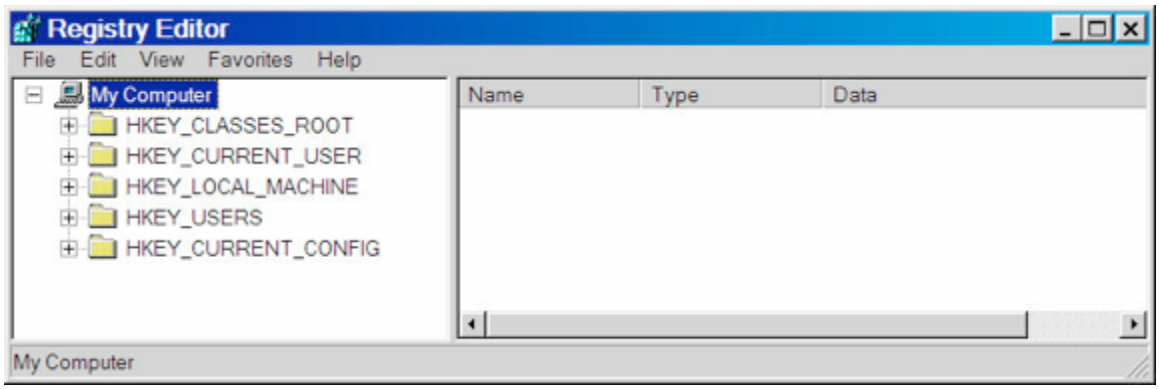
triCerat's Simplify Profiles: Roaming Environments *Without* Roaming Profiles

Part of the attraction of server-based computing is the promise of giving users the same computing experience from any device they log into. However, accomplishing this reality can be more difficult than it may appear at first. The problem lies in the way that NT-based operating systems (including not only NT4 but also Windows 2000 and Windows Server 2003) store and apply per-user settings. The following pages will explain some of the difficulties of tuning user settings for terminal server users and explain how triCerat's Simplify Profiles can alleviate these difficulties.

How NT Operating Systems Store Configuration Information

Because understanding the way that NT operating systems store and apply configuration settings is key to understanding how Simplify Profiles works, let's review a little background. As you probably know, NT-based operating systems store user and machine settings in a flat database called the Registry. You can edit the Registry either from a tool called the Registry Editor or through any of the many graphical interfaces to machine and user configuration information--the applets in the Control Panel or Administrative Tools. You can also edit the Registry through an application's Properties, since modern applications (that is, those written for NT operating systems) store their configuration information in the Registry. The Registry is organized into five main sections--*keys*--but the two that really matter are HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER--all other keys are duplicates of the settings in HKLM and HKCU. HKLM contains all machine-wide settings. HKCU is created from a sort of template in HKEY_USERS and populated with the settings applied to the person currently logged into the computer. (More accurately, it's the settings applied to the security ID (SID) associated with the user account used to log in. Ergo, if more than one person is using the same user account, they get the same settings in HKCU.) If more than one person is logged into the computer--for example, for terminal servers and Windows 2000 and later servers with Remote Administration connections to them--then the Registry will contain one copy of HKCU for each person, although each person will only be able to see their copy of HKCU in their session. If a setting appears in both HKLM and HKCU, the settings in HKCU take precedence.



The information visible in HKLM comes from information gathered during system boot (for example, all system hardware is inventoried each time the computer reboots) or set in the operating system (such as the preferred DNS server for the computer). Computer-wide settings can also be applied through system policies, used in NT domains, or through group policies, used in Active Directory domains.

HKCU's information is pulled from the *user profile* for the person currently logged in and (optionally) from system policies or from group policies. The user profile consists of per-user Registry settings (stored in a file called NTUSER.DAT, with one copy of NTUSER.DAT associated with each user) and files associated with that user: application temporary files or saved files stored in My Documents. If a user is logging onto a computer or domain for the first time, they'll use the default version of NTUSER.DAT assigned to Default User, which will then be copied to their user profile directory and thereafter loaded whenever they log on. To start users with a specific set of settings, therefore, you can configure a profile and copy it to the Default Users section. By default, users can edit any information in their profile--registry settings or files. When they log off, the edits are saved to NTUSER.DAT.

