Effective Use of Individual User Profiles with Software Distribution

Mark Murray
Interprovincial Pipe Line Inc.
Troy Roberts
Interprovincial Pipe Line Inc.

Abstract

Many system administrators have found themselves supporting user communities who are either quite mobile and use many different workstations or share one workstation among multiple users. Within Windows NT networks, these users can still maintain unique personal settings for the software packages they each use through the use of individual profiles. Individual user profiles have, however, introduced a problem in the typical software distribution process.

During software installations, whether software distribution tools are used or not, there is usually a portion of the "package" that requires updates in each users personal profile or configuration storage area (often their home directory). These updates include default, and often required, user customizable settings and can only be performed for the user that is logged in at the time the package is distributed to each workstation.

Problems occur when a user who was not logged in during the distribution of a particular package attempts to use that package. This user will not have the required registry entries or configuration files and the software package will not work properly, if at all. Some software packages work around this by creating a default configuration on the fly, but they are not in the majority, and this does not allow organizations to customize the default settings that a user will receive.

At Interprovincial Pipe Line Inc. we have implemented a combination of commercial and home-grown utilities that provide a flexible and complete software distribution mechanism to a network of Windows NT 4.0 client workstations. These utilities specifically address the issue of applying individual user defaults and customizations after an application has been installed.

1. Overview

During a recent corporate rollout of Windows NT 4.0, several hurdles were encountered when developing an effective desktop rollout and desktop management framework. Some of these hurdles were dealt with by using widely available software solutions, while others were handled by creating in-house solutions.

The specific problem addressed by this paper, is the co-existence of roaming user profiles and software distribution/installation. At the time of our implementation, tools were not available that addressed the issues created by implementing both roaming profiles and software distribution. Both of these, however, were required to meet the design principles in place at Interprovincial Pipe Line Inc. (IPL).

This paper describes, in detail, the computing environment at IPL, the requirements for roaming profiles and software distribution and the issues created by implementing them together. Also outlined, is one solution we have implemented at IPL that addresses the problems presented. This solution has been in successful production for approximately 6 months at the time this paper was written. Prior to implementation, an independent review by Microsoft of this solution provided these remarks:

"The software distribution system which has been created at IPL solves some very difficult problems with respect to automating software installation in a large company environment on the Windows NT Workstation platform. The solution demonstrates a clarity of understanding to do with software installation on that platform that I rarely encounter."

– Alex Nicol, Microsoft Consulting Services.

As with any solution, this may not be right for everyone. It has been working well at IPL and we have not yet seen any other solutions presented.

2. Background

Interprovincial Pipe Line is engaged in the business of operating the world's longest, and possibly most complex, pipeline network for transporting liquid hydrocarbon products. Geographically the organization is spread from Canada's Northwest Territories across North America into Chicago and farther east to Montreal. There are over 30 manned pumping stations connected to the companies' Wide Area Network and close to a dozen district offices each with a number of field and office personnel, all requiring access to various elements and services from the corporate computing environment.

Like many other large organizations, our environment has evolved over time through a variety of technology platforms. The current computing environment consists of a heterogeneous mixture of VAX/VMS, UNIX, and multiple versions of Windows. In order to minimize the maintenance and support activities (and staff) a number of design principles have evolved that are applied across all platforms, including:

1. Users should be able to log into the systems that they have access to from any desktop, in any office, with seamless access to the same files and working environment that they would have on their local workstation. This eliminates the need for a user to have multiple different logins when traveling or connecting to the network in a field office.

2. Users should be abstracted from the technical details of where computing resources are physically located, but rather should only have to deal with selecting the required resources from online lists. The users should not have to know or worry about what piece of hardware the database server is physically located on, and what other applications are hosted there, they should only have to know that there is a database service available on the network on a known connection point.

3. Administrators need to be able to perform troubleshooting and system maintenance remotely, again from any workstation on the network. Using remote control and systems management tools, systems administrators should be able to manage systems located in the field sites without having to travel to any of those sites.

