its impact on their corporate network, both LAN and WAN.

However, SMS requires a great deal of effort to plan and implement SMS client agents on all computers in an organization. These client agents must collect inventory data and provide it to the central SQL database before SMS can use this database to select targets for software distribution. With IntelliMirror, an organization will inherit all the information of potential targets immediately—objects in its Active Directory database for Windows 2000. No extra planning and implementation is required before software distribution.

Advertising and Scheduling Distribution

Advertising Distribution

Software distribution is advertised to users and computers through SMS site servers. Each user or computer has its own instruction file that contains all the information about the software advertised to them including the scheduled time, and the appropriate distribution point from which to

download the software, etc. The instruction file is the vehicle to advertise a software distribution to targeted users. The SMS client agents on user's computer will check their instruction files during each polling interval (default is 60 minutes).

Advertised only software distribution and assigned software distribution are the two types of software distribution methods in SMS. When advertised only software is available to users or computers, it is the users' choice as to if and when they install it after it is available. Assigned software is considered a mandatory installation after a certain time if users haven't done so.

Scheduling Distribution

As we discussed earlier, the software distribution feature in IntelliMirror does not have any capability to schedule a software deployment. The software assigned or published to users or computers is available to them immediately. They choose when to install the software and system administrators have no control over when and where software distribution traffic hits the network.

Unlike IntelliMirror, SMS 2.0 provides key tools to allow system administrators to schedule distributions at desired times. Usually, distributions occur outside working hours. The SMS administrator interface, a MMC snap-in, is the central place where a software advertisement is created. First, all the software advertisements are placed in the SQL database. Then, the SQL database works with SMS services on SMS site servers to dispatch the advertisement instructions through the SMS site hierarchy to all the SMS client access points (CAP). The potential targeted users or computers poll for any software advertisements available to them at the CAP.

With either an advertised only or an assigned software distribution in SMS 2.0, an available time is set on the software package, so the targeted users or computers will only be able to see and install that software after that available time. Users can then choose if and when to install it.

Furthermore, with an assigned software to a computer, a mandatory time can be set on the package so that the targeted computer has no choice but to install the software after that mandatory time. If no user logs on after that time, the SMS client agent on the computer will be intelligent enough to install the software on that computer automatically. With this option and unattended installation capability of the SMS client agent, system administrators can make scheduled and controlled software distributions. Large-scale software deployment will happen only during the time they specify to limit the network impact.

Distributing Packages

SMS 2.0 is designed with many mechanisms geared toward supporting software distribution in very large organizations with a complex LAN and WAN structure. A well-implemented SMS infrastructure can be very efficient in dispatching application packages close to where targeted users and computers are located, thanks to following mechanisms of SMS 2.0:

- Flexible SMS site hierarchy that can be designed to match a complex network topology
- Automated load balancing between multiple distribution points
- Various Senders that are network efficient and WAN aware
- Courier Sender
- Fan-out distribution method
- Scheduled distributing

- Bandwidth management

The very flexible SMS hierarchy enables an organization to design a SMS hierarchy with multiple primary/secondary sites to match its business units and locations, but most of all, its network topology. SMS sites can be tailored to match the IP network boundary in an organization, similar to the Windows 2000 site structure. Each SMS site can contain a certain number of IP network segments, and one or more SMS distribution points can be placed in each site so that they are very close to targeted users and computers. With less hops away from the download points, the software downloading process from SMS distribution points to targeted users can be extremely network efficient.

Any Windows 2000 file server can be configured easily as a distribution point for SMS. Multiple distribution points can be created for distributing software to a group of users in a SMS site. The SMS client agent, the installing engine, can be configured to download software from any of these distribution points randomly. This random selecting from multiple distribution points by the SMS client agent achieves an automated load balance between multiple points for software distribution via SMS 2.0. With this automated load balance between multiple distribution points, a very large application can be deployed via SMS 2.0 to numerous computers without overloading the distribution servers.

SMS 2.0 also provides a rich set of Senders to communicate between sites and dispatch software packages through the SMS hierarchy. A Sender is a SMS thread that transmits instructions and data from one site to another using existing network connectivity. To dispatch software packages efficiently through a complex network topology, especially when slow WANs are involved, SMS offers Senders that are specifically tailored for various WAN links. These links include the Asynchronous RAS Sender, ISDN RAS sender, X25 RAS Sender, and SNA RAS Sender. To be more efficient, software packages are compressed when they are delivered through the SMS hierarchy to reach distribution points from a central place.

In addition to the network specific Senders, SMS 2.0 also provides a new Sender—the Courier Sender for organizations with a very slow and expensive WAN. The Courier Sender allows software to be sent from site to site by CD-ROM, or other media, rather than across the network. This is useful in situations when the network bandwidth is too slow or too expensive to use for the delivery of a large package.

Fan-out distribution, another useful distributing feature in SMS 2.0, is designed for a very large organization with complex WAN links. With this Fan-out mechanism, SMS site servers can act as hubs, fanning the software packages out intelligently so that a distribution crosses a WAN link once. The Fan-out distribution will be extremely efficient for an organization with a very large network with multiple large sites that hub for many smaller satellite sites.

