

Snow Distribution

White Paper – Version 4.0

Snow Software

Snow Software develops and provides software that substantially improves cost efficiency and control over software and licenses in an enterprise environment. Snow Software's line of products enables an IT-department to; publish, install, maintain, track and manage installed software throughout the entire lifecycle.

Snow Software has delivered more than 500,000 user licenses in Europe including some of Scandinavia's largest enterprises. The first product, Snow Distribution was released as early as 1995 and the company has over the years developed extensive competence within the software management area.

Snow Software's products utilize the latest communications and programming technologies and the HTTP based Snow System Services deliver a common platform, allowing easy and cost effective implementation of:

- Software Publishing
- Software Installation
- Software Maintenance
- Software Inventory
- Software Metering
- Software Management

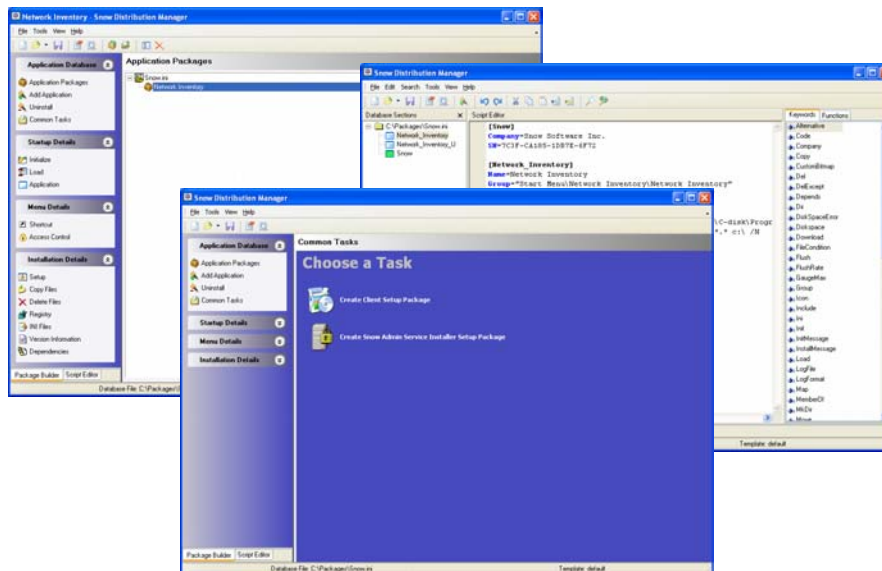
Version 4.0 - The Next Generation

09.12.2003 - Snow Software announces the release of Snow Distribution Version 4. The scope with version 4 is to make advanced package functionality easier to use and learn. Snow Distribution is with 127 functions and 64 keywords one of the most powerful software distribution tools in the marketplace, however powerful scripting can be complex and requires training.

Version 4 includes a brand new manager with an intuitive GUI and wizard based scripting that allows beginners and less experienced users to gain full advantage of all functionality within Snow Distribution. The manager also includes a conflict checker and a new and improved template editor.

A created package is dynamic and adjusts at the time of installation to different operating systems, users, other software, privileges or bandwidth. Amongst the new features you find is integration with SQL databases allowing distribution based on 3rd party systems like Navision and SAP. Enhanced security features allow the administrators to encrypt confidential information used during software distribution.

"The new Manager looks very impressive. The intuitive graphical interface gives us experienced users a very good overview and an easy start for the beginners. I can also see a clear focus on security. SnowPassEncrypt and AskPassword solve eventual security problems when automating administrative tasks" says Mats Jonsson a recognized software packaging and distribution expert at Lan International AB.



Screenshots – Snow Distribution Manager

Snow Distribution - Key Functionality

Snow Distribution is a highly flexible and powerful software distribution and application management tool that automatically distributes all types of applications, as well as updates and changes to the operating system. Software distribution is all about version control, copying files, and manipulating the registry, or .ini files.