4. Patches and software enhancements should not require a physical visit to a workstation. In the past, any kind of software rollout or upgrade required a visit to each desktop system to install the update. Due to the geographic spread of the company, this is a very costly exercise and as a result, software was not frequently upgraded.

5. User login IDs and resource names should be consistent across all platforms. The name of a printer should be the same regardless if it is being accessed from Windows, UNIX, or VMS. Likewise, a user should not have to remember multiple different user names depending on which platform they were connecting to.

6. Users should have to deal with a minimum number of desktop devices. From a desktop perspective, the corporate standard that was deployed consisted of Windows 3.11 and DOS 6.2. A number of end users also had UNIX workstations and VT terminals deployed on their desktop in order to support mission critical applications and legacy systems. In some cases a user may have two, three, or four different desktop devices an their office depending on what systems they needed to access.

7. As much as possible, consistency of software deployment and initial user preference selections must be established. In the past this has been elusive because many individuals have been involved in manual software distribution projects. Automated software distribution and installation tools will help to alleviate this problem. This will enable easier support and more predictable installation of software and patches within the environment.

The former operating environment for Intel desktops (DOS/Windows 3.11) was beginning to show many limitations, and was a jumble of different configurations depending on who last worked on the system. As a result, an upgrade project was initiated with the goal of deploying a standard desktop environment to our 700+ PCs, hopefully meeting some of the design principles above.

When deploying a new operating system to a large number of desktop PCs, a strategy is required. At Interprovincial Pipe Line Inc. (IPL) the deployment strategy was influenced by the existing corporate computing environment mentioned above.

In attempting to create a deployment strategy and select products to include for ongoing management of this environment, we first defined what would be the “ideal” rollout. This was generally defined as follows:

- Windows NT 4.0 as the desktop operating system.
- Fully automated software distribution and installation to desktops.
- All current and future applications packaged for distribution and not installed manually (to ensure consistency).
- No data or user configuration settings stored locally, but stored instead in the user’s profile.
- Roaming profiles available from any location on the network.

The following are examples of the kind of problems we encountered in attempting to implement the ideal solution:

- Almost 1/3 of our PCs are portables on which NT is not effective (portable and non-network attached PCs).
- Many PCs are not connected to a LAN with a local server. This means that there is no distribution point for software distribution.
- Windows NT 4.0 would not run all of the legacy applications we still maintain.
- Full software distribution and roaming profiles are relatively ineffective over slow network connections.
- When roaming profiles are enabled, software distribution is ineffective for many applications.

For the first four points mentioned, an alternative to the ideal desktop was developed using Windows ’95. This alternative included the use of software installation tools with a manual distribution process. This ensures that over time, installation consistency on the Windows ’95 desktops will be maintained. The alternative rollout has been used on approximately 1/3 of our PCs.

The last item mentioned is where the title of this paper comes from and is discussed in detail throughout the remainder of this paper.

3. Problem Analysis - Using Roaming Profiles and Software Distribution Together

Roaming profiles provide significant benefits by allowing users to use any desktop to access the systems they require, in any office, with seamless access to the same files and working environment that they would have on their local workstation. This meets one of the primary design principles applied to all environments at IPL.

Implementing automated software distribution and installation allows us to meet some of our design principles as well. A problem arises, however, when roaming profiles and software distribution are implemented together. Since multiple users will use the same software installed on many PCs, the user preferences and settings for each software package must be updated within each user’s profile. This section of this document describes how typical software installation procedures and also current software distribution tools do not effectively handle this.

The third section of this document discusses some of the technical details of the solution implemented at IPL to resolve this problem.

3.1. PC Profiles and Roaming User Profiles

For Microsoft professionals, the term “profiles” or “user profiles” both refer to the files and directories that define each user’s registry, desktop, start menu, etc.. Using a Windows NT domain configuration, it is possible to have “roaming” profiles that are stored on the server and loaded down to the PC each time the user logs in and updates are copied to the server when the user logs out. This feature allows users to roam between PCs on the network, while maintaining their personal preferences and customizations. Each user becomes effective on all networked PCs. The use of profiles, and particularly roaming user profiles, helps meet two of the design goals mentioned above (remove data from local PC and seamless roaming users).