Furthermore, SMS 2.0 offers bandwidth management facilities so system administrators can indicate what percentage of bandwidth to use for a specific management task at a specific time. This makes it possible to schedule distribution based on network bandwidth, therefore ensuring package delivery does not affect regular network tasks.

Because of these sophisticated distributing mechanisms, SMS 2.0 can be scaled to very large organizations with very complex network and very intensive software distribution requirement.

Packaging Applications

As we have emphasized earlier, packaging applications is one of the most critical steps to achieving a successful deployment. SMS 2.0 provides its own package tool, SMS Installer, for developing application packages for commercial software, in-house software, and data files.

The SMS Installer provides repackaging facilities in addition to core wrapper technologies for software distribution. This reduces the preparation time for software distribution. The SMS Installer uses a snapshot technology that allows you to:

- Create a reference machine
- Take an image of its state before and after a software installation
- Calculate the difference between those two states
- Create a software package that can be installed by the SMS client agents on user's computers.

The SMS Installer also provides facilities for developing sophisticated package scripts that can contain flexible functions, such as allowing conditional checks during the installation process. These scripting capabilities are very useful for organizations that want to customize the behavior of software installation of any commercial or in-house software.

Unlike the block and database style packages of Windows Installer, both methods of the SMS Installer will create a script-driven package file that contains all the instructions and compressed source files. In addition, the SMS installer includes (in a software package) the capabilities of removing itself and reporting its installation status, if it is installed by the SMS installing engine—SMS client agents.

Installing Engine: SMS Client Agents

The installing engine for software distribution in SMS is one of the SMS client agents. The SMS client agents consist of software installed on users' computers by SMS servers running as services on user's Windows 2000 systems. SMS client agents include Remote Tools Client Agent, Software and Hardware Inventory Agent, Advertise Programs Monitor Agent, and Advertise Programs Agent. The Advertise Programs Agent is responsible for installing distributed software on users' computers. In the following discussion, when we use the term SMS client agent, we mean the Advertise Programs Agent located in the Control Panel.

Unlike software distribution via the IntelliMirror, where the software-installing engine—Windows Installer service—is built into Windows 2000 systems, the SMS client agent must be installed on Windows 2000 systems before distributing software to these computers.

Installing SMS client software on all users' computers is not a trivial process during the setup of a SMS infrastructure. It requires a lot of planning and implementation. Because of this, SMS provides two installation mechanisms combined with several discovery methods to help system administrators to rollout SMS client agents on users' computers. SMS client agents can be installed manually by running a client setup program from a computer, or by running a logon script when a user logs on to the domain. In addition, SMS client agents can also be pushed to computers by system administrators through the Remote Installation Services that are running on SMS site servers, if these computers have been discovered already.

Compared with the installing engine in the IntelliMirror—Windows Installer—the SMS client agent doesn't support on-demand install, transaction installation (Rollback), customization transform, and software auto-repair. The SMS client agent does support installation in lockdown

environment, and upgrades and patching if the applications are packaged correctly with the SMS Installer tools or tools from other vendors. The upgrade and patching through the SMS client agent is treated the same way as any regular software installation, unlike in the IntelliMirror, where upgrade and patching are integrated tightly with the product itself.

As we discussed earlier, some applications must be installed in a user context to set up the appropriate icons, whereas others need administrator rights to access system configurations. Often, a user without local administrative privilege will need to install software that requires higher privileges. The SMS client agent is intelligent enough to check if a currently logged on user has the appropriate privileges. If not, it will use its own administrator privileges (local system account) on that computer to install the software.

Reporting Distribution Status

SMS stores and presents information to report the progress of a software distribution from a central database. Two types of status report are available regarding software distribution in SMS. Package status provides details on the success or failure of software distributing to targeted computers, including Received, Failures, Program Started, Program Errors, or Program Success. Advertisement status adds the program status information and scheduling details of the package, including when a distribution is scheduled, whether it is successfully distributed to its distribution points, etc.

During the software distributing process, SMS site servers will collect the advertisement status of how packages are distributed to their distribution points; meanwhile the SMS client agent on each targeted computer will report the installation status to its SMS Client Access Point (CAP) servers. Then, the CAP servers will forward the package status to the central database—the SQL database of the SMS primary server. From the central database, system administrators can compile customized and complete status reports about the entire deployment. Most of all, if the installation in some computers failed, administrators can create a simple SQL query to select those failed computers as targets for a second-round deployment.

When the distributed application is critical to the business operation, this sophisticated status reporting method in SMS will help the organization prepare and improve response to any problem during a large software deployment.

Maintaining Application: Post Software Distribution

In addition to the ability of distributing software to computers, SMS 2.0 has added options to remove software when a user or a computer no longer satisfies the targeting collection rule. When software is advertised to a collection of computers or users, system administrators can specify that the software should only be available as long as the computer or user remains in that targeted collection. For example, the targeted collection could be the finance group and the software could be a finance application. When a user changes groups (moves from finance to marketing), he or she is removed automatically from the finance collection, and the SMS client agent will remove the finance software automatically from his or her PC.