Snow Distribution does this efficiently with a minimum of workload on the network as well as for the workstation and offers efficient software distribution to all users on the network, the main objective being to make users independent of machine and location. An application environment is dynamic not static, and in order to keep the environment efficient and up-to-date, adjustments have to be made on continuous basis. With Snow Distribution's unique "Just in time conditioning" and dependencies features you will be sure to keep your application environment running in top shape 24 hours a day.

Some of the key functionalities are:

- Flexible and centralised menu management through Snow Menu
- Generate individual menus
- Show only accessible applications on the menu
- Show different menus to different users on the same machine
- Simple and flexible scripting language
- Simple versioning system for differentiating installation and execution commands
- Just in time conditioning
- Multiple commands for solving tough special case issues
- Support for using other software distribution tools from within Snow (MSI)
- Simple dependency system
- Snapshot tool
- Create a snapshot package of applications, upgrades, or modifications
- Flexible package by using a directory structure instead of a locked down package file
- Easy management through Snow Distribution Manager
- Easy to use wizard for creating application packages

Overview

Enterprise Software Distribution is a complex task where every case has their unique challenges, but some are reoccurring in many projects. Below you will find a list of some typical issues that needs to be addressed:

- Central and distributed management
- No user interferences
- Laptop users
- Desktops
- Standalone PC's
- Home offices
- Multiple operating systems
- Multiple language editions
- Roaming users
- Multiple users on same machine
- Terminal servers
- Branch offices
- Many applications
- Variation of application need
- Individual customized application configuration
- Group customized application configuration
- Menu management
- Non-administrative privileged users
- Reinstallation
- Upgrading
- Patching
- Removing
- Updating

Dynamic Application Environment

An application environment is dynamic not static, and in order to keep the environment efficient and up-to-date, adjustments have to be made on continuous basis. The application dynamics can be divided into two categories:

- Installation
- Runtime

Within the installation category, you have:

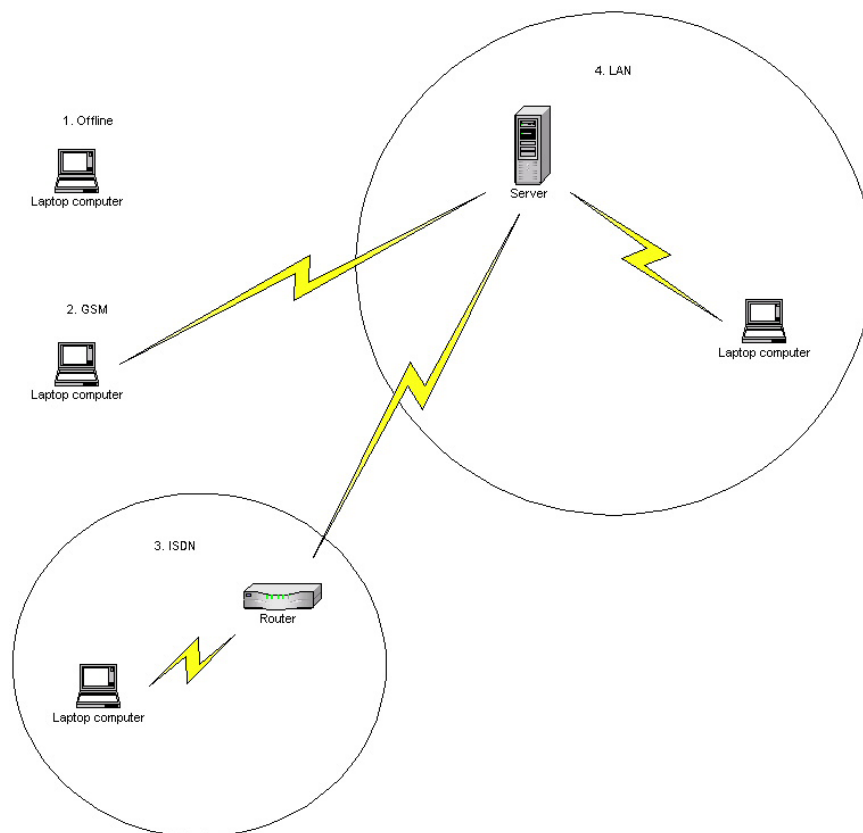
- Installation of new applications
- Removal
- Updates and Patches
- Configuration changes
- Installation of plug-ins

This need to be done from a central position and it must be simple and intuitive.