You can control where user profiles are stored. By default, they're stored locally on each computer (*local* profiles), but you can edit user account properties to point to a network location (*roaming* profiles). In fact, not only can you use roaming profiles, but anyone supporting users in a server-based computing environment is probably using them. First of all, the profile suitable to a person working in a fat-client environment often isn't suitable to working in a shared environment, so you'll need to define a separate profile path on the Profile tab in the user account properties. Second, you don't want to store user profiles locally on terminal servers. Doing so would mean maintaining a copy of each user's profile on each terminal server, making it likely--almost certain, really, if you let users edit their environments at all--that the profiles will get out of synch with each other. In other words, using local profiles means that a user's settings depend on which server her session's on, and also waste storage space on the servers. (You also don't want to lose user settings if you must rebuild a terminal server.) For users to get a consistent work environment when using multiple terminal servers, you pretty much *must* store their profiles on the network, not locally.

When a user logs onto the domain or onto a single server through a terminal server session, the following steps take place to load their preferences:

1. The GINA (Graphical Identification and Authentication--it's a DLL containing logon and authentication code) initializes the network providers and authenticates the user.
2. Once the user is authenticated, the GINA initializes a structure containing pointers to the policy and profile information.
3. The GINA starts the Winlogon process.
4. Winlogon creates the terminal session and calls Userinit, which sets up the user's environment. During this process, it restores network connections, loads profile settings such as fonts and screen colors, and runs any logon scripts. After it's finished loading the profile information, Userinit exits and the shell programs (for example, explorer.exe) inherit the environment that Userinit set up.

The Problem with Profiles

That's the short version of how profiles work. Here's why they don't always work very well.

The GINA can load profiles either from the local computer or from a network share, according to the settings in the user's Properties sheet. However, both choices have their problems. Local profiles are a bad idea any time that people will be logging into more than one computer--a scenario very likely in all but the smallest of server-based computing environments. If you're trying to present a consistent user environment to the people using the server farm, roaming profiles appear to be the best option, since they don't take up disk space on the servers in the farm and will be consistent regardless of which server a user logs into.

Using roaming profiles in this situation introduces its own set of problems: lost edits, slow logons, possible profile corruption, and lack of administrator control over user settings.

Lost changes. If a user edits the profile, then their changes are cached until they log off, at which time the changes are written to NTUSER.DAT. The changes are not made on an ongoing basis. Therefore, if someone has more than one copy of her profile open at a time--is logged onto the domain more than once--and edits her user environment settings in both sessions, then only the edits made in the last session she logs off will be saved. More precisely, both sets of edits will get written to NTUSER.DAT, but the last version of the profile will overwrite any other versions.

Slow user logons/logoffs. Roaming profiles are stored in their network location, but when open they're cached on the computer where their owner is logged in. But to get to that computer, they need to cross the network. Recall that user profiles have two pieces: the Registry settings and the files associated with that user-temporary files (such as those generated by a browser) and saved files. By default, all these user-specific files are stored with the profile, which means that profiles can get pretty big. The larger the profile, the longer it takes to load--we're talking minutes here, or even tens of minutes with a very large profile. Similarly, writing a large profile back to its home on the network can take a

long time if available bandwidth is short, or won't work at all if you run out of room on the disk where they're stored.

Profile Corruption and Non-Loading User Settings. Profile corruption is a real possibility in a multi-user environment. The contents of each user's HKCU are stored in a region of memory reserved for this purpose. This region has a size limit. For that matter, the *Registry* itself is only allowed to be so big--this is a setting in the operating system. All separate instances of HKCU are part of the same Registry. If memory is so constrained that an addition to HKCU can't be written to the necessary area of memory, then the profile in memory can get corrupted. If the corrupted version gets written back to the user's NTUSER.DAT, then the saved profile is corrupted. When the user next logs on, they won't be able to load their profile and will instead get the Default User profile--a situation not likely to endear you your user base, since they'll have lost all their customizations. If this area of memory is already full when someone logs on, it's even possible that that user could suddenly have Administrator privileges when they logged back on--the OS's helpful way of letting them increase the size of the Registry.

Lack of control--*your* control. Once a user has a profile created from the Default User profile, you lose control over it. For example, if your default user environment includes wallpaper with your company logo on it, every person who started with that default profile will see the company logo…until they decide that they like their personal wallpaper better. If their personal wallpaper is something innocuous like spaceships, this is a mild annoyance. But let's just say that not everyone favors spaceships for their desktop wallpaper, and some choices, seen by the wrong people, could get you or the user in a lot of trouble. If policies are an option--we'll talk about why they might not be an option in a minute--then you can control most desktop settings with policies.

