

Windows User Profile Management

A Solution Overview

Includes:

- ▶ What is the Windows User Profile?
- ▶ Standard Approaches to User Profile Management
- ▶ Solving the problem of User Profile Management

AppSense[®]

Contents

Introduction	3
What is the Windows Profile?	3
The Challenges	4
Standard Approaches	5
Group Policy Objects	5
Login Scripts	5
Logout Scripts	5
Profile Management – Solved in a unique approach	6
AppSense Environment Manager Overview	7

Introduction

This document details the issues that are frequently faced by IT administrators of the various Microsoft Windows platforms when it comes to deploying business applications to users. The document will detail common issues as well as common methodology that is often employed in order to combat these issues. We will focus specifically on issues that come to light when needing to deliver well performing workstations / servers across the enterprise whilst maintaining valuable user specific information. Finally we will focus on solutions that make the management of all of the highlighted issues a thing of the past by introducing some unique methods to control the environment from the administrator's desk.

What is the Windows Profile?

On computers running Microsoft Windows Operating Systems, user profiles¹ automatically create and maintain the desktop settings for each user's work environment on the local computer. A user profile is created for each user when he or she logs on to a computer for the first time.

The profile defines customized desktop environments, which include individual display settings, network and printer connections, and other specified settings.

User profiles provide several advantages to system administrators:

- More than one user can use the same computer, and each receives personalised settings when he or she logs on.
- When users log on to their workstation, they receive the personalised settings as they existed when they logged off.
- Customization of the desktop environment made by one user does not affect another user's settings.
- User profiles can be stored on a central server so that they can follow users to any computer running Windows NT 4.0, Windows 2000 or Windows Server 2003 / XP on the network. These are known as "roaming user profiles". A little more information can be found about roaming profiles later in this document.

The System Administrator can elect to make use of the "local user profile" that is created the first time a user logs on to a machine and is stored on a computer's local hard disk. Any changes made to your local user profile are specific to the computer on which you made the changes. The settings stored in the profile are not reflected on any other machine that the user logs onto.

The advantages of personalised desktops and application

configuration can be taken to the wider enterprise by making use of "roaming profiles" where the profile is stored centrally on a file server, and copied to the workstation at login time and then back out to the central location at logout. This offers the advantage of the users' settings following the user to any Windows device that the user has the ability to log on to, and hence the user always has consistency in the way that they interact with corporate desktops.

The final type of user profile available to the systems administrator is 'mandatory profiles' that exist to prevent users from altering any settings within their desktop environment that will persist to the next login session. This sort of profile is ideal for the kiosk scenario where the administrator does not wish for the state of the machine to alter under any circumstances, but is perhaps too stringent for ordinary use within the organization, as no user preferences are able to be stored on a per user basis.

To make all three types of profiles function more effectively the Systems Administrator can create a "default profile" which is effectively the starting point that loads when a user logs onto a machine (and an existing mandatory or roaming profile does not exist). This enables the administrator to pre-configure various options prior to the user beginning to work on the machine. This default profile needs to be carefully constructed before the user(s) accesses the computer for the first time to ensure that the relevant settings are delivered to the user at profile creation time.

1. For more details of user profiles please consult <http://technet2.microsoft.com/WindowsServer/f/?en/library/40140a2e-950a-41ea-85e5-c7f1eb27939b1033.msp>

The Challenges

The Systems Administrator has many things to consider when delivering the application content to the user population.

The first such question that springs to mind is “does the end user require personal settings?” If so the administrator must first consider Roaming Profiles, and then to determine where to store the roaming profiles so as to be accessible at all times for the users. Finally, which settings need to be setup ready for the user prior to them actually accessing the computer(s) as part of the initial “default” configuration?

The Windows User Profile can very easily grow in size to be 100's of MB's in size, which in itself presents several issues to the enterprise, not least of which being a group of files and folders (the profile) that are copied down to the workstation / server each time a user logs in, and then copied back up to the central profile area at logout. This in itself can cause huge performance degradation during these times and can result in heavy network utilization as well of course.