Runtime; covers application settings modified during usage, examples include:

- Applications that are available for different users on the same computer
- Roaming usage on different computers
- When the computer is offline compared to online
- LAN links compared to different levels of WAN links
- Which operating system the user are currently using

Snow Distribution is designed to be flexible enough to easily solve all these issues.



The different scenarios below describe a user using a laptop in four different situations. In the first, the user is completely offline. In the second situation the user is connected in the office through a GSM mobile phone. In the third, the user is connecting from a home office, using an ISDN line, and in the fourth, at the office connected directly through a LAN link.

1. On the laptop, several applications are installed. Among them are Microsoft Office and an in house developed client/server application. There are two application updates on the server. One is a patch for Office and the other is a new set of Office templates. The client/server application does not work if the user isn't connected to the network.
2. In the offline situation, the user continues to use Office without neither the patch nor the new templates. The client/server application does not exist on the menu, because it wouldn't work anyway.
3. When the user connects to the office via GSM, Snow checks the bandwidth and compares it with the requirements set by the administrator, and decides that there should not be any updates of neither patch nor templates. The client/server application however, is tuned so it can be used with a GSM link, and appears on the menu again.
4. From the home office, the user has an ISDN link. Snow is configured so that it enables the in house application. The installation of the new templates over the link occurs when the user starts one of the office applications. The patch however is too large so it is not installed.
5. Finally, our end user arrives at the office, with an unlimited LAN link. When the user starts Word, the installation of the patch executes and Word starts immediately afterwards.

This is just one of many different scenarios requiring a high level of flexibility. In this scenario we would use a Snow function called NetSpeed. It checks the actual speed of the network, and returns a value containing the speed. Similar commands often used in complex scenarios are; Memory, Processor and Diskspace.

The Snow script

The Snow script is designed to be both simple and powerful for systems administrators, without requiring advanced programming skills. It is structured as an .INI file with sections representing collections of commands. Keywords represent the commands and the values are strings with either functions or values to be executed. Versioning is a key concept in the script engine. The commands are divided into installation commands and run commands.

When the script runs, it checks whether the section has been installed previously and if it is up to date. If the version is OK, the installation is skipped. If it is not OK, the installation keywords are executed and the version is updated. Then the running commands are executed. The running commands are responsible for starting the application. They will be executed every time the script runs.

This versioning solution is very simple and has proven to be very efficient, both on programming- and management level. It is also very effective for troubleshooting. Simply reset the version and the application will be reinstalled the next time the user starts the application.

Dependencies

Applications, documents, templates, patches and plug-ins all have dependencies.

Meaning that they require something else to be installed first:

- Outlook 98 requires Internet Explorer 4.x
- Office 97 SR2 requires SR1, which requires Office 97 to be installed
- Shockwave browser plug in requires a browser in order to work
- A .PDF file requires Acrobat Reader
- A Norton 2000 Virus Definition update requires Norton 2000 to be installed

Snow Distribution has the ability to link scripts together according to their dependencies. This makes sure that everything necessary will be installed.

Just in time conditioning

There are several applications, which have many requirements for them to install. This will typical be hardware requirements like memory, free disc space, and processor speed, or it might be other things like username or computername.

Most software distribution systems solve this by having an inventory database they can use as a basis for targeting computers and users for installation. This requires an inventory scan upfront, and preferably a recent one.