But policies *don't* apply to applications, not most applications, anyway. When you install applications onto a terminal server in Install Mode (using either Add/Remove Programs or the change user command-line utility) then all settings pertinent to those applications are stored in HKLM\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install. Any edits that an application makes to HKCU or HKLM are copied to HKLM\Software\ Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Machine. When the session is in Execute mode, if an application attempts to read an HKCU Registry entry that doesn't exist, Terminal Services will look in HKLM\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install for the missing key. If the key is there, Terminal Services will copy it and its subkeys to the appropriate location under HKCU, and copy any INI files or user-specific DLLs to the user's home directory (or to their profile directory, if they don't have a home directory). The key point here is that that the OS checks to see only whether the keys *exist* in HKCU, not whether the keys in HKCU are *different* from the ones in HKLM. Therefore, if you reconfigure an application after people have started using it and that change results in different registry values then those change won't get propagated to the people who aren't first-time users.

So what are the options, if both roaming and local profiles have their troubles when combined with Terminal Services?

Locking Down the User Environment With Mandatory Profiles

One way to avoid the problem of lost edits or corrupted profiles to use *mandatory* profiles, which are just roaming profiles with a .MAN extension instead of .DAT. (Mandatory profiles could be local, but they're generally roaming profiles.) Although users can edit the cached version of their mandatory profile while logged in, those changes don't get written to their user profile--NTUSER.MAN is read-only. Using mandatory profiles speeds up user logoffs (since no changes need to be written to the profile file) and avoids consistency problems. However, it provides these advantages by giving the user no control over the user environment--an approach that can make people who have full control over their home computers cranky. One fundamental problem with profiles is that, when it comes to user control, they're an all-or-nothing proposition. You cannot make one part of a profile mandatory and leave other parts open to user customization.

What about using policies to define parts of the user environment? Let's take a look at that option.

Customize the User Environment With Policies

Polices are one way to get more granular control over the user environment, but they work only to a certain degree and can be complex to apply. System policies (in a SAM domain) use one format, group policies (in Active Directory domains) another. Group policies may be applied to organizational units, sites, and domains, while system policies may be applied to users and groups. In other words, although you can import .ADM files into the Group Policy Object Editor, you'll still need to apply them again to take the new domain structure into account. For that matter, all policies require a domain security structure to apply to many servers at once. Standalone server farms not using a domain structure--as some terminal servers are--can't use domain policies at all, but only local policies.

You also can't control everything--or everybody--with policies when working in a server-based environment, but only those settings exposed through policies. Not all user settings are defined through the standard policies that come with the OS. You can configure general desktop settings, including control over the contents of the Control Panel, and Windows Server 2003 adds more system-setting policies to the mix. You can also configure some settings for applications included with Windows, such as Internet Explorer, and any settings controlled with group policies will update regularly--not even requiring users to log off in most cases. However, many common applications, including Microsoft Office and Adobe Acrobat, *don't* have policies defined. If you have .ADM files, you can import them to the Administrative Templates section, but that may mean writing an ADM file if you don't already have one.

Additionally, group policies apply to users and to computers, organized into sites, domains and (optionally) organizational units. Assuming that you're defining user environments for terminal sessions, the computers in this case are the *terminal servers*. You cannot define policies according to the computer that a user has initiated their terminal session *from*. Since some user environment settings should be location-specific

(for example, the printer that you should use while logged into the kiosk in a third-floor hospital hallway is probably not the best one to use while working from your office on the ninth floor) this lack can make it difficult to tune user environments appropriately. Loopback policies can make policies specific to a terminal server override any conflicting policies for a user account. However, they can't provide settings specific to the client computer from which the user is initiating the RDP or ICA session. And because you can't apply group policies to groups--calling them "group policies" is simply an attempt to throw off the unwary--you can't use the same policy structure for Active Directory domains and NT4 domains.

To recap, these are the problems we're facing in applying appropriate settings to a user environment:

- Roaming profiles are very exposed to lost changes and profile corruption, but local profiles are a pain to manage across many servers and must be stored on each server.
- Storing profiles locally means backing up terminal servers, and is also a waste of disk space. Hard disk space is a lot cheaper than it used to be, but it's neither free nor inexhaustible.
- Mandatory profiles avoid the difficulties of roaming profiles because users *can't* change them, but restrict the user's control over their workspace.
- Policies can be difficult to configure, don't work in quite the same way across NT4 and Active Directory domains, and don't expose all user environment information.