Unlike the IntelliMirror, SMS does not provide auto-repair and on-demand installation. After the software has been installed in targeted computers, SMS is not responsible for any maintenance of that software.

How to Choose between IntelliMirror and SMS for Software Distribution in a Windows 2000 Environment

To better compare IntelliMirror and SMS 2.0 for software distribution, we created the following table to summarize their characteristics during each stage of the software distribution cycle based on the previous discussion.

Features	IntelliMirror	SMS
Selecting targets	**Method**: Apply Group Policy on objects stored in active directory. **Properties**: No need to collect information before targeting; targeting is dynamic, fewer flexible choices.	**Method**: Apply SQL Queries on rich information collected by SMS. **Properties**: Need to implement SMS clients and collect inventory before targeting; targeting is dynamic, more flexible choices.
Advertising/scheduling distribution	**Method:** Group Policy informs the Winlogon.exe service on targeted computers. Two types of advertisements: assign and publish. **Properties**: Applications are available immediately to users or computers; can't schedule distribution.	**Method**: SMS admin console sends the instructions to SMS servers; SMS clients then poll servers for instruction. Two types of advertisements: advertise only and assigned. **Properties**: Distribution can be scheduled flexibly, off time distribution.
Distributing packages	**Method**: Requires manual placement of the package on file server shares. **Properties**: Need to plan the distribution points before deployment. No built-in network awareness, no built-in load balance and fault tolerance (DFS can be used to work around this).	**Method**: Packages are dispatched automatically to SMS distribution points that are close to users. **Properties**: Automatic and WAN aware distributing, fan-out distribution, easy-to-distribute package close to users, built-in load balance and fault tolerance.
Packaging applications	**Method**: Windows Installer packages **Properties**: Block structure package files, support enriched software installation features: rollback, upgrades, auto-repair, unattended, and customization installation.	**Method**: SMS Installer packages **Properties**: Script-driven package files, support customization, unattended installation.
Installing engine	**Method**: Windows Installer Service **Properties**: Windows 2000 built-in service, supports on-demand installation, transacted installation, upgrades and patching, generic customization, software auto-repair, operation in lockdown environment, auto-remove software.	**Method**: SMS Client Agent **Properties**: SMS Client Agent needs to be deployed to all computers before software distribution, supports auto remove software, installation in lock down environment, unattended installation.
Reporting status	**Method**: Localized log file of Windows Installer on user's computers. **Properties**: No mechanism to collect the distribution status from each target.	**Method**: SMS servers collect status from SMS client agent, and store them in a central place (SQL database). **Properties**: Centralized status report for all the targets; failed distribution can be selected for another distribution.
Maintaining application	**Method**: Maintenance is done by Windows Installer service. **Properties**: Software auto-repair, easier upgrades and patching, cleaner uninstall software.	**Method**: None

Many system administrators, who have worked on or heard about the built-in software distribution feature of Windows 2000 systems via IntelliMirror, probably will ask whether they still need to

implement Microsoft SMS to achieve automated software distribution in Windows 2000 environments. If you are also confused, we suggest you answer the following questions before making any decision. Does my organization need:

- Hardware and software inventory of users' computers?
- Support down-level operating systems, such as, Windows 95/98, Windows NT for software distribution?
- Remote control and diagnostics of user's computers?
- Software metering?
- Network tracing or monitoring?
- Server health monitoring?

If the answer is yes to any one of these questions, especially those on the top of the list, you will still need to implement the SMS infrastructure or keep your existing SMS infrastructure. The Windows 2000 IntelliMirror does not replace any of these functionalities and will not in the near future. If software distribution is the only feature you are looking for, IntelliMirror can be a possible alternative of SMS for software distribution; much of the decision will depend on the intensity and scale of the software distribution and complexity of network infrastructure in your organization, as revealed in the following discussion.

The only overlap feature between the IntelliMirror of Windows 2000 and SMS 2.0 is automated software distribution. The properties of both systems in various stages of a software distribution cycle have been discussed in detail in previous sections, and summarized in the previous table. Keep in mind that IntelliMirror is a built-in component of the Windows 2000 operating system, and beside the software distribution feature, IntelliMirror also provides user data management and user setting management. These are not available in SMS 2.0. You should take advantage of these new features built into Windows 2000 systems, even if you choose SMS 2.0 as the method for software distribution in your Windows 2000 environment.

Given the above arguments, the question one should really ask is how to choose between the IntelliMirror and SMS for software distribution in a Windows 2000 environment? We will address this question from following perspectives:

- Scalability and flexibility
- Reliability and performance
- Interoperability
- Ease of implementation
- Manageability

Scalability and Flexibility

When you design an automatic software distribution system, you want to know whether it can match your organization's deployment requirement, and whether it is scalable when your organization grows, or acquires/merges with other companies.

