ClonePrincipal User Guide

Document version 2.1
November 9, 1999

Customers deploying Microsoft® Windows® 2000 might want to migrate users and resources incrementally to a new Windows 2000 environment in order to minimize disruption to their existing Windows NT® version 4.0 production environment. ClonePrincipal supports migrating users and resources from Windows NT to Windows 2000 by creating clones of Windows NT 4.0 users and groups in the new Windows 2000 environment. Once the users and resources have been migrated successfully to the new environment, the original source accounts are no longer needed and should be deleted.

Note: ClonePrincipal should not be used in a production environment without an understanding of the security and administrative implications described in the white paper "Planning Migration from Microsoft Windows NT to Microsoft Windows 2000," which has the file name Dommig.doc. Dommig.doc is included in the Windows 2000 Support Tools, which are included on the Windows 2000 CD.

The benefits achieved using ClonePrincipal in the scenarios described above are:

- Users may log on to destination account (clone), yet have emergency fallback to the source account during a trial period.
- Users may be introduced a few at a time to the Windows 2000 destination environment.
- Source production environment is not disrupted during migration of users to destination Windows 2000 environment.
- It is not necessary to update group memberships and Access Control Lists (ACLs) in order to preserve network access for destination accounts.
- Multiple groups from different source domains may be merged into the same destination group.
- Windows NT 4.0 Backup Domain Controllers (BDCs) acting as resource servers can be upgraded, then demoted to a member servers and moved to a Windows 2000 domain without redefining local groups or updating the ACLs on that server.

A "clone" is an account in a native mode Windows 2000 domain, for which Windows NT 4.0 account properties and group memberships have been copied from a source account. Although the clone necessarily has a different primary security identifier (SID) than the source account, the SID of the source account is copied into the clone's sIDHistory attribute. Populating the sIDHistory attribute with the SID of a source account allows the clone to access all network resources available to the source account, providing trusts exist from the resource domains to the clone's account domain.

Network access is preserved via sIDHistory because in a native mode Windows 2000 domain, a user interactive logon creates an access token containing not only the user's primary SID and global group SIDs, but also the user's sIDHistory and group sIDHistory values. The SIDHistories grant the user access to resources protected by local groups and ACLs containing the pre-migration source user and group SIDs. Additionally, user authentication in a resource domain will add local group SIDs and SIDHistories to the user's access token, enabling access to migrated resources protected by ACLs containing pre-migration local group SIDs.

Contents

ClonePrincipal Files and Versions

The ClonePrincipal tool includes a COM object and sample scripts that use that COM object. ClonePrincipal includes the following files:

clonepr.dll: Library that contains the **DSUtils.ClonePrincipal** COM object, implementing the ICloneSecurityPrincipal interface which supports three methods:

1 **Connect:** establish authenticated connections to the source and destination domain controllers.
2 **AddsIDHistory:** copy the security identifier (SID) of a source principal to the sIDHistory of an existing destination principal in a different forest, by calling the DsAddSidhistory API.
3 **CopyDownlevelUserProperties:** copy the Windows NT 4.0 properties of the source user to the destination user.

sidhist.vbs: Sample script that copies the SID of a source principal to the sIDHistory of an existing destination principal in a different forest.
clonepr.vbs: Sample script that clones a single security principal. It creates the destination principal (user, global group, domain local group) if it doesn't already exist, copies the properties of the source principal to the destination principal, then copies the source SID to the sIDHistory of the destination principal. When cloning a global group or user, it establishes group memberships in the destination domain to reflect the memberships in the source domain. When cloning a local group, it copies the entire source membership list to the destination local group.
clonegg.vbs: Sample script that clones all global groups in a domain.
cloneggu.vbs: Sample script that clones all global groups and users in a domain.
clonelg.vbs: Sample script that clones all "shared" or domain local groups in a domain.

To determine the version of clonepr.dll, in Windows Explorer, right click on the clonepr.dll file, select **Properties**, then click on the **Version** tab. To determine the timestamp of the sample scripts, run **cscript <*script.vbs*>**. This displays the command line usage followed by a time stamp, which serves as the version of the script. Note that clonepr.dll must be registered (registration is performed during Support Tools installation) before the command line usage and time stamp can be displayed in this manner.

ClonePrincipal Terminology

This section defines the terminology used in subsequent descriptions of ClonePrincipal operations.

Account Principal. A user, global group, or universal group. Account principals are the security principals, representing users or groups of users, that make up an Account Domain.
Resource Principal. A local group. Resource principals are the security principals defined in a resource domain, used to identify which account principals have access to resources.