One solution to these difficulties is triCerat's Simplify Profiles, which can work with your existing user management infrastructure to provide a customized user environment with fewer hassles. Let's look at how this works.

What is Simplify Profiles?

Simplify Profiles cooperates with policies and profiles to tune the contents of HKCU after a user logs in and after his or her profile loads. The contents of HKCU may be controlled either by the Simplify Profiles administrator or by the user--it depends on how you've set it up. You can apply settings to users, groups, organizational units, domains, or to the computers on which sessions are running. Simplify Profiles gives you more granular control than either policies or profiles, permitting simple, reversible Registry editing from an easy-to-understand interface and giving you the ability to create user environments based on criteria not available through other channels. And, by overlaying the existing settings stored in user profiles, it ends the tug-of-war between users and administrators.

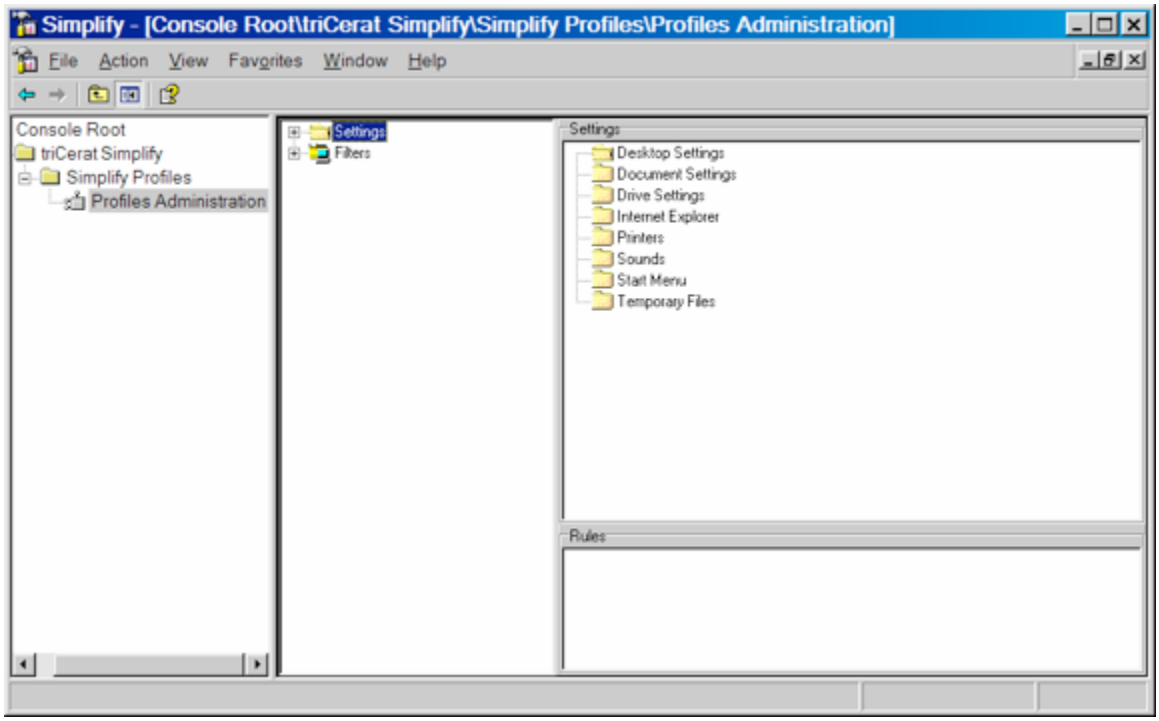
You can use Simplify Profiles to:

- assign printers based on terminal location
- Use mandatory profiles but let users edit application settings if necessary

- Avoid downtime caused by lost user profiles
- Update application settings for applications already in use on a terminal server
- Reduce time dedicated to profile maintenance

Managing User Settings with Simplify Profiles

Simplify Profiles gives you as much control as you like over the contents of the keys in HKCU and how they're applied to the people and machines logging onto your terminal servers. To manage these contents, you'll use the Simplify Profiles Administration tool.