The Microsoft "profile" model works well for applications that fully and properly utilize the registry and the Windows NT user profile directory structure. This is not the case with all applications. Many applications still rely on .ini files and other user specific files that must also follow the user when they log in at different workstations. These files are not always accessed and if they were included in the user's profile directories, the process of copying them locally to the PC during the login process would cause significant delays particularly over low speed WAN connections.

In order to accommodate the user specific storage of .ini files and other application configuration files, we must add to the Microsoft concept of a profile. What needs to be added is a filespace that is user specific, and available to the user no matter where they log in, without copying it local (this would take to much time). This concept of a profile can be applied the same way to each PC. This has been accommodated within Windows NT through the NT profile known as "All Users" and the fact that there is a locally available filespace that applies to the PC (the local hard drive).

This leaves us with four areas to manage:

- NT User Profile (a directory named after the user, copied local during login)
- User Filespace (a configuration file directory that remains on the network but is available during user sessions)
- NT PC Profile (a typical NT profile called "All Users")
- PC Filespace (the local hard drive)

In order for a PC to run with working software, all four of these areas need to be present during any user session. The separation of each of these four areas is important because during software installation, each area has the potential to require preparation before the software will run properly. As will be discussed later, when roaming profiles are enabled and software distribution is used, these are never available for all users of the software at the same time to perform proper software installation. At IPL we have developed an effective workaround to this problem that is outlined in this paper.

The following sections provide more detail on each of these areas.

3.1.1. Typical NT Profiles

Typical NT profiles are comprised of two parts. One is the sections of the registry that it will contribute to the total registry available during a user session (this is different for Users than for PCs). The other section is the directory structure that is usually not separated from the registry (these directories are copied to the PC as part of the roaming profile). Within the registry, it is possible to tell NT to expect the directories to exist in another location. If this is done, these directories are accessed from their location on the network, without being copied local, which is more effective.

3.1.1.1. The Registry

The development of the Windows NT registry has provided many advantages to system administration professionals, particularly those of us who administer medium to large Windows NT installations. When used properly, the advantages include remote registry viewing and updating and a single location for all configuration information to be stored that can be backed up and restored as a single unit. The registry also promotes consistent design of applications by providing consistent tools and examples for registry use.

One feature of the Windows NT registry, is its ability to separate the storage of configuration settings that are specific to the machine itself, and the settings that are specific to each user of that PC (the settings may be different for each user).

For the discussion of software distribution and installation, there are three of the five registry sections that we are concerned with:

- HKEY_LOCAL_MACHINE - contains the configuration information that applies to this particular PC. All users that login to this PC will be affected by these settings. Usually, settings related to hardware configuration (e.g. network configuration, video device settings, etc.)
- HKEY_CLASSES_ROOT - contains references to and information about the object classes that are available to this PC. The list of available object classes will grow

with the installation of additional software packages that use objects.

- HKEY_CURRENT_USER - contains the settings for the user that is currently logged into this PC. This section will either remain consistent for all users that login and be shared by this group of users or it will be populated during the login of a user (either from locally stored settings or from a roaming profile available on the network).

These registry sections can be categorized into two groups as follows.

Registry Sections Local to the PC

These registry sections are stored locally on the PC and form a part of the profile for the PC.
HKEY_LOCAL_MACHINE
HKEY_CLASSES_ROOT

User Registry Section

This registry section is stored with the user's roaming profile and is loaded to the PC during the user login.
HKEY_CURRENT_USER

It is expected that the installation of software may have updates and additions to apply to each of these registry sections. The fact that they are stored in different locations forms part of the problem for software distribution.

3.1.1.2. The Directories

Microsoft has defined within NT the directories that form part of a user or PC profile. The PC profile directories are stored in the "All Users" profile directory. These remain local at all times and therefore require no changes.