RID. Relative Identifier. The "second half" of a SID that uniquely identifies a security principal within a particular domain. The "first half" or "domain part" of a SID identifies the domain in which the security principal resides.
SID. The Security Identifier of a security principal (for example, user or group account). Account SIDs, placed into a user's access token at logon, are used to grant to the user access to network resources. A SID is composed of two parts: the "domain part," which identifies the domain in which the security principal resides, and the relative identifier (RID), which uniquely identifies the principal within that domain.
sIDHistory. An attribute, available in Windows 2000 native mode only, on a security principal (the destination account), that may be populated with the SID of a source account in order to grant to the destination principal access to network resources available to the source principal. The source and destination accounts must be in different forests since a SID (whether a primary SID or a sIDHistory value) must be unique within a forest.
Source Domain. The Windows NT 4.0 or Windows 2000 domain containing the original security principals to be cloned.
Source Object or Source Principal. The security principal object to be cloned.
Destination Domain. The Windows 2000 native mode domain in which the cloned principal account is created.
***Destination Object or Destination Principal* (*Clone*)**. The security principal, in the destination domain, whose attributes have been cloned from a source object and whose sIDHistory contains the primary SID of the source object.
Well known RID. A well known RID identifies an account, such as Domain Admins, whose SID contains an identical RID in every domain, but contains a different "domain part" in each domain. An account with a well known RID can be cloned only onto a destination account with the same well known RID. Accounts with well known RIDs include the following users:

1 Administrator
2 Guest

and the following global groups:

1 Domain Admins
2 Domain Guests
3 Domain Users

Well known SID. A well known SID identifies an account, such as Administrators, whose SID is identical in every domain. An account with a well known SID cannot be cloned, nor would it be useful to clone an account whose SID already exists in the destination. Accounts with well known SIDs include the following local groups:

1 Account Operators
2 Administrators
3 Backup Operators
4 Guests
5 Power Users
6 Print Operators
7 Replicator
8 Server Operators
9 Users

Sample Scripts Functional Description

This section describes the behavior of each sample script along with some tips for usage in special cases. These scripts are idempotent, that is, they may be called multiple times with the same input parameters, leaving the system in the same end state, and reporting no error for redundant calls. This behavior makes it easier to rerun the scripts in case a temporary, or subsequently remediated error condition, prevents successful completion on the first run.

ClonePrincipal sample scripts perform fundamental operations useful in migration scenarios. They provide example usage of the ClonePrincipal COM object, but will not necessarily be suitable for use in every environment without some customization, for example to handle unique configurations, to support partial migrations, or to provide additional error handling.

Sidhist.vbs

This script copies the SID of a source principal to the sIDHistory of an existing destination principal in a different forest from the source principal. If the source domain is Windows 2000 native mode, then any SIDs in the source account sIDHistory attribute are copied to the destination account sIDHistory. Sidhist.vbs calls the AddSidHistory method (see ICloneSecurityPrincipal Programmer's Reference below) to perform this function.

Supported Account Types: The source and destination principals must be one of the following types:

1. User
2. Security enabled Group, including:
   "Shared" Local Group (local group defined once and shared by all domain controllers in an Windows NT 4.0 or Windows 2000 mixed mode domain),
   Domain Local Group (Windows 2000 native mode only),
   Global Group,
   Universal Group (Windows 2000 native mode only).

The following types of accounts are not supported:

1. Computer (workstation or domain controller)
2. Inter-domain trust
3. Temporary duplicate account (This is a virtually unused feature, a legacy of LAN Manager. These account types are established by creating a user and setting the account type to local.)
4. Accounts with well known SIDs

Matching Account Types: The account types of the source principal and the destination principal must match.

1. If the source principal is a User, the destination principal must be a User.
2. If the source principal is a "Shared" Local Group or a Domain Local Group, the destination principal must be a Domain Local Group.

3 If the source principal is a Global Group or Universal Group, the destination principal must be a Global or Universal Group.

Duplicate SIDs: The source account SID must not already exist in the destination forest, either as a primary SID or in the sIDHistory of an account. The exception to this is that when attempting to add a SID to a sIDHistory that already contains that SID, sidhist.vbs will not add the duplicate SID and will return without error. This behavior allows sidhist.vbs to be run multiple times successfully with identical input, resulting in the same end state, for developer and user ease-of-use.