Many key components built into SMS can be designed to address specific environments and requirements, as well as to be scaled in very large organizations. These include the following:

- The flexible SMS site hierarchy can match a very complex network structure, and support either a centralized or decentralized distribution system.
- Rich information pertaining to all users and their computers in the entire organization, combined with sophisticated SQL queries, provide a very flexible targeting mechanism for granular distribution.
- WAN-aware site communication can fan-out large packages automatically and efficiently from a central place to all the distribution points crossing various LANs or WANs.
- Scheduled distribution can restrain the distribution traffic during off-work hours, and limit the network impact of software deployment.

The software distribution feature of IntelliMirror does not provide these functionalities. If you are looking for a more advanced software distribution that is very scalable and flexible for a large and growing environment comprised of a complex LAN and WAN infrastructure, SMS is definitely a better candidate than IntelliMirror for software distribution.

Reliability and Performance

If the software distribution is critical to your business operation, and it requires very high reliability and performance from the distribution system, SMS probably will be a better choice than the built-in software distribution feature of IntelliMirror. SMS provides multiple distribution points to achieve automated load balance and fault tolerance between these distribution points for distributing packages to numerous users. It also provides scheduled software distribution to avoid the impact on regular network traffic.

SMS also provides a sophisticated status report system to monitor the whole distributing system, as well as the status of the final installation on users' computers. A lot of organizations rely on these status reports for proactive actions against unexpected problems during distribution to achieve high reliability.

In addition, SMS also provides handy network tracing and monitoring tools to monitor the network traffic during the software distribution process.

IntelliMirror does not provide any features to fulfill these reliability and performance requirements. If IntelliMirror is employed to distribute software in on organization, the system administrators will need to plan and manually implement a reliable and load balance file share systems as software distribution points that can satisfy the network and server load. Distributed File System, a built-in component of the Windows 2000 system, is a good candidate of software distribution points in the IntelliMirror For details to setup an efficient DFS system in a Windows 2000 environment, please read related documentation. There is no central status reporting system in IntelliMirror, nor scheduling for distribution.

Interoperability

SMS 2.0 will still support down-level operating systems other than Windows 2000 systems, including Windows 95/98, Windows NT 4.0, and other server systems such as Novell Netware as client access point (CAP) or distribution points. If interoperability is the biggest concern in an organization, then SMS 2.0 will be a better choice for software distribution, because IntelliMirror will only distribute software to Windows 2000 systems.

Ease of Implementation

As we indicated earlier, IntelliMirror is part of the Windows 2000 operating system. After an organization upgrades or migrates to Windows 2000, it will gain immediate access to the software distribution function of IntelliMirror. It does not need to:

- Setup and implement an extra distribution system
- Gather user or computer information for targeting
- Deploy installing engine on all computers

The Active Directory and Group Policy are the only requirements for leveraging the software distribution feature in the Windows 2000 environment.

In contrast, designing and implementing a SMS infrastructure requires a substantial investment before achieving a scalable, reliable, and efficient software distribution system. You need to collect information on all users and computers in your organization (for targeting) and to deploy SMS client agents (installing engine) to all computers before you can distribute software.

In a small organization with little budget to implement a SMS infrastructure, IntelliMirror is a perfect candidate to achieve software distribution without a lot of investment. In many cases, IntelliMirror's software distribution feature probably will be a major driver for those small organizations to embrace Windows 2000.

Manageability

The manageability of a software distribution system is used in this paper to measure how much work is required to manage the distributing infrastructure, and how the distributed software can be managed and maintained.

The infrastructure for IntelliMirror is the combination of the Windows 2000 Active Directory, Group Policy Objects, and Windows Installer service on each Windows 2000 system. There is no extra effort required to maintain a separate system other than a healthy Windows 2000 environment. In contrast, to maintain a healthy SMS infrastructure, an organization needs to maintain a SMS hierarchy for software distribution besides Windows 2000 systems. To maintain a SMS hierarchy includes maintaining at least one SQL server, many SMS site servers, their services, and communication between them (Senders), as well as the SMS Client Agents on each computer.

Several Scenarios for Software Distribution in Windows 2000 Environments

The software distribution scenarios vary from many different perspectives in a particular environment. Beside the size of your organization, these perspectives can include, among others:

- Frequency of software distribution
- Size of software packages
- Granularity of software distribution (targeting)
- Criticality of software distribution
- Complexity of network topology (especially WAN)

- Reliability of the server infrastructure
- Education level of the targeted users.

You may have a long list of your own to think of before designing a software distribution system.

It is almost impossible in this paper to provide valuable suggestions for designing a software distribution system if all these factors must be considered. Therefore, only some of those factors will be considered when we are discussing a software distribution system for a sample organization. The following scenarios are categorized first by the size and infrastructure of an organization, then by their software deployment requirements.

Scenario 1: A Small-Sized Organization

Company Information and Infrastructure

Company S is a small organization with 500 Windows 2000 Professional users and a dozen Windows 2000 servers. All the users are located in one building and well connected to a high-bandwidth, 10/100 MB Ethernet network. Their Windows 2000 active directory is composed of a single forest and a single domain.

Software Deployment Requirements

Company S has a small IT department that is responsible for implementing and maintaining the desktop standard, and deploying applications to all desktops. Most of the applications are commercial software with built-in Windows Installer packages, and are company-wide (for every user in the company). Large and small applications require deployment. Microsoft Office 2000 is an example of a large deployment, but such a large deployment does not happen often in Company S.