We can do this by integrating Snow Distribution it with Snow Inventory, but prefers to do it slightly different. Instead of using Snow Inventory it is solved by putting the conditions in the installation script. Then the targeting is done during the execution of the script, and the application will be available on any menu where they are both in the targeted group to receive, and fulfil the conditions set. If these two parameters are not met, the application will simply not be distributed.

Snow Distribution can make conditions based on:

- Username
- Computername
- IP address
- Hostname
- Disk size
- Free space
- Memory
- Processor
- Existence of file, directory or device
- Time, week, month, date and year
- Operating system, version and build
- Network speed, Ping and DNS lookup
- Regional settings, file properties
- Calculations, and, or, not
- Random
- Settings in registry or ini files
- ...and even more...

Menu management

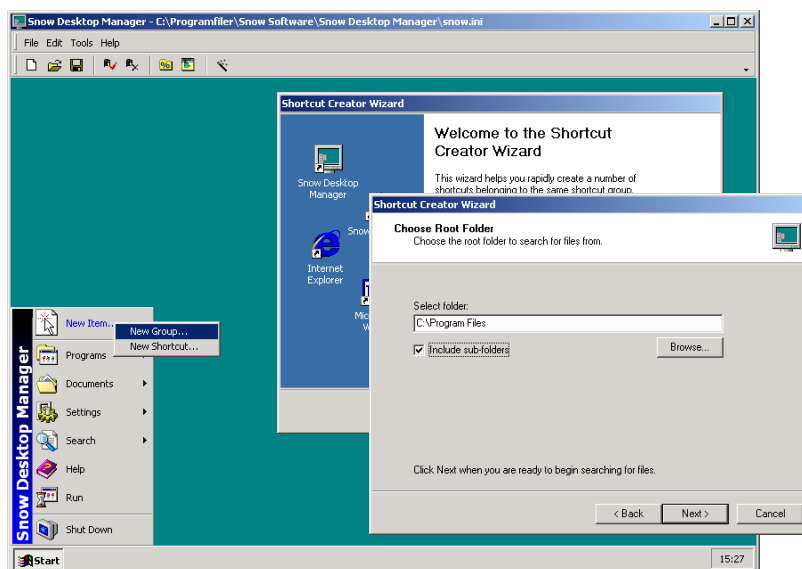
This is the process of managing the shortcuts on end users computers, both on the Start Menu and on the Desktop.

Snow uses a program called Snow Menu to take care of this. It runs when the user logs on. Snow Menu reads the snow script, looking for packages, which have a shortcut associated with it. When it finds one, it checks to see whether the user should have access to it or not. If the user has access, the shortcut is created according to the settings in the script. The application will be on the menu whether the application is installed or not. It works the same way the other way around. If you don't want to have menu choice for a locally installed application, you can use Snow to remove it. This is especially handy if you have applications, which only works under specific conditions, removing it when the application won't work and adding it when it is available.

Instead of having the shortcut point directly to the application's executable file, it points to the Snow script engine with a parameter saying which script to run. The script makes it possible to check whether there should be an installation/upgrade or just starting the application. It also makes it possible for some just in time modifications.

In short Snow Menu utilises the consequences of the network's catalogue services, and verifies the user rights before icons are presented to the user. The menu and the icons can be placed on the Start button, or directly on the desktop. The administrator decides which user or group of users will get which shortcuts simply by setting permissions per shortcut. Snow Menu is dynamic and presents only the applications to which the user has legitimate access and can thereby present different menus to different users on the same machine.

The Snow Desktop Manager is provided to help the administrator build the Snow Menu shortcut database. It has a Windows Explorer look-alike and feature rich user interface that makes it easy to build your menus exactly the way you want them to appear on the end users Start Menu or Desktop.