Well known RIDs and SIDs: If the source object has a well known RID, the destination object must have the same well known RID. For example, the SID of REDMOND\Domain Users may be copied into the sIDHistory of NORTHAMERICA\Domain Users, but not NORTHAMERICA\Domain Guests. Neither the source, nor destination object might have a well known SID. Since well known SIDs are constants that already exist in every domain, there is no need to import a well known SID into a domain using sidhist.vbs.

Multi-valued sIDHistory: Sidhist.vbs may be used to combine the access privileges of multiple source accounts in a single destination account. By calling sidhist.vbs multiple times with different source accounts and the same destination account, the destination account acquires multiple values in its sIDHistory attribute. This destination account is granted access to resources accessible to any of the source accounts. Although the sIDHistory attribute is multi-valued and does not have a hard limit on the number of SIDs it may contain, its practical limit is based on the total number (1023) of group SIDs and sIDHistories that may be contained in a user's access token.

Clonepr.vbs

This script clones a single security principal. It creates the destination account (if it doesn't exist already), copies the source account properties (Windows NT 4.0 properties only) into the destination account, and adds the SID of the source account to the sIDHistory of the destination account. Windows 2000 account attributes (that do not exist in Windows NT 4.0) are not copied, even if the source domain is Windows 2000. If the source domain is Windows 2000 native mode, then any SIDs in the source account sIDHistory attribute are copied to the destination account sIDHistory. Because clonepr.vbs calls the same AddSidHistory method used by sidhist.vbs, the source and destination principals are subject to the same restrictions described above for sidhist.vbs. For users, global groups, and universal groups, clonepr.vbs attempts to update the Members and Membership relationships of the destination account to reflect those of the source account. For local groups, clonepr.vbs copies the entire membership list to the destination local group. Additional detail about how clonepr.vbs handles each object type is provided below.

Renaming Accounts: It is possible to specify a SAM name for the destination account that is different from the SAM name of the source account. This is useful when multiple source domains are being consolidated into a single destination domain and accounts

with duplicate SAM names, but representing different users, exist in the source domains. Since clonepr.vbs will not complain if the destination SAM name identifies a pre-existing account (see Pre-existing Destination Account below), take precautions against accidently cloning onto a pre-existing destination account with the same SAM name:

1 Clone accounts from a given source domain into an organizational unit reserved for accounts from that source domain. This ensures that clonepr.vbs will fail if a duplicate SAM name exists for an object in a different organizational unit in the destination domain. Or,
2 Modify the clonepr.vbs script to report, prompt for input, or fail when a duplicate SAM name is found in the destination domain.

Pre-existing Destination Account: If the destination SAM name and destination distinguished name identify an object in the destination domain, the destination account is considered to be "pre-existing." If the destination SAM name and destination DN identify two different objects in the destination domain, then clonepr.vbs returns an error. When clonepr.vbs is called with a pre-existing destination account, then the Windows NT 4.0 properties of the destination account are overwritten with the properties of the source account, and the source account SID (and sIDHistory, for Windows 2000 native mode source accounts) are appended to the destination sIDHistory.

Merging Multiple Source Accounts: Multiple source accounts may be cloned onto the same destination account as a way to "merge" the source accounts. This can be useful when a single individual has a user account in multiple domains, but should possess a single account in the destination forest that gives him access to all resources available to his multiple source accounts. Likewise, if a global group had been created for the same purpose in multiple source domains, it may be desirable to merge these global groups onto a single destination global group. The membership of the merged destination group will reflect the combined membership of all of the source groups. This combined membership will be granted access to the combined resources available to the source groups.

Cloning a User: When a user is cloned, the following Windows NT 4.0 properties are copied from the source account to the destination account (using the ICloneSecurityPrincipal::CopyDownlevelUserProperties method), overwriting existing properties on the destination account:

1 General Properties
    Full Name
    Description
    Password/Account Flags (with exceptions noted below)
2 Profile Properties
    User Profile: Profile path; Logon script
    Home directory: Local path; Connect drive and location
3 Dial-in Properties
    RAS access (allow/deny/control through policy)
    Verify Caller ID
    Callback Options (No callback/set by caller/callback to number)

Static IP address
Static Routes (table of destination:network mask:hops)

4 File and Print for Netware Properties
5 Terminal Server Properties
6 Other third-party Application Properties that are returned in User Parameters

The following properties are explicitly set on the destination user:

1 "Account is disabled" flag is selected
2 "User must change password at next logon" flag is selected
3 "User cannot change password" flag is not selected
4 "Password never expires" flag is not selected
5 Password is set to NULL

User properties unique to Windows 2000 are not copied by ClonePrincipal, even if the source domain is Windows 2000.