Strategies for Software Distribution

Neither SMS infrastructure, nor any other software distribution system was implemented before Company S upgraded to Windows 2000 systems. Because of the lack of both IT budget and expertise, there is no plan to implement SMS to manage their Windows systems. To save the cost on application support, both deployment and maintenance, the IT department is looking for a simple and affordable solution for software distribution in their Windows 2000 environment. As expected, IntelliMirror becomes the candidate to deploy applications automatically to users' desktops thanks to the built-in software distribution feature of IntelliMirror in Windows 2000 systems. No extra budget is needed for this distribution system. It is easy for Company S to implement and achieve an automated deployment system.

To better serve the reliability and good performance of their distributing system via IntelliMirror, Company S has undertaken the following preparation before deploying any software:

1) They categorized all the users in a single OU to several user groups, such as sales, finance, programmers, etc. They will use these groups as targets for software distribution; no group has more than 100 members.
2) They set up a DFS root on a Domain Controller and created a logical volume on the DFS root with two file server shares linked to its name. This logic volume on the DFS root will act as the distribution point for future deployment. The two file servers will provide load balance, as well as fault tolerance, during distribution.

3) All the applications have their Windows Installer packages; these packages are customized and tested.

To deploy a large company-wide application via IntelliMirror, such as Microsoft Office 2000, Company S needs some planning and consideration because of the lack of scheduling and controlling methodology in IntelliMirror. As we have mentioned earlier, the software assigned to users through IntelliMirror is available immediately to those users after their next log on. It removes control from system administrators over when the users will install the software on their computers. Within in a week, they have created five Group Policy objects that contain the same Office 2000 package to granulize and schedule this large deployment of Office 2000. Each group policy object will only apply on one or a few small user groups by using group filtering—by specifying security on that GPO. During this deployment week, each day they create and apply a Group Policy object with the Office 2000 package to about 100 users. The Office 2000 will appear in these users' *start* menu after they log on to their computers in the morning, and a user on one of the Office 2000 icons will install it automatically after a mouse click.

By using this approach, Company S can deploy Office 2000 automatically in a much more controlled process to all 500 users and ensure that their network will not be slammed by a large company-wide deployment.

On the other hand, it is easier for Company S to deploy a small company-wide application, such as Norton Anti-virus software. They can create a single Group Policy object to contain the Norton Anti-virus package, and apply it to the domain in their Active Directory without any group filtering. The new Norton anti-virus will be available immediately to all users in the company, and users will install it the first time they click the Norton icon shown in their *start* menu. The two file share servers acting as the distribution points are reliable enough to handle the downloading activity of 5 MB files by 500 users in a short period of time (one or two hours). The network bandwidth in Company S is also proven high enough to handle this kind of downloading traffic without affecting other normal traffic.

Also on the list are several other applications that are only used by dozens of users and need to be rolled out via IntelliMirror. Because it is very difficult to group these users together, the IT department decides to create a Group Policy object to publish these applications to all users in the company OU. Any user who wants to install such applications, will be able to do so from the Add/Remove Programs feature in the Control Panel because these applications appear as available locally on all users' computers. However, the IT department will need to find a method to track the license if it applies.

As a conclusion, Company S is a perfect example where IntelliMirror can be leveraged to achieve an automated software distribution system.

Scenario 2: A Medium-Sized Organization

Company Information and Infrastructure

Company M is an organization with about 5,000 Windows Professional users and about 200 Windows 2000 servers. Most of users are located in three main locations and each location has about 1700 users. These users are well connected with a high-bandwidth LAN within each location. Three main locations are connected with each other via three OC3 ATM WANs. Their Windows 2000 active directory composes of a single Forest and a single domain.

Requirements for Software Deployment

Company M has a centralized IT department responsible for implementing and maintaining the desktop standard, and deploying applications to all desktops. The IT department also manages the server and network infrastructure in all three locations. The IT department will need to deploy only a few applications and these applications are not critical to business operations. Most of the applications for deployment are commercial software with built-in Windows Installer packages; there is no internally developed application. Both large and small applications require deployment. Microsoft Office 2000 is an example of a large deployment, but such a large deployment does not happen often in Company M. Additionally, when an application that is specific to a group of users needs to be deployed, those users are usually within one location.

Strategies for Software Distribution

A desktop management system from another vendor was implemented before Company M upgraded their Windows NT systems to Windows 2000 systems. Because that management system had not been a success as an automated software distribution system, Company M decided to take advantage of the Windows 2000 IntelliMirror for distributing software and to keep the legacy system for other tasks.

The main reasons for choosing IntelliMirror for software distribution in Company M include the following:

4) IntelliMirror is a built-in feature of Windows 2000; there is no cost for software license and implementation

5) Software distribution is light in Company M. Only a few commercial applications, mostly small packages, require company-wide deployment and large packages, such as Microsoft Office 2000, are only deployed in a several-year cycle

6) Software distribution is not mission critical so no deadline for accomplishing a distribution exists; scheduling is not required;

7) Groups using one specific software package are mostly localized in one area so distribution traffic via IntelliMirror can be controlled easily within a location without crossing the company WAN.