The user profile directories are copied to the PC along with the registry and are stored in a profile directory named the same as the user's name. As mentioned above, if certain registry settings are modified, these directories will be referenced in a different location and will not be copied along with the profile. This method allows for a more efficient login process.

The user profile directories have restricted access privileges that match the privileges on the rest of the user's home directory, which means that usually only the user has rights to modify them. Any process that needs to manage them must be executed as the user.

3.1.2. File Spaces

The filespace that is dedicated to software installation and configuration settings is different for the PC than for the User. The filespace dedicated to the PC is required locally and is not required to be available to any other area of the network. This area is located on a local hard drive for each PC.

Each user's filespace (designed to store user specific software configuration files) must be available to the user from any location on the network if we want roaming profiles to work properly. To accommodate this, the filespace could be included as a sub-directory located in the user's home directory. This share is available to the user (and is connected) anywhere where they log into the network.

The user and PC filespaces each require different privileges to access. Some parts of the local hard drive (such as the WinNT system directory) are locked down to prevent damaging updates by user software installations. These areas require a local "admin" level of privilege in order to update. Each user's area is restricted to that user only. Most of the user's don't have any "admin" privileges on the network PCs.

This raises the question; Which account should be used for software installations if there is no account that has privilege to both of these areas and to the NT profiles discussed above?

3.2. Software Installation - How it affects the registry and file spaces

When software is installed, the installation process may update each of the four areas mentioned above. The PC portion of these updates will be required on each PC where the software is to be executed. The User portion will need to be applied to the NT profile and filespace of each user that will run the software package. The updates to each of the four areas is as follows:

- PC Filespace – the actual software is generally installed locally within the PC filespace unless it is to be run from the network drives.

- PC Profile – the configuration settings for the software that apply to the specific PC or have been selected for the corporation are typically stored in the PC profile.
- User Filespace – this is where we prefer to store software configuration files that are specific to the user. Examples include .ini files, Internet browser bookmark files, or user specific security files. As is described below, some packages require some customization in order for them to look here for these files instead of on the local PC.
- User Profile – user specific configuration settings are stored here. This mostly applies to applications that are registry aware and make use of the registry properly. A good example is Microsoft Office '97.

Some software packages do not make a distinction between the user or PC specific configuration settings and software files. Most packages do however, have a means of customizing the location of the configuration files through setting in an .ini file, the registry or command line options. Others can be customized by properly selecting the application working directory or the PATH that the PC uses.

3.3. Software Distribution - The issues

Software distribution involves more than just installing software on each PC. In fact I believe that "Job" distribution would be a more appropriate term. One type of job (by far the most common type) is a software installation job. Other types of jobs include; hardware and software inventory jobs, anti-virus jobs, and any other job that can be scheduled and distributed to a group of PCs or users.

Job distribution includes:

- distribution of jobs (delivery to the PC the job is to be executed on)
- identification of which user or PC should get a software distribution job
- identification of which account should the job execute as (each job must have the correct privileges to execute)
- scheduling when and in which order jobs are processed

Most jobs are easy to distribute to PCs. A user or PC can be selected to receive the job, an ID for the job to run as can be identified and the job can be scheduled for a certain time/date. An issue develops when a job must run partly under one ID and partly under another ID, or when a job needs to run at this time to get part of the job done and at a different time to get a different part done. This is the case with many software installation jobs.

Software installation jobs might need to update a PC profile and each user profile that is going to use that PC. These profiles are not all available at one time to ensure that this software installation job can be executed completely. Additionally, each profile requires the installation job to run as a different user ID to permit updates. In order to accommodate this, these packages must be split into independent parts that can execute completely on one profile (user or PC) and as a single user ID. These usually go hand in hand.

Splitting a software installation job into two parts (the user profile and filespace updates and the PC profile and filespace updates) means that we must independently schedule and deploy two jobs instead of one. The target for each job is different as well. For each software package, we must now decide which users will use the software and on which PCs they will be allowed to use it. We must also have a distribution tool that allows for a very specific definition of a target (specific users or PCs) and schedule for deployment of each package.