Screenshots – Desktop Manager

The new version

Snow Menu version 3.7 consists of two parts: Snowgrp32.exe and SnowMenu.dll. Snowgrp32.exe is the original Snow Menu that cooperates with Snow32.exe to make up the Snow Distribution product. Prior to version 3.6 Snow Menu only functioned as a part of Snow Distribution, not as a standalone application. With the new version we have included functionality for managing application shortcuts, or menus, that are not Snow Distribution related, keeping the flexibility and functionality as found in prior versions.

SnowMenu.dll is a new product that includes all the functionality of Snowgrp32.exe, and adds the flexibility of COM objects. This means that an administrator can create Snow menus as well as ordinary application shortcuts from inside a script or programming language, or even an HTML document.

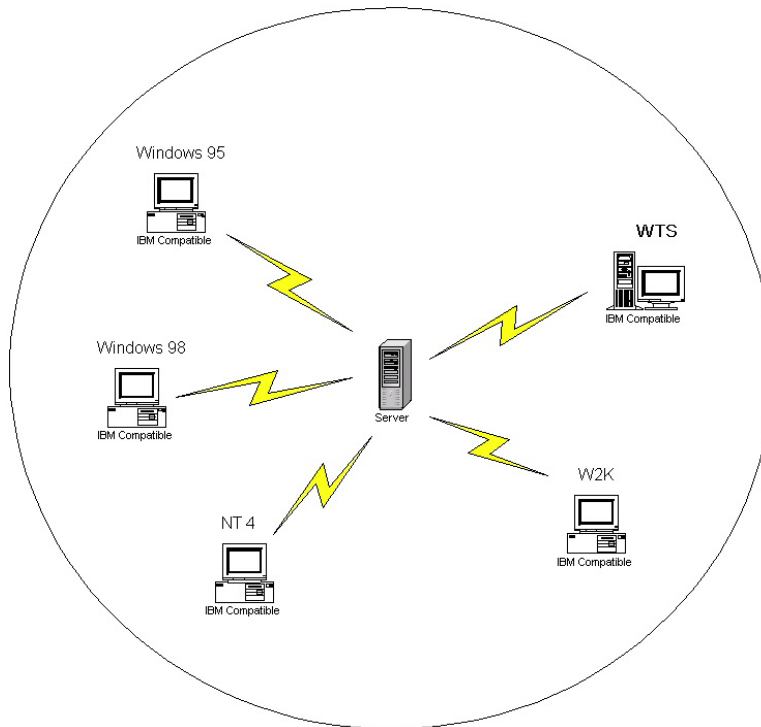
Case example

A Company wants to distribute several applications from their intranet web site as well as providing links to commonly used applications that are already installed on the users' workstations. By including a small java or VB script that calls functions from the Snow Menu COM object in an HTML document, applications can be managed from a web page rather than the desktop or start menu. Snow Menu includes functions to create icons that can be displayed in an html document, or just links to applications or distribution packages, or both. If preferred, Snow Menu can also create Desktop or Start Menu shortcuts for the same applications from the same script, by calling the Execute function from the COM object.

Multiple Operating Systems

This is one of the key challenges of software distribution. Even in environments with strict standardised policies where it's clearly stated on which operating systems to run, there will be breaking of the rules. For example, in a pure Windows 2000 environment, someone will sooner or later log in with a laptop running Windows 95/98. Or what will happen if someone logs on with a home PC with an old operating system on the network? What happens when they get software configured for Windows 2000?

Below is a sample scenario using several of the Win32 operating systems:



In this scenario the organisation has not been able to standardise the operating system. They have users running Win95, Win98, NT4, W2K and Windows Terminal Server. The users are roaming, and moves between the various systems. NT4, W2K and Terminal Server users does not have administrative privileges.

They have two applications that they want to distribute to the users:

The first application is a snapshot package generated with Snow Manager. It will be distributed to any of the operating systems. The second application comes with its own silent setup routine, and is divided into two versions: one for Win9x, and one for NT. The Terminal Server will not use this application. Snow will be used for the distribution.

