



[TechNet Home](#) > [Sysinternals Home](#) > [Utilities Index](#)

NewSID v4.10

By Mark Russinovich and Bryce Cogswell

Published: November 1, 2006

Introduction

Many organizations use disk image cloning to perform mass rollouts of Windows. This technique involves copying the disks of a fully installed and configured Windows computer onto the disk drives of other computers. These other computers effectively appear to have been through the same install process, and are immediately available for use.

While this method saves hours of work and hassle over other rollout approaches, it has the major problem that every cloned system has an identical Computer Security Identifier (SID). This fact compromises security in Workgroup environments, and removable media security can also be compromised in networks with multiple identical computer SIDs.

Demand from the Windows community has lead several companies to develop programs that can change a computer's SID after a system has been cloned. However, Symantec's SID Changer and Symantec's Ghost Walker are only sold as part of each company's high-end product. Further, they both run from a DOS command prompt (Altiris' changer is similar to *NewSID*).

NewSID is a program we developed that changes a computer's SID. It is free and is a Win32 program, meaning that it can easily be run on systems that have been previously cloned. *NewSID* works on Windows NT 4, Windows 2000, Windows XP and Windows .NET Server.

Please read this entire article before you use this program.

Version Information:

- Version 4.0 introduces support for Windows XP and .NET Server, a wizard-style interface, allows you to specify the SID that you want applied, Registry compaction and also the option to rename a computer (which results in a change of both NetBIOS and DNS names).
- Version 3.02 corrects a bug where NewSid would not correctly copy default values with invalid value types when renaming a key with an old SID to a new SID. NT actually makes use of such invalid values at certain times in the SAM. The symptom of this bug was error messages reporting access denied when account information was updated by an authorized user.
- Version 3.01 adds a work-around for an inaccessible Registry key that is created by Microsoft Transaction Server. Without the work-around *NewSID* would quit prematurely.
- Version 3.0 introduces a SID-sync feature that directs *NewSID* to obtain a SID to apply from another computer.
- Version 2.0 has an automated-mode option, and let's you change the computer name as well.
- Version 1.2 fixes a bug in that was introduced in 1.1 where some file system security descriptors were not updated.
- Version 1.1 corrects a relatively minor bug that affected only certain installations. It also has been updated to change SIDs associated with the permission settings of file and printer shares.

[↑ Top of page](#)

Cloning and Alternate Rollout Methods

One of the most popular ways of performing mass Windows rollouts (typically hundreds of computers) in corporate environments is based on the technique of disk cloning. A system administrator installs the base operating system and add-on software used in the company on a template computer. After configuring the machine for operation in the company network, automated disk or system duplication tools (such as [Symantec's Ghost](#), [PowerQuest's Image Drive](#), and [Altiris' RapiDeploy](#)) are used to copy the template computer's drives onto tens or hundreds of computers. These clones are then given final tweaks, such as the assignment of unique names, and then used by company employees.

Another popular way of rolling out is by using the Microsoft *sysdiff* utility (part of the Windows Resource Kit). This tool requires that the system administrator perform a full install (usually a scripted unattended installation) on each computer, and then *sysdiff* automates the application of add-on software install images.

Because the installation is skipped, and because disk sector copying is more efficient than file copying, a cloned-based rollout can save dozens of hours over a comparable *sysdiff* install. In addition, the system administrator does not have to learn how to use unattended install or *sysdiff*, or create and debug install scripts. This alone saves hours of work.

[↑ Top of page](#)

The SID Duplication Problem

The problem with cloning is that it is only supported by Microsoft in a very limited sense. Microsoft has stated that cloning systems is only supported if it is done before the GUI portion of Windows Setup has been reached. When the install reaches this point the computer is assigned a name and a unique computer SID. If a system is cloned after this step the cloned machines will all have identical computer SIDs. Note that just changing the computer name or adding the computer to a different domain does not change the computer SID. Changing the name or domain only changes the domain SID if the computer was previously associated with a domain.

To understand the problem that cloning can cause, it is first necessary to understand how individual local accounts on a computer are assigned SIDs. The SIDs of local accounts consist of the computer's SID and an appended RID (Relative Identifier). The RID starts at a fixed value, and is increased by one for each account created. This means that the second account on one computer, for example, will be given the same RID as the second account on a clone. The result is that both accounts have the same SID.

Duplicate SIDs aren't an issue in a Domain-based environment since domain accounts have SID's based on the Domain SID. But, according to Microsoft Knowledge Base article Q162001, "Do Not Disk Duplicate Installed Versions of Windows NT", in a Workgroup environment security is based on local account SIDs. Thus, if two computers have users with the same SID, the Workgroup will not be able to distinguish between the users. All resources, including files and Registry keys, that one user has access to, the other will as well.

Another instance where duplicate SIDs can cause problems is where there is removable media formatted with NTFS, and local account security attributes are applied to files and directories. If such a media is moved to a different computer that has the same SID, then local accounts that otherwise would not be able to access the files might be able to if their account IDs happened to match those in the security attributes. This is not possible if computers have different SIDs.

An article Mark has written, entitled "[NT Rollout Options](#)", was published in the June issue of Windows NT Magazine. It discusses the duplicate SID issue in more detail, and presents Microsoft's official stance on cloning. To see if you have a duplicate SID issue on your network, use [PsGetSid](#) to display machine SIDs.

[↑ Top of page](#)

NewSID

NewSID is a program we developed to change a computer's SID. It first generates a random SID for the computer, and proceeds to update instances of the existing computer SID it finds in the Registry and in file security descriptors, replacing occurrences with the new SID. *NewSID* requires administrative privileges to run. It has two functions:

changing the SID, and changing the computer name.