In determining the target and schedule for the "user part" of the software installation and the "PC part" of the installation, a couple of points must be remembered:

- for a software package to work on any particular PC for a certain user, the PC part must be installed on the PC and the user part must be deployed to the user profile and filespace.
- prerequisites must be met - other software packages that this package is dependent on must be installed first

3.4. The Software Distribution / Roaming Profile Problems

If there were only one user profile per PC, as is the case with Windows for Workgroups, then installation of software packages would be a lot easier. A single job would be sent to that PC the next time any user logged in and the user and PC profiles stored locally would be updated. This is not the case when roaming profiles are enabled. If a complete software package install was delivered to a PC when any user logged in, only the PC profile and THAT user's profile would be updated. Any other user that logged in would not be able to operate the software correctly because the updates required to their profile were not done.

If only there was no security implemented on PC or user profiles, as is the case with Windows for Workgroups, then installation of software packages would a lot easier. With Windows NT, and roaming user profiles, different login ID's are required to make updates to the PC profile than to make updates to each user's profile (each user profile requires that user's privileges).

These two issues are not currently overcome by any software distribution or automated installation utilities without some extra customization.

4. Detailed Discussion of The Solution

4.1. Tool Selection

In developing a solution to the problems of combining roaming profiles and software distribution and installation, the first thing that needed to be identified was the tool set that we would use. Since no single tool met all of the requirements, the tools selected would have to be the most flexible tools possible.

4.1.1. Roaming User and PC Profiles Tools

When used in conjunction with system policies, Microsoft has developed a fairly complete and flexible solution to roaming profiles. There was no reason to not use roaming profiles as implemented by Microsoft.

The only missing component in the roaming profiles is the file spaces. For PC file spaces, the local hard disk in its entirety stores the PC based application configuration and setup files. The local hard disk is assumed to be the "filespace" for each PC's profile.

For the roaming user profiles, a file space that will be available to the user without being copied as a part of the roaming profile was required. For this purpose, at IPL we created a sub-directory within each user's network home directory called "config". This was named this way because typically the type of files that will be stored here are user specific, application configuration files.

We have also set it up, using system policies, so that the directories that are part of the user's profile that have the potential to be large are referred to within the user's configuration filespace. Once these directories are referred to in another location, they are not copied local, but are instead expected to exist in the referred to location.

4.1.2. Software Distribution Tools and Automated Software Installation Tools

In selecting appropriate software distribution and automated installation tools, three products warranted full evaluations.

- Seagate – WinInstall
- Microsoft – SMS
- McAfee – Saber Lan Workstation

Tools from other vendors were reviewed but were rejected for reasons not related to software distribution or installation.

In addition to the standard software selection criteria, (such as stability, supportability, etc.), one of the key factors was to purchase tools for software installation and software distribution that supported the concept of split packages.

WinInstall, by Seagate Software, is a product designed mostly as a tool for automated software installation. There is a software distribution component to the product but it is not as strong as other products. The software installation component, however, is very good. Not only is the product flexible enough to support splitting software installation packages into the user and PC parts, but they have in fact automated the package splitting process in their more current releases.

WinInstall is marketed as a product that can augment many software distribution tools (including SMS) and as a software installation tool was a very good choice at IPL. WinInstall was selected as the tool for automated software installation, but not for job distribution.

The other two products were not focused as much on the software installations, but dealt with job distribution and scheduling far better. Both of the other two products offered centralized job scheduling, a more "corporate" perspective and provided greater scheduling flexibility. At the time when we needed to choose a particular product for job scheduling, McAfee was unable to deploy a proper 32-bit tool compatible with NT 4.0, which we needed.

Microsoft had just released version 1.2 of SMS, which met much of our selection criteria, but fell short when handling the distribution of software installation jobs that were split into user and PC components. SMS 1.2 could not deliver software installation jobs identified to be distributed to certain users. This made distribution of split software installation jobs, discussed above, impossible.