The first application uses Snow functionality to create one single package to be used with every Win32 operating system. It makes sure that files are stored in the correct directory, under all operating systems, to have correct references in the registry and in the ini files.

On the NT computers the Snow Admin Service takes care of the installation part that requires administrative privileges. The package is split into user specific and computer specific installations, so that only the first user on a computer installs the computer specifics and the user only needs to install the user settings the first time. When he goes to a new computer, his configuration remains the same.

The second application requires different installation files for Win9x and NT operating systems. They have a built in routine for silent installation. So instead of using the Snow snapshot routine, Snow is used for building the menu and making sure that the correct setup program is executed according to the operating system the user is logged in to. Snow makes sure that the application will not be shown on the menu if he logs on to a terminal server.

The first scenario is an example of how flexible the Snow package is for making the most out of one single package, while the second example demonstrates how to use Snow functionality to differentiate which package should be applied.

Packaging/Repackaging

Many applications come with their own routines for silent installation (installation without any user intervention). This might be MSI script, Sysdiff packages, Setup /Q (Quiet mode), InstallShield Silent files (Scripts with an .ISS extension), IEAK or similar systems.

These routines may be good enough for the complete installation, or they might benefit repackaged customisations that are linked to them. Packaging is the process of creating a package of the application to be distributed. Most software delivery systems, including Snow Distribution, uses snapshot technology for this task.

Simply described, snapshots are based on taking a "picture" of the computer before and after installing the application that will be deployed. The "pictures" include the files, the registry and ini files. When the snapshot is finished, the pictures are compared and the difference is extracted and set up as a package. Most software delivery vendors set this package up as a single file to be distributed. This file can then either be modified through the vendor's application or it cannot be modified at all.

Snow takes a different approach. Instead of creating a single file, Snow Distribution creates a directory structure, which resembles the way the files will be installed on the target computer, and a script for how it will be installed. Both the directory structure and the script can be easily modified through either the Snow tools, or directly from the Windows Explorer and Notepad. This has two great advantages: First it makes it very simple to modify a package, secondly you only have to do the same changes on the packages deployed on remote distribution servers. No need for transferring a whole new package over a slow link.

Server-to-Server Distribution

Snow System Services utilizes HTTP and support both encryption and compression. And since it uses HTTP as transport protocol, we can utilize proxy servers for efficient distribution to the various sites. The product will make sure that the solution is synchronized. It will feature rollback, push and schedule functionality.

Snow Distribution supports HTTP & FTP in its scripts with the following keywords:

- Setup
- MSI
- Download
- Upload
- Proxy

This can benefit your solutions in many ways:

- Use HTTP and Proxy servers to reduce/eliminate the server-to-server package distribution problem, in WAN environments.
- Script downloads to be executed (nice with the Task Scheduler)
- Use web servers instead of file servers
- Install to home office users
- Upload log files to central FTP archive
- Many more...

Software distribution in WAN environments is a challenging task with plenty of room for problems waiting to happen. Some of them are mentioned on the next page.

When you have slow WAN links, it is vital that installation packages are retrieved from local servers. If you install a product from a remote server, bandwidth will be a bottleneck, so you need a solution to place packages at servers local to the clients in some way.

The more servers you have, the more administration points you must manage. Every server must be maintained and updated. And every time the central package server receives a new package, a package is removed or modified; the branch office servers must be updated. For most this is a cumbersome task, hard to overcome. Mobile and roaming users are also a headache. You have to make sure that the users always installs from the local server, even when he is not at his "home" location.

This is often handled through logon scripts of various complexities whether they are written in batch, Kix or Windows Scripting Host. Many choose to only install when the users are at their "home" location, to avoid the problems altogether, others write complex scripts, that solves the problems until they loose control and mess it up.

There are several products out on the market, which addresses these problems. Replication software or dedicated distribution systems (Microsoft SMS, CA's Ship-It, etc...). A shared trait for all these products is that they are only partially successful, and only if you have the competency, a standardized environment and the time to set it up and manage it. They do not provide a truly good solution.