The sIDHistory attribute of the destination user is updated with the SID (and sIDHistory in the case of a Windows 2000 native mode source domain) of the source account. This means that the destination user is granted access to all of the resources available to the source user, providing that trusts exist from the resource domains to the destination domain.

ClonePrincipal does not preserve the primary group of user accounts. Destination user accounts will, by default, have a primary group of Domain Users.

The global and universal group memberships of the source user are recreated in the destination domain if possible. When a source user is cloned, the destination user is made a member of each global group in the destination domain and each universal group in the destination forest that is a clone of (that is, has the sIDHistory of) a source group in which the source user is a member. Clonepr.vbs achieves this by creating a list of SIDs representing the source global and universal groups in which the source user is a member, then identifying each destination group that contains one of those SIDs in its sIDHistory. If the source user is a member of a source group that does not have a clone in the destination domain, then this membership relationship obviously cannot be restored by clonepr.vbs in the destination domain. A subsequent clone of the source group, however, will restore this membership.

Clonepr.vbs does not make the destination user a member of any local groups, nor is this required to preserve network access for the destination user. Local groups, in which the source user is a member, implicitly grant access to the destination user because the destination user carries its sIDHistory (the SID of the source user) in its access token.

Cloning a Global or Universal Group: When a global group or universal group is cloned, the Description property is copied from the source account to the destination account, overwriting the Description on the destination account. A global group may be

cloned onto a pre-existing destination global or universal group. Likewise, a universal group may be cloned onto a pre-existing global or universal group.

The sIDHistory attribute of the destination group is updated with the SID (and sIDHistory in the case of a Windows 2000 native mode source domain) of the source account. This means that the members of the destination group are granted access to all of the resources made available via membership in the source group, providing that trusts exist from the applicable resource domains to the destination domain.

Clonepr.vbs updates the membership of the destination group to reflect the membership of the source group if possible. When a source global or universal group is cloned, the destination group membership is updated to contain all the destination users whose corresponding source users are members of the source group. To restore memberships in this fashion, clonepr.vbs creates a list of SIDs representing the membership of the source group, then identifies each destination member that contains one of those source SIDs in its sIDHistory attribute. If the source group contains members that have not been cloned into the destination domain, then this membership relationship cannot be restored by clonepr.vbs. However, a subsequent clone of the member into the destination domain will restore the membership.

For Windows 2000 native mode source domains only, if the source group was a nested member of global or universal group (lets call this the "parent" group) in the source domain, these memberships are recreated in the destination domain if possible. When a source group is cloned, the destination group is made a member of each global group in the destination domain and each universal group in the destination forest that is a clone of (that is, has the sIDHistory of) the source "parent" group in which the source group is a member. Clonepr.vbs achieves this by creating a list of SIDs representing the source "parent" groups in which the source group is a member, then identifying each destination "parent" group that contains one of those SIDs in its sIDHistory. If the source group is a member of a source "parent" group that does not have a clone in the destination domain, then this membership relationship obviously cannot be restored by clonepr.vbs in the destination domain. A subsequent clone of the source "parent" group, however, will restore this membership.

Clonepr.vbs does not make the destination group a member of any local groups, nor is this required to preserve network access for members of the destination group. Local groups, in which the source group is a member, implicitly grant access to the destination group members because the destination members carry the destination group sIDHistory (the SID of the source group) in their access tokens.

Cloning a "Shared" Local Group or Domain Local Group: When a "shared" or domain local group is cloned, the Description property is copied from the source account to the destination account, overwriting the Description on the destination account.

The sIDHistory attribute of the destination group is updated with the SID (and sIDHistory in the case of a Windows 2000 native mode source domain) of the source group. This means that the members of the destination group can access resources that have been migrated to the destination domain (this is the scenario that motivates cloning local groups), resources that are protected by ACLs referencing the source local group.

Unlike global and universal groups, the local group clone's membership is updated with the SIDS of all the members of the source local group (providing that trusts exist from the destination domain to all the account domains in which the members reside). This ensures that when a resource server is migrated to the destination domain, members of the source local group can still access resources on the migrated server because their membership is maintained in the destination local group whose sIDHistory grants access to resources on the migrated server. Note that if the source local group includes source domain members whose accounts have already been cloned, the destination local group membership still includes the source member accounts, rather than the member clone accounts. There is no automatic update of the destination local group membership to replace source member accounts with destination (clone) accounts.
In order to preserve the entire membership list of the local group, the destination domain must trust all account domains in which member accounts reside. Without such trusts, the members cannot be added to the destination local group. It makes sense to create these trusts anyway, since without them, the members would be unable to access resources in the destination domain. If, by chance, the local group is cloned before all required trusts have been established (and therefore, members from the untrusted domains are not added to the local group clone membership), then the administrator can fix this problem by establishing the missing trusts and re-cloning the affected local groups (which will update the membership list without adding duplicates).