If the organization delivers application content via a Terminal Services layer then this issue can be compounded further, due to differing servers delivering different types of applications. Therefore, as the user accesses the application set, multiple connections to different computers will be made, all of which will be making use of the “roaming profile” simultaneously and when the user closes the applications, the profiles will be copied back to the central location at similar times, leading to potential contention in file overwriting. The worst case scenario being that the roaming profile itself in the central location will be come corrupt and the user will lose their personalised settings as the profile will need to be deleted in order to continue. In any other situation, two (or more) different computers attempt to place their local copy of the profile back onto the central location resulting in a contention issue, i.e. which settings are actually saved? The only rule that applies in this case is that the last computer to be logged out from will be the computer that provides that last set of complete data to the central profile area, and therefore anything that was in place previously will be overwritten (and hence lost).

As a note, the cost associated with corrupt profiles to a typical business can be very high but also very difficult to track down. For example, the cost to the systems administration team is most likely very low since all they need to consider is deleting the roaming profile on the central server and then asking the user to start again. This delivers the end user into the position where all of their previous personalised settings have been lost, and that they need to re-configure these personalization's. This will no doubt inconvenience the end user tremendously and could take many

hours to return to the point that they were at prior to the issue, potentially through no fault of their own.

The next thing that the systems administrator needs to consider is once the decision regarding personalisation has been made, the challenge becomes the configuration of the environment in which the user will function. For example, how do we configure those personal options, what desktop items do we deliver, which printers will the users receive and which network file systems will be mapped?

Finally, in a Terminal Services / Citrix environment we often find different applications are segregated on to different servers. In such environments, variations in the profiles are required to avoid conflicts and issues such as users clicking shortcuts not matching the installed application set. This is also an issue in traditional FAT client environments where users may roam from one computer to another, sometimes in various offices. These situations should be reflected as differences to the user's environment, for example attaching to printers in the office where the computer is, or application shortcuts should be altered as once again the installed application set may vary.

Therefore the profile is not solely dependant on the user but other factors such as location and device. All of these factors must be considered when creating a fully automated profile solution to ensure that the user receives the functionality that they require, while minimising the administrative overheads of supply.

Standard Approaches

There are many standard approaches that have been taken over the years to deal with the very issues highlighted above. A typical solution tends to be a mixture of different approaches that together combat many of the issues of managing the user and have not necessarily been aimed at managing the actual profile. The most common approaches are listed below.

Group Policy Objects

In the delivery of Windows 2000, Microsoft introduced the Active Directory, which brought with it the Group Policy Object (GPO) that would be applicable to Windows 2000 / 2003 / XP desktops and servers. The GPO is a powerful mechanism to pre-define common configuration that the Systems Administrator wants setting for a specific user / group on a specific device / location for a specific application.

The GPO is the place that a Systems Administrator would typically configure some desktop / application settings that must always be set to the same value, regardless of what the user wants the value to be.

The GPO can also make use of login scripts (to fit in with previous approaches) as well as logoff scripts to give a rudimentary mechanism to effectively write information out to disk during logoff and can then write the information back into the registry during the login sequence to grant the ability to "manage" the user profile.

Login Scripts

The traditional login script has long since been the defacto method to configure enterprise required options for a user and as it sounds, the login scripts executes during login, and hence is a one stop set and forget solution in that once the value has been applied, the script has performed its job. The user will be able to make changes to the value (by means of the application set) for the duration of their login period.

A typical login script will connect network share mappings, printers and perform other tasks such as ensuring corporate email clients are correctly configured as well as copying necessary files and / or folders into place within a users' home directory or profile.

Login scripts have been historically written in the standard Microsoft command script language, although KiXtart² and Visual Basic Scripting (Windows Script Host)³ have become more commonplace over the last few years due to their further levels of granularity of control.

Logout Scripts

Logout scripts were really introduced to most systems administrators when they became an option within the Active Directory GPO's. They are typically used to extract data to file during the logoff process, with typical examples being things such as user preferences and other application specifics that are copied out to the home directory for later use (usually to be put back in during the next login sequence).