The solution

Installation over HTTP via proxy

Use HTTP to transfer the packages, and let them go through a proxy server positioned at each local net.

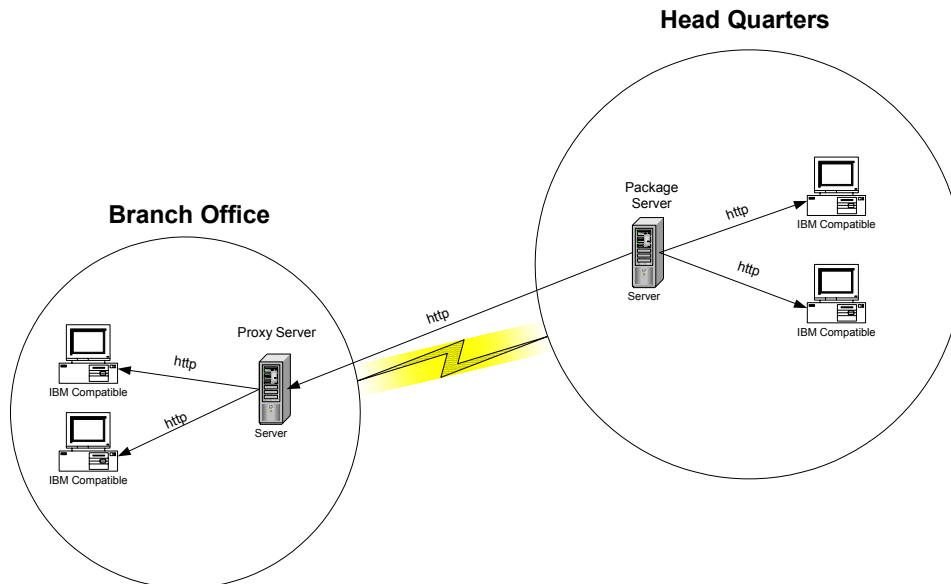


Figure 1

The figure shows a solution with a web server at the HQ, where the local clients retrieve their packages directly. The clients at the branch office also retrieve their packages from the same web server, but because they do their request through a proxy server, it is already cached within the LAN, making the installation fast and without impacting the WAN link.

The Proxy server keeps a cache with the installation packages. The users connect to the same web address no matter where he is, but the package is always being installed from a local proxy cache. This way you obtain LAN speed everywhere, but you still have a single package storage location.

Administration of the packages is done at one single server at one location, the main office. This way, the management for complex WAN environments becomes as simple as a small LAN. When the package is no longer needed, it is automatically removed, when the cache expires period is over.

What about the first user installing the package?

The first user will have to wait for the package to be transferred over the slow WAN link, if you do nothing more than setting up Proxy servers. The solution to this challenge is however quite easy. You simply set up a scheduled download script at each branch office. This download script works as a "Proxy refresher", and should run regularly before the package expires from the cache. This way, all users experiences LAN speeds, including the first one.

Installations must be initiated from the client. Some alternatives are: Logon scripts (W2K: logoff, startup & shutdown also), Start Menu (Advertise), from Add/Remove Programs, or Task Scheduler. If you use Push technology, it is vital that the push system asks the client to initiate the installation. This distribution concept will not work if the push system transfers the package all the way to the client.

The HTTP + Proxy solution does not require a proprietary software distribution system it only requires that HTTP can be used to transfer the packages, and that it can keep the proxy cache fresh. You can for instance use it with the Windows Installer, using the following from batch or command line:

```
MSIEXEC.EXE /QN /I:"http://www.snow.no/data1.msi"
```

You can use a Microsoft Proxy, either the old Proxy Server 2.0 or the new ISA server; or any other proxy from vendors such as Novell, Cisco and others. The investment is not just a software distribution investment; all HTTP traffic within the organization benefits from proxy usage.