The Settings folder contains some common Registry entries that triCerat pulled out to make it easier to see their purpose and organized into folders. (Some built-in settings, such as the one controlling the default printer, consist of more than one Registry entry.) In addition to the existing folders, you can create folders and put settings into those folders to organize them in a way that makes sense to you. The Filters folder contains a tree view of the users and computers within the currently selected domain that Simplify Profiles can see, and to which you can apply settings.

So what are these settings and filters? In the world of Simplify Profiles, you're combining settings with filters to make rules. The *settings* are the settings in HKCU that you can control. The *filters* are the domains, OUs, groups, users, or machines to which you can apply settings (or entire folders of settings) by dragging them onto a filter. Notice that Simplify Profiles supports both NT4 and Active Directory domains--you can apply settings to OUs or to groups.

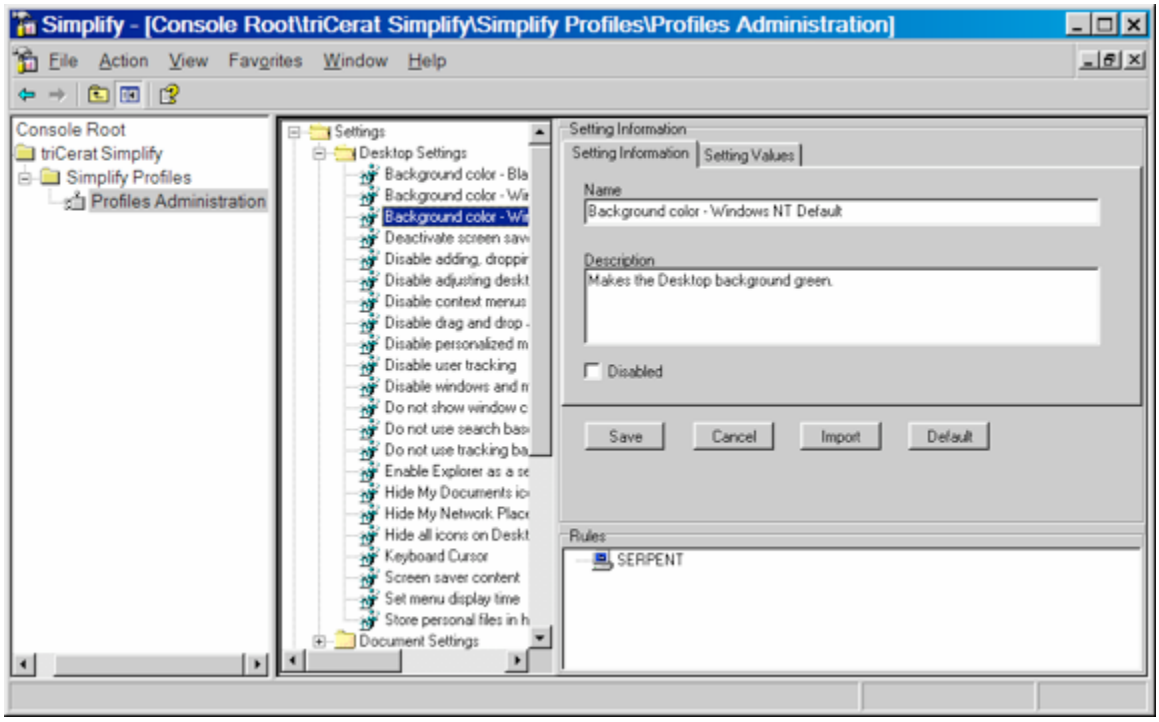
Once you've assigned a setting to a filter, you've created a *rule*. For example, say that you assign the Background Color--Black setting to the Domain Users group. You have

now created a rule saying “Any time that someone in Domain Users logs onto the domain, make their Desktop black.” The Simplify Profiles database consists of these rules, ignoring any HKCU settings not found there. Any settings not controlled by the database use the information in the profile or policy for the current user

Creating Rules

To create a rule from an existing setting, select the Settings folder in the left-hand pane so that the settings are displayed in the right, and then click the Filters folder to select a domain. Drag the setting to the appropriate filter in the selected domain. Simplify Profiles will ask you if you want to create the rule from that setting and that filter. Click Yes, and you’ll create the new rule. You can organize settings by creating folders and putting the settings within those folders. If you do, then you can apply a bunch of settings at once by dragging the entire folder onto the filter.