As with logon scripts many different scripting languages may be utilised, but typically the same language will be used as used in the login scripting as the same script developer will have been responsible for both types of script.

Profile Management – Solved in a unique approach

As can be seen there are many elements to the subject of user profile management, and only when they are all brought together do they get close to representing true profile management.

AppSense Environment Manager is a unique technology with the distinct objective of removing the burden of managing user profiles but also that of other general desktop management tasks. This is achieved by leveraging the trusted AppSense Common Framework system to ensure that enterprise deployment (of software and configuration) is completely taken care of regardless of computer location (the computer does not even need to be on the physical network in order for the rules to be applied), thus allowing the administrator to focus on the task of ensuring that the desktop and application settings are managed easily and efficiently using the Environment Manager MMC (Microsoft Management Console) Interface in a drag and drop environment.

Environment Manager allows the enterprise administrator to make use of mandatory⁴ profiles (hence removing any issues with large unwieldy roaming profiles) and to make use of the profile management options within the product to securely store important user information. Quite simply during the logoff process, the Environment Manager Agent will save relevant registry entries (and any necessary files) out to the users' home directory (or similar user owned location) such that upon any subsequent logon, the information can simply be uploaded onto the computer in question, ready for the user when they begin to work. This mechanism removes the issues of large unwieldy roaming profiles by only ever copying relevant user information and simply allowing anything else to be deleted (the deletion being a standard process with mandatory profiles) as part of the logout process. Furthermore, user profile corruption becomes a thing of the past since no longer is there file copy contention during the logout process, leaving the support teams to better spend their valuable time in other projects or initiatives.

The Environment Manager product presents itself as a simple to use Microsoft Management Console (MMC) to the systems administrator and enables rapid setup of profile management options that can be active within just a few minutes. The administrator may also select to perform many other tasks (than simply profile management / hiving) during the login process (or log out process for that matter), such as connecting / disconnecting printers and / or network drive mappings, making registry edits, creating / copying / deleting files and / or folders, creating shortcuts on the desktop or start menu, system policy edits as well as any other login configuration options that may be required. Therefore, Environment Manager is not only able to deal with the management of the actual user profile but is also completely able to create the profile in the first place, delivering a true end to end profile management solution.

The administrator may configure the users' working environment relating to the physical device being used, the user themselves, the location or a combination of these offering unlimited flexibility that has previously been unheard of in this space. All of this is delivered without any need for knowledge of scripting languages (or indeed the challenge of trying to read and understand somebody else's scripting work) and is presented to the administrator in the intuitive standard MMC interface. At runtime, the Environment Manager Agent software reads the "rules" from the local registry of the computer protected, delivering the login sequence asynchronously, improving efficiency on the computer during the logon process resulting in quicker logins and potentially fewer user complaints due to performance.

2. <http://www.kixtart.com/>

3. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/scriptinga.asp>

AppSense Environment Manager Overview

AppSense Environment Manager simplifies the configuration, lock-down and maintenance of system and user environments.

Providing a centralized and intuitive interface AppSense Environment Manager delivers a single, effective solution for creating and maintaining your IT infrastructure, enabling Administrators to streamline previously complex and labor intensive system administration tasks and concentrate resources on more critical projects.

By defining computer and user policies, Administrators are able to deliver tailored server and desktop environments no matter where users log on from. Both computer and user policies are enforced using the AppSense self-healing mechanism, which prevents users from making critical changes to server or desktop settings.

A comprehensive set of rules allows for the most complex of enterprise environments to be managed with a single configuration, whether desktop or server.

AppSense Environment Manager is not tied to specific application or operating system versions, so there is no need to deploy a different management solution to target specific environments. Where necessary, the software will automatically discover the environment that it is running within, such as desktop, server or terminal services, and apply the specific and appropriate policy settings. Administrators can dynamically remove unwanted application features to simplify the end user experience and ensure potential security loopholes are removed – especially where remote or mobile users are concerned. Flexible rules and conditions may also be placed on individual actions within a computer or user policy to provide granular-level control of user, session, client, computer and Active Directory settings.