As Company S has done before using IntelliMirror to deploy software, it also needs to implement a solid distributing infrastructure, and to optimize its active directory for distribution, but in a much larger scope and more complex way.

8) Because each location is quite an automated business unit, they create three OUs under their single active directory domain; each OU represents one physical location in Company M. They apply Group Policy objects for software deployment independently on separate OUs. Within each OU, they categorize users to dozens of user groups such as doctors, residents, nurses, etc. They will use these user groups as targets for software distribution. To control the distribution traffic no user group has more than 100 members.

9) Obviously, more sophisticated and robust DFS systems are required for handling massive distribution traffic in Company M compared with Company S. Company M implements at least six file-share servers as distribution points. At each location, they link two or more dedicated file servers to a DFS logic volume for distributing software packages in that location. The multiple file share servers will provide load balance and fault tolerance for

software distribution. The recommendation is to have at least two DFS root servers for fault tolerance.

10) All the commercial applications in Company M have Windows Installer packages; these packages have been customized and tested.

As you can imagine, to deploy a large company-wide application like Microsoft Office 2000 via IntelliMirror in Company M requires much more planning than it does in a small company. It will be much more time consuming too. First, on a manual basis, they will need to copy the package files for Office 2000 to six or more file-share servers that act as distribution points. To granulize and schedule the deployment manually, they create 10 Group Policy objects on each OU in a 2-week period, i.e. total of 30 Group Policy objects for one application need to be created. Each day during these two weeks, about 500 users will install Office 2000 on their desktops. Most of these installations will probably happen in a short period of time in the morning when users first log on their computers. Fortunately, the installation traffic does not cross the company WAN between three physical locations, because each Group Policy object is only applied on the OU for one location.

It is also easy for this middle-sized company to deploy small company-wide applications via IntelliMirror, such as Norton Anti-virus software. To simplify the deployment, they can create three Group Policy objects with the software package on each of the OU to make the new anti-virus software available immediately to all the users. However, if all 5,000 users begin to install this new anti-virus software during the first hour after it is available to them (assuming the package size is 5 MB), 25 GB (about 8 GB at each location) of software downloading traffic will be thrown on to Company M's network during that hour. As a precaution, they should create more granular Group Policy objects in a longer period of time to schedule the deployment manually.

For user-specific applications, where it is not possible to create an appropriate user group, IntelliMirror is still the best vehicle to publish these applications to all the users (as was the case for Company S). However, it will be much more difficult for Company M to track the license issues than it is for Company S. This is a scenario where the software metering feature of SMS can be very beneficial.

Scenario 3: Another Medium-Sized Organization

Company Information and Infrastructure

Company N is the same size as Company M, with about 5,000 Windows Professional users and about 200 Windows 2000 servers. Most of users are in three main locations; each location has around 1700 users. These users are well connected with high-bandwidth LANs within each location. Three main locations are connected with each other via three OC3 ATM WANs. Their Windows 2000 active directory composes of a single Forest and a single domain.

A centralized IT department is responsible for implementing and maintaining the desktop standard, and for deploying applications to all desktops. The IT department also manages the server and network infrastructure in all three locations.

Requirements of Software Deployment

Despite the same infrastructure, Company N is completely different from Company M as far as software distribution. Company N has the following requirements for software distribution that differ from those of Company M:

- The software distribution is very intensive; that means a lot of large application deployments and frequent deployments.
- Software deployments are critical to business operation in many cases, scheduling is required, and feedback about any failures is required.
- Groups of users who use same applications usually spread across three physical locations. OU is not based on location but on business functions.
- There are many internally developed applications that require deployment , and they are not Windows Installer compatible.

Strategies for Software Distribution

As shown in the Company M example, a lot of manual planning and scheduling are required to control a single large deployment via IntelliMirror. For instance, over 30 Group Policy objects were created for a single deployment in Company M, package files were copied manually to over six file-share servers, and a robust DFS system for software distribution was set up and maintained manually by system administrators. With the much more extensive deployment requirement in Company N than that in Company M, a huge amount of administration overhead will result if Company N uses the IntelliMirror to distribute software.

Unlike Company M where most software distributions are not business critical, Company N relies on its software distribution system to deliver business-critical applications to its users. For this reason, Company N requires the capability to anticipate when and whether the software will be delivered to every targeted user or computer. To achieve such high expectation, a central and automated status report system is inevitable for discovering possible failure during and after a deployment, so that a remedy can be developed and distributed out to recover problems. As we have discussed earlier, IntelliMirror does not provide any centralized status report system, so it does not satisfy Company N's requirement of a reliable software distribution system.

Also, unlike in Company M, a group of users who use one application in Company N are usually crossing three physical locations; their application distribution among users is not geographic oriented, but business oriented. Therefore, the application deployment to that group will cause a lot of downloading traffic crossing company WANs if IntelliMirror is employed to deploy the application. Even worse, the downloading traffic usually happens during work hours and in a short period because IntelliMirror cannot schedule off-hour software distribution. It will be much more difficult for Company N to restrict the distribution traffic via IntelliMirror within one location as it is done in Company M.