Once you’ve created a new rule, you can see it by selecting either its filter or setting in the left-hand pane. The right-hand pane will display properties for that filter or setting, including (in the bottom half) a list of the filters/settings that it’s associated with, in the order in which you created the rules.

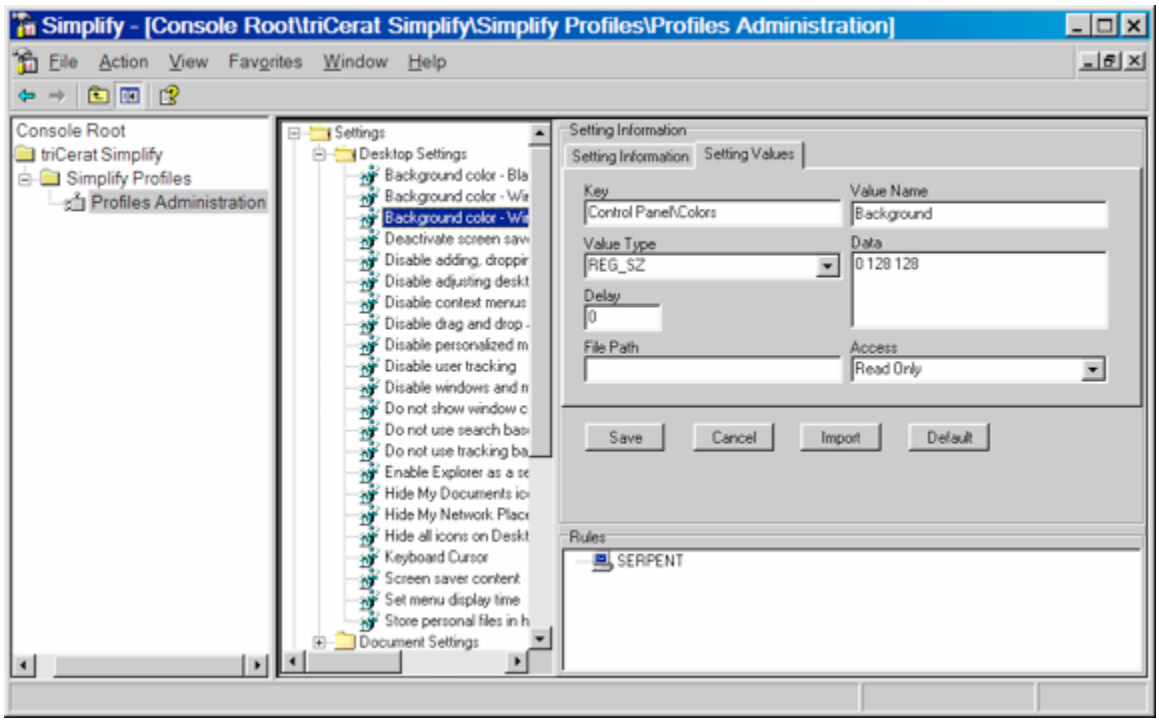


Creating New Settings

You’re not restricted to the settings and values already present in Simplify Profiles. You can add your own settings to the database by changing the value of an existing setting, or importing a single key from the Registry. You can also edit settings to make them Read Only (meaning that user changes are saved only in the database and that users cannot change them), Read/Write (meaning that users can edit their settings and save those

changes to a location accessible to the database) or Delete, which deletes a Registry key in HKCU. A deleted key still exists in the profile, but when the Simplify Profiles rules are applied will no longer be accessible to the session.

To change the value of an existing setting, double-click it to open its Properties sheet, and turn to the Setting Values tab. The part that you need to edit is the Data portion. (Editing any other portions can and probably will make the key stop working altogether, since the Registry expects to find its information labeled in a certain way. Be sure to enter the data in the form expected for that setting--don't put string data in a key prepared for binary data. (Simplify Profiles will not allow you to change the value type.) When you've finished editing the value, click Save to add the edited value to the database. Those changes--including any change to the name of the setting--will be reflected in any rules already using that setting.