We concluded that the best solution to the tool choice was to augment the job scheduling features of WinInstall with a web based centralized job scheduling utility, built in-house.

4.2. Implementation

4.2.1. User and PC Package Installation

Current software distribution tools deploy a software package as a single unit. The two problems identified above can only be handled by splitting the automated software installation instructions into user and PC components and deploying them independently. In order to do that, the tool to be used for automated software installations must be flexible enough to support splitting packages. Once a package is split, there is the issue of deploying the parts.

First, the PC component of the package must be installed with local administrator privileges. This can not be initiated during user login because the user typically does not have the required administrator rights. It is also not feasible for someone to come to the PC and log in as administrator to start the software distribution process (in fact only half of it) because that would defeat the concept of software distribution. Therefore, this must be done through a service.

Using WinInstall, a service can be set up to run as administrator, and continually check for new or updated PC components for installation. If a new package or an update is discovered, it can be safely started from the service with the proper permissions. WinInstall has the ability to run as a service and continually check for updates in a "package listing" file on the PC. The next issue is how to populate the "package listing file" for each PC from a central (corporate) and managed database.

This is accomplished during a user's login session. When a user logs in, the login script checks a database on a central, corporate server to determine what PC components the machine scheduled for. This information is entered into the HKEY_LOCAL_MACHINE part of the local registry. The next time the WinInstall service checks the package listing file to determine if there are updates, it will detect new information and install them.

Once the login script has updated the list of PC components that should be installed, attention is focused on what user components should be installed for the user that is logging in. This information is stored in the HKEY_CURRENT_USER section of the registry. This allows information about which packages a user has been assigned to follow the user even if access to the software distribution database is not available.

In order to apply the user based updates, the login script directly initiates a process. This process (since executed by the user's login script) run as the user's ID and therefore has the privileges required to apply updates to the user's profile. The decision to install the user components is sometimes based on the fact that the user is currently logging into a PC with a particular software package and some user setup is required before they can use it. If this is the case, the user component is automatically installed. These settings will then follow that user in their roaming profile. That user can now use that software on any PC where it is installed.

4.2.2. The Central Software Distribution Database

During each user login, the login scripts query a central software distribution database to determine which software should be installed on the PC and which software the user should be set up for. This information, once retrieved from the central database, is stored in the PC or user's section of the registry respectively. If contact to the central database is not possible, the information that was available during the previous login attempt will be used.

The central software distribution database is accessed using WinSock calls from the login script to a server process running in one location within the wide area network. The server process feeds information in response to the queries from the client.

The actual database is accessed through a series of PERL scripts, the core structure of which is shared by the software distribution mechanisms built for our UNIX workstations. Access to the database for administrators who will provide updates, is accomplished through a series of PERL based, interactive and access controlled web pages that have been included as part of IPL's intranet.

With this web page system in place, it makes it possible for anyone identified as a software distribution administrator to make database updates from any web accessible location.

5. Follow-up and Futures

At the time of this writing, this solution has been in production for approximately 6 months. We consider this solution to be quite successful and have also found it to be extremely flexible in the short time we have been using it. There has been some discussion about the following potential further enhancements:

1. We would like to move the central database into Oracle (which is our corporate database platform). This will enable us to make use of the information in many additional ways through increased reporting and access capabilities. A move of this type should have no operational impact on the solution, as it does not impact the method that the clients use to communicate to the server process.

2. It would be beneficial for us to further develop the capabilities of the web page interface and the access control to the web page. This would permit additional levels of read only or role based access to the web pages.

3. In the review done by Microsoft, they suggested that the software distribution database be linked to our UNIX and NT account creation tools and even perhaps our human resource systems. We would like to see this as well, starting by linking the web page to the account creation tools.

As Microsoft continues development of NT 5.x and future releases of SMS, they have indicated their commitment to "zero administration". We do anticipate growing this solution to participate in future administration tools shipped with NT and possibly to a commercial software distribution system such as SMS when they overcome the issues that our in-house solution addresses.