In addition, Company N will have to make a large investment to convert its internally developed applications to Windows Installer compatible packages, and to test those converted packages should they decide to deploy them via IntelliMirror.

For these reasons, Company N decides to implement a SMS infrastructure in its Windows 2000 environment to satisfy its specific software distribution requirements. As you can see from our earlier discussion, SMS can be designed easily to fulfill the specific requirements for Company N. Since it is not the intention of this paper to discuss SMS designing for a specific organization, we will simply illustrate how a SMS infrastructure can be used to satisfy the software distribution requirements in Company N.

To localize the software distribution traffic within each location in Company N, they can create a SMS site hierarchy to match Company N's geographic structure, as well as its network structure,

by building a SMS site at each location. One of these sites will be the central site. The system administrators in Company N will then be able to accomplish highly reliable and automated distribution from a central place, based on many software distribution functionalities built-in SMS as follows:

- Because the sophisticated targeting method enabled by the SQL query in SMS, Company N can now distribute an application easily to the users in one specific location. They can select a distribution point at their location specifically for them, despite the fact that users in that group are using applications across three locations.
- The enriched status information of a software distribution at a central place in SMS enables the system administrators in Company N to discover any possible distribution failures immediately. And, the administrators can even schedule a remedy or a second distribution to only those failed targets via an SQL query.
- The scheduling ability in SMS will help Company N to distribute their large applications during off-work hours to eliminate the impact of distribution traffic on their network during regular work hours.
- The automated package dispatching system in SMS will enable system administrators in Company N to send packages to all the distribution points from a central place.
- Furthermore, the automated load balancing and fault tolerance between these distribution points will enhance the distribution performance and reduce network impact of large-scale deployment via SMS.

Scenario 4:A Large-Sized Organization

Company Information and Infrastructure

Company L is an organization with about 60,000 Windows Professional users and about 1000 Windows 2000 servers. Most of the users are spread cross the United States, and about 2000 users are international. There are five large regional locations each with about 5,000 users, 40 middle-sized locations each with about 800 users, and 100 small-sized offices each with about 30 users. The international users are located in 10 countries with about 200 users each. The five large regional locations are connected via high-bandwidth ATM WAN backbone (multiple OC12 connections at each location), 40 middle-sized locations are linked directly to one or two large location via OC3 ATM links. All small-sized offices and international sites are connected to a middle-sized location or a large-sized location via T1, ISDN, or Frame Relay connections.

Requirements for Software Distribution

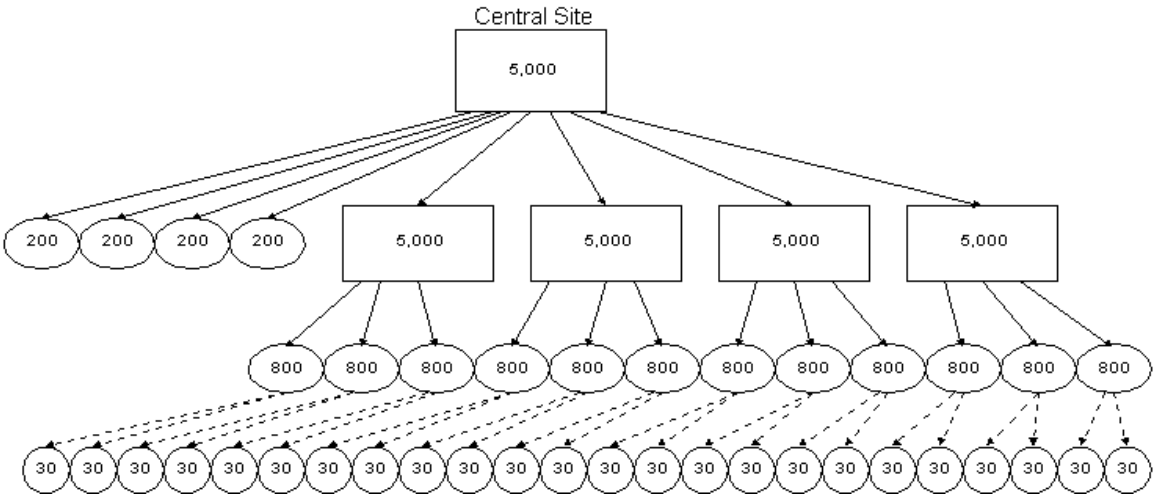
Company L has a centralized IT department that controls corporate desktop standards as well as the standard on how to deploy and install applications on a user's desktop. Four regional IT departments also help the central IT department. The central site is directly responsible for the desktop change control management for their largest location—headquarters, plus all international locations. The four other regional IT departments are responsible for assisting the central IT department to manage the desktops in their locations, as well as those middle-sized and small-sized locations connected to them respectively.

Since software distribution is centralized and decentralized the Central IT department is responsible for creating software packages for all company-wide applications. However, the

regional IT departments will deploy these packages to their regional locations and those directly connected to them. In addition, each regional IT department may need to develop and deploy its own software packages that are used only in their region. Software distribution is intensive, and includes a lot of large packages.