Highlights include:

Simplified Profile Management

AppSense Environment Manager can be used to resolve roaming profile issues that may be encountered when users are logged on to multiple servers. By using a mandatory profile, AppSense Environment Manager may be configured to save different portions of the user's profile at logout, such as registry settings and files, and then restore them when the user next logs on. This has the added benefit of minimizing network bandwidth consumption, saving and loading relevant areas of a user's profile, rather than transferring the whole profile across the network.

Self-healing

AppSense Environment Manager's self-healing technology automatically protects and repairs essential elements of the system and users' environment. For instance, if a user deletes important configuration settings in the system registry or removes any vital files,

AppSense Environment Manager can be configured to automatically detect and correct these problems. This self-healing functionality leads to a reduction in help-desk calls and administrative support, by resolving many of the common configuration issues that arise once a system is deployed into a production environment.

Actions, Rules & Wizards

With AppSense Environment Manager you get a comprehensive set of actions and rules that enable administrators to configure and maintain desktop environments without the need for advanced scripting skills. Configurable actions include the management of files, folders, drives, printers, registry, shortcuts, environment variables, ODBC connections, user interface controls and ADM policies. Rules may also be configured based on Active Directory, user, session, client or computer settings, giving you complete flexibility to apply precise actions in specific scenarios. Inherent, task driven wizards simplify and accelerate the creation of both actions and rules.

Application Lockdown

Administrators are empowered to strip out unwanted functionality from third party software either for security reasons or to reduce the complexity of the end user experience. Lockdown actions are configured using the Lockdown Control Wizard. For example, Administrators can hide or disable user interface controls and block keyboard shortcuts for all, or specific, applications.

Active Directory & Windows Policy Integration

AppSense Environment Manager can either be used alongside Active Directory implementations to extend Group Policy settings or can be used independently. Windows ADM files can be imported and reconfigured, providing centralized management and deployment of existing policies from within the AppSense Environment Manager console.

XML-based Configurations

A rich Microsoft Management Console (MMC) provides centralized configuration and the underlying XML-based configuration replaces the need for synchronous logon scripts, ensuring logon actions are processed in parallel. The result is dramatically reduced user logon times. Dependencies can also be applied to ensure dependent actions run in sequence as necessary.

For more information on AppSense Environment Manager, please visit the AppSense Product Centre at <http://www.appsense.com/environmentmanager>

4. Roaming or Local User Profiles may also be utilized where appropriate

Visit our website for news and support

www.appsense.com

Head Office

AppSense
3200 Daresbury Park
Daresbury Warrington
WA4 4BU United Kingdom

Tel +44 (0)161 216 3200
Fax +44 (0)161 216 3232
Email info@appsense.com

North American Office

500 W Cypress Creek Road
Suite 690
Fort Lauderdale
FL 33309

Tel +1 954 730 7400
Fax +1 954 730 7380
Email us-info@appsense.com

Central European Office

AppSense GmbH
Am Söldnermoos 17
85399 Hallbergmoos
Deutschland

Tel +49 89 607 68530
Fax +49 89 607 68540
Email de-info@appsense.com

Benelux Office

AppSense
Postbus 54
6665 ZG Driel
The Netherlands

Tel +31 (0)611 045 113
Fax +31 (0)848 333 217
Email benelux-info@appsense.com

Australian Office

St Kilda Road Towers
Suite 1022
1 Queen's Road
Melbourne, Victoria 3004

Tel +61 (0)398 637125
Fax +61 (0)395 257091
Email australia-info@appsense.com

AppSense®

The information contained in this document ("the Material") is believed to be accurate at the time of printing, but no representation or warranty is given (express or implied) as to its accuracy, completeness or correctness. Neither AppSense nor the publisher accepts any liability whatsoever for any direct, indirect or consequential loss or damage arising in any way from any use of or reliance placed on this Material for any purpose.

© 2000-2005 APPSENSE LIMITED. ALL RIGHTS RESERVED

AppSense, Security from within, Management made easy and Performance for everyone are registered trademarks of AppSense Ltd. All other brands or product names are trademarks or registered trademarks of their respective companies.