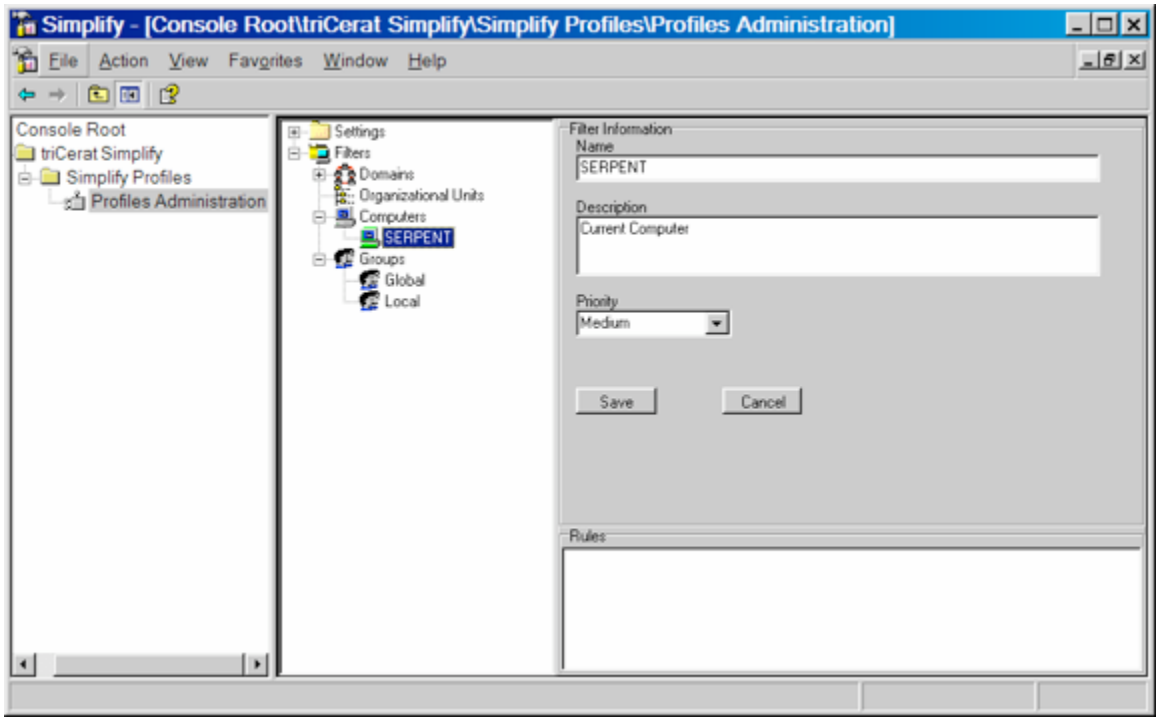
You can also import settings from the Registry. Right-click on the Settings folder and choose Create Setting. This will create a new setting entry with no name, no description, and no value. You can give the new setting a name and description from the Setting information tab. To flesh it out with some data, turn to the Setting Values tab, and click the import button to browse HKCU for the appropriate key. When you've done so, you can edit its value as you saw above. On the Setting Information tab, you can call the new setting anything you like and provide a description to remind yourself or another Simplify Profiles administrator of what the setting controls. When you've finished editing the setting, click Save. (The Windows 2000 Resource Kit has a guide to the Registry that you can use to find important settings in HKCU, and also to see how you should format the values of those settings.)

Settings predefined in Simplify Profiles start with a default value. If you edit a setting and want to restore it to its original value, just click the Default button. This will immediately change the contents of the database to their default setting.

Changing Rule Priority

Two rules could conceivably conflict. For example, say that the Simplify Profiles administrator creates one rule that members of the Domain Users group should have black Desktops, and another rule that people logging on from the terminal in the hallway (using that machine's filter) should have green Desktops. If a member of the Domain Users group logs on from the terminal in the hallway, what color is their Desktop?

The answer to that question depends on the priority that you assign a filter. All filters begin with a priority of Medium. To edit a filter's priority, select it in the left-hand pane to display its properties. If two filters have conflicting rules, then rule associated with the filter with the highest priority always wins.



Deleting Rules

You may know that NT 4's system policies have the problem of "tattooing" the Registry--that, once you apply a policy, to remove it you must explicitly reverse the policy you put in place. Unlike system policies, Simplify Profiles does not tattoo the Registry because it never touches the copy of HKCU stored in the profile. If you remove a rule from the database, then Simplify Profiles will no longer apply that rule, so the setting in the profile will control.

To delete a rule, you'll just select the appropriate setting or filter so that the rule is visible in the right-hand pane. Right-click on the appropriate rule and choose Delete Rule from the context menu. This will delete the rule, but not affect the setting or the filter.