Strategies for Software Distribution

Based on our previous discussion, it is impossible to establish an enterprise software distribution system in Company L based on the IntelliMirror of Windows 2000 because of the large deployment scope and the network complexity in Company L. After some research and comparison between various software distribution solutions, the central IT department chooses SMS 2.0 to address the software distribution requirement in their complex network infrastructure, with the following design of SMS site hierarchy:



In this diagram of their SMS site hierarchy, the rectangle represents a primary SMS site, the ellipse represents a SMS secondary site, and the circle represents a location without any SMS site but a distribution point. The number in each object shows the number of users in that location. The solid line indicates a site connection between two locations, the dash line indicates only a distribution point in the end location. The diagram does not show all locations because of space limitations.

As we have mentioned earlier, this paper should not be used as a guideline for designing SMS architecture for a specific environment. Instead, we want to use this simple SMS design to illustrate how the SMS site hierarchy can be designed to fulfill a software distribution in a very large and complex environment. As shown in above diagram, the central IT department controls the central SMS site and will cover all the users in their location (headquarters). Directly under the central SMS site, the diagram shows four child secondary sites, each for an international location, and four child primary SMS sites that are delegated to four regional IT departments (each covers a regional location). Under these four child primary SMS sites, 40 child secondary (grandchild to the central site) sites exist, each for a middle-sized location. Within these child secondary sites, each small site has a distribution point with about 30 users directly connected to them.

As we discussed for Company N in the previous section, the system administrators in Company L will be able to accomplish a highly reliable and automated distribution system from a central place, based on these built-in SMS functionalities:

- The sophisticated targeting method enabled by SQL query in SMS
- The enriched and centralized status information of a software distribution
- The scheduling capability of large deployment
- Automated package dispatching system
- Flexible site boundary based on IP segment numbers.

In addition, we want to emphasize that SMS provides efficient ways of distributing a large package crossing Company L's complex network topologies based on various SMS Senders. These SMS Senders are designed to work over WAN environments; they support scheduling, batching, compressing techniques to dispatch packages efficiently over WAN connections. With the above SMS site hierarchy, Company L can distribute a company-wide software package from the central place (headquarters) to all the distribution points in the whole enterprise with just three hops or less. And more importantly, with the Fan-out mechanism of SMS 2.0, each package will only need to pass over a WAN link once, and each SMS site server will only need to distribute the packages to a site directly under it and the distribution points within its site. Furthermore, the Courier Sender in SMS 2.0, allows Company L to dispatch software packages by *courier* (by packing them on CDs) to their international sites, so that they can use their slow and very expensive international WAN link for other normal business activities.

As a conclusion, Company L is a perfect example where the Microsoft System Management Server can be leveraged to achieve an advanced software distribution system in a very large and complex Windows 2000 environment.

Summary

Both Microsoft System Management Server 2.0 and Windows 2000 IntelliMirror provide software distribution features in a Windows 2000 environment. Both methods have their specific characteristics for selecting targets, advertising and scheduling distribution, distributing packages, packaging applications, installing an engine, reporting distribution status, and maintaining applications. It is important for organizations to understand these characteristics before they select one of these two methods or before they choose integrate them together for software distribution in their specific environment.

The ultimate business goal for software distribution is to reduce the cost for deploying software to large number of desktops and for maintaining this software. An organization's specific environment will dictate the appropriate software distribution systems to implement. We have discussed the following considerations in this paper: the cost of implementing an enterprise software distribution system and the scalability, reliability, manageability, and interoperability of the system. The Windows 2000 IntelliMirror is easy to implement and manage for small and middle-sized organizations. Microsoft SMS 2.0 is reliable and scalable for large-sized organizations with complex network infrastructure, and it supports down-level clients, such as Windows 95 and Windows NT.

To determine the best software distribution methods, organizations should define their requirements such as the intensity, granularity, and scale of the software distribution. Further, organizations should consider how critical the software distribution system is in relation to their normal business function, and how quickly they can deploy a business-critical application. Microsoft SMS provides

a centralized status report system to enhance the reliability of its software distribution functionality.

Finally, organizations need to consider how the software distribution traffic will impact their current network infrastructure and their normal business activities. A thorough assessment of current network infrastructure, both LAN and WAN, is highly recommended before designing an enterprise software distribution system. Microsoft SMS 2.0 has been designed to be network aware, and WAN efficient. A well-designed SMS structure performs efficiently in a complex LAN and WAN environment. If an organization chooses the Windows 2000 IntelliMirror, it should give extra consideration of the network impact of the distribution traffic when designing the software distribution system.

For further information, see the Lucent NetworkCare℠ Professional Services (Lucent NPS) website at http://www.lucent-networkcare.com, or call 1-877-369-1115 in the U.S. or 1-727-217-2303 outside the U.S.

NetworkCare is a servicemark and "The knowledge behind the network" is a registered trademark of Lucent Technologies Inc. All other trademarks and registered trademarks are properties of their respective holders.

Copyright © 2000, Lucent Technologies Inc. All rights reserved.