To use *NewSID*'s auto-run option, specify `"/a"` on the command line. You can also direct it to automatically change the computer's name by including the new name after the `"/a"` switch. For example:

newsid /a [newname]

Would have *NewSID* run without prompting, change the computer name to "newname" and have it reboot the computer if everything goes okay.

Note: If the system on which you wish to run *NewSID* is running IISAdmin you must stop the IISAdmin service before running *NewSID*. Use this command to stop the IISAdmin service: `net stop iisadmin /y`

NewSID's SID-synchronizing feature that allows you to specify that, instead of randomly generating one, the new SID should be obtained from a different computer. This functionality makes it possible to move a Backup Domain Controller (BDC) to a new Domain, since a BDC's relationship to a Domain is identified by it having the same computer SID as the other Domain Controllers (DCs). Simply choose the "Synchronize SID" button and enter the target computer's name. You must have permissions to change the security settings of the target computer's Registry keys, which typically means that you must be logged in as a domain administrator to use this feature.

Note that when you run *NewSID* that the size of the Registry will grow, so make sure that the maximum Registry size will accommodate growth. We have found that this growth has no perceptible impact on system performance. The reason the Registry grows is that it becomes fragmented as temporary security settings are applied by *NewSID*. When the settings are removed the Registry is not compacted.

Important: Note that while we have thoroughly tested *NewSID*, you must use it at your own risk. As with any software that changes file and Registry settings, it is highly recommended that you completely back-up your computer before running *NewSID*.

[↑ Top of page](#)

Moving a BDC

Here are the steps you should follow when you want to move a BDC from one domain to another:

1. Boot up the BDC you want to move and log in. Use *NewSID* to synchronize the SID of the BDC with the PDC of the domain to which you wish to move the BDC.
2. Reboot the system for which you changed the SID (the BDC). Since the domain the BDC is now associated with already has an active PDC, it will boot as a BDC in its new domain.
3. The BDC will show up as a workstation in Server Manager, so use the "Add to Domain" button to add the BDC to its new domain. Be sure to specify the BDC radio button when adding.

[↑ Top of page](#)

How it Works

NewSID starts by reading the existing computer SID. A computer's SID is stored in the Registry's **SECURITY** hive under **SECURITY\SAM\Domains\Account**. This key has a value named F and a value named V. The V value is a binary value that has the computer SID embedded within it at the end of its data. *NewSID* ensures that this SID is in a standard format (3 32-bit subauthorities preceded by three 32-bit authority fields).

Next, *NewSID* generates a new random SID for the computer. *NewSID*'s generation takes great pains to create a truly random 96-bit value, which replaces the 96-bits of the 3 subauthority values that make up a computer SID.

Three phases to the computer SID replacement follow. In the first phase, the **SECURITY** and **SAM** Registry hives are scanned for occurrences of the old computer SID in key values, as well as the names of the keys. When the SID is found in a value it is replaced with the new computer SID, and when the SID is found in a name, the key and its subkeys are copied to a new subkey that has the same name except with the new SID replacing the old.

The final two phases involve updating security descriptors. Registry keys and NTFS files have security associated with them. Security descriptors consist of an entry that identifies which account owns the resource, which group is the primary group owner, an optional list of entries that specify actions permitted by users or groups (known as the Discretionary Access Control List - DACL), and an optional list of entries that specify which actions performed by certain users or groups will generate entries in the system Event Log (System Access Control List - SACL). A user or a group is identified in these security descriptors with their SIDs, and as I stated earlier, local user accounts (other than the built-in accounts such as Administrator, Guest, and so on) have their SIDs made up of the computer SID plus a RID.

The first part of security descriptor updates occurs on all NTFS file system files on the computer. Every security descriptor is scanned for occurrences of the computer SID. When *NewSID* finds one, it replaces it with the new computer SID.

The second part of security descriptor updates is performed on the Registry. First, *NewSID* must make sure that it scans all hives, not just those that are loaded. Every user account has a Registry hive that is loaded as **HKEY_CURRENT_USER** when the user is logged in, but remains on disk in the user's profile directory when they are not. *NewSID* identifies the locations of all user hive locations by enumerating the **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\ProfileList** key, which points at the directories in which they are stored. It then loads them into the Registry using RegLoadKey under **HKEY_LOCAL_MACHINE** and scans the entire Registry, examining each security descriptor in search of the old computer SID. Updates are performed the same as for files, and when its done *NewSID* unloads the user hives it loaded. As a final step *NewSID* scans the **HKEY_USERS** key, which contains the hive of the currently logged-in user as well as the .Default hive. This is necessary because a hive can't be loaded twice, so the logged-in user hive won't be loaded into **HKEY_LOCAL_MACHINE** when *NewSID* is loading other user hives.

Finally, *NewSID* must update the **ProfileList** subkeys to refer to the new account SIDs. This step is necessary to have Windows NT correctly associate profiles with the user accounts after the account SIDs are changed to reflect the new computer SID.

NewSID ensures that it can access and modify every file and Registry key in the system by giving itself the following privileges: System, Backup, Restore and Take Ownership.

[↑ Top of page](#)

Using the Source

Full source code to *NewSID* has been provided for educational purposes. You may not use this code in a commercial or freeware SID-changing product, but you may use its techniques in other programs for private or commercial use.



[Download NewSID \(68 KB\)](#)

[↑ Top of page](#)

[Manage Your Profile](#)

© 2006 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

Microsoft