Replicating the Rules Database

The rules database needs to be consistent across all terminal servers. When you close the Simplify Profiles administrator, the tool will prompt you to replicate the changes to the other Simplify Profiles servers, identified by Net BIOS name or IP address. You can also choose to replicate the database at any time. You'll need the right to copy files to any server on which you're replicating the Simplify Profiles database. As the database is small (generally under 1MB even for large domains) the replication process does not take long.

Read/write user settings that the user has edited are stored not in the terminal server database but in a filter-specific database located on a network location you defined when setting up Simplify Profiles.

Protecting Simplify Profiles Settings

Because the copy of the rules database on each server is identical to that on all other servers, you don't need to back it up. If you need to recreate it on a server--or add a new terminal server with Simplify Profiles installed--you need only replicate the database to the server. However, you *do* need to protect the filter-specific database files stored on the network share available to all servers. This directory should be part of your regular backup scheme. However, this directory does not have to be stored on a terminal server, so the user-specific backups can be part of a normal server backup. Rebuilding a terminal server will not affect the filter-specific settings.

How Does Simplify Profiles Work?

That's how you create and manage rules, but how do those rules apply to people using the terminal server?

Simplify Profiles is fundamentally very simple. A user management database maintained by the Simplify Profiles administrator resides on each terminal server running Simplify Profiles. (Why on each server instead of on a network share accessible to all servers? I'll get to that in a minute.) This database contains configuration information in HKCU, associated with users, groups, organizational units, or domains. Each time the Simplify Profiles administrator edits the database and closes the Simplify Profiles administration tool, they're prompted to replicate the database. Per-user settings that users have read/write access to are stored in subfolders of a network share that you specify when installing Simplify Profiles.

When a user logs onto a terminal server using Simplify Profiles, the tool checks to see which filters they're associated with through their user or machine identity, and determines the priority of those filters. It then links to that user's profile and then loads the appropriate rules database information. For any Read/Write settings, the rules

database follows the link to the network share where that filter's editable settings are stored, and loads them as well. In other words, Simplify Profiles doesn't replace the user profile, it just overlays it. Only those settings explicitly defined are included in the Simplify Profiles database, not the entire contents of HKCU.

Alternate Methods of Editing the Registry

Simplify Profiles is not the only available method of editing the Registry. It's just a simple and very reversible way of doing it.

It's possible to edit the Registry by exporting a set of values to a .REG file and then importing those saved values, either manually, automatically when the .REG file is executed, or during a batch file. It's also possible to edit settings using VBScript's RegWrite method, which allows you to edit individual Registry values instead of importing data in chunks. However, both methods have two major failings. First, they require you to find the key you want to edit, a process that can be difficult and always induces eyestrain. Second, *the Registry has no Undo function*. If you edit the Registry manually, that edit is there until you explicitly reverse it. This fact can make testing Registry edits a tedious procedure--and actually dangerous if working in machine-wide keys.

Simplify Profiles does not touch the contents of HKCU in any permanent way. If you remove a rule, the value in HKCU reverts to whatever's defined in the profile. If you don't like the change you made to an existing setting, you can click the Default button to reverse it to the out-of-the-box value. This, combined with Simplify Profiles' intuitive and flexible organizational scheme, makes it much easier and safer to test Registry edits than it is when editing the Registry by hand.

Simplify Profiles and Your Existing Network

If you've read this far, then you know that using Simplify Profiles doesn't require scrapping your entire network and starting over.

Although Simplify Profiles will work with roaming or local profiles, we recommend that you take existing user profiles, configured as their owners want them, and make them mandatory profiles. Having done so, create Read/Write rules making any settings that you don't need to control. Settings that need to stay as they are can be left alone, or you can edit them from Simplify Profiles and make them Read Only.

Need to update application settings after people have been using an application for months? Edit the appropriate settings with Simplify Profiles and create rules to get those settings to the right people.

In short, Simplify Profiles works with your existing network to make customizing the user workspace easier, whether for users, domains, or the computer that a person is using to connect to a terminal server. It works for both Active Directory and NT 4 domains (and for all generations of Windows Terminal Services, from TSE to the not-yet-released
Windows Server 2003) and is more flexible in the way that it can customize user

environments than either policies or profiles. Test the 30-day trial version on your network and see how Simplify Profiles can make user workspace administration easier.