Important note regarding local group cloning: When a user or global group is deleted, local groups defined in the *same* Windows NT 4.0 domain or the *same* Windows 2000 forest are updated to remove that member. When cloning both account principals (users, global groups, and universal groups) and resource principals (local groups) from the same source domain, it's important not to delete the source account principals prior to cloning the local groups so as not to alter the original local group membership list before it is copied to the local group clone. Maintaining the original local group membership is required to preserve access via destination local group memberships for account principal clones. By the same token, if account principals are cloned from a source domain in which local groups and resources will remain for some time, the local group memberships should be manually updated to reference the account principal clones prior to deleting the source accounts of these clones. These warnings are not relevant for local groups defined in a *different* Windows NT 4.0 domain or a *different* Windows 2000 forest from the domain in which a member account is deleted. In this case, the local groups are unaware that the member account was deleted; thus their membership continues to reference the deleted account. The presence of the SID of a deleted account on a local group membership provides access to clones of the deleted account.

Cloneggu.vbs

This script clones all of the account principals (users, global groups, and universal groups) in a source Domain such that the destination users, groups, and group memberships correspond to the source domain account principals and group memberships. See the clonepr.vbs description for additional detail regarding how user, global group, and universal group cloning is performed . This sample script supports migrating users to Windows 2000.

Note: This script clones accounts with well known RIDs, for example Domain Users, in addition to administrator-defined account principals in the source domain. By default, these accounts with well known RIDs reside in the Users container in the destination domain. Although this script requires an input destination organizational unit as the target in which to create the clones, the script will fail unless either the destination organizational unit indicates the Users container or the accounts with well known RIDs have been moved to the specified destination organizational unit. Cloning accounts with well known RIDs might not be desirable in all cases. For example, the members of the source Domain Administrators group are not necessarily appropriate for membership in the destination Domain Administrators group (though care should be taken that resources that grant access to the source Domain Administrators group are updated to provide access to some security principal in the post-migration environment). This script may be customized to ignore accounts with well known RIDs by "uncommenting" the section that looks like this:

```
'To Stop Cloning Well Known Sids Uncomment 4 lines below
' if HasWellKnownRid(sidString) then
'    ShouldCloneObject = False
'    exit function
' end if
```

If accounts with well known RIDs are not cloned (in particular, if the destination account SIDHistories are not updated), then some loss of network access to clones that were members of the source "well known" account may occur. To preserve network access to these resources, update ACLs and group memberships that reference a source well known account to reference additionally an appropriate destination account. The Domain Users group is a special case, in that all user accounts created in the destination domain are automatically made members of this group. Nonetheless, without the sIDHistory copied from the source Domain Users group, members of the destination Domain Users group will not have access to resources made available only to the source Domain Users group.

If cloneggu.vbs will be used to consolidate multiple source domains into a single destination domain, then it may be useful to modify this script to add a prefix or suffix to the destination account names to reduce the likelihood of encountering duplicate SAM names in the destination domain.

Clonegg.vbs

This script clones all of the global and universal groups, including groups with well known RIDs, in a domain. This script behaves like cloneggu.vbs (see description above) with the exception that user accounts are not cloned, therefore memberships of users will not be restored in the destination domain by this script unless
the users were cloned prior to running clonegg.vbs. Cloning source users after running this script will restore destination group memberships. Also see the description above for clonepr.vbs for additional detail on cloning global and universal groups. This script supports migrating users incrementally to Windows 2000 by performing a clone of all the global and universal groups and allowing subsets of users to be migrated independently with subsequent use of the clonepr.vbs script.

Clonelg.vbs

This script clones all of the resource principals ("shared" or domain local groups) in a source domain. Cloning the resource principals is required prior to migrating machines referencing those local groups to the destination domain. Affected computers are limited to domain controllers in a source Windows NT 4.0 or Windows 2000 mixed mode domains, or any computer in a source Windows 2000 native mode domain (since native mode Domain Local Groups may be referenced on any computer in the domain, not just domain controllers).

Cloning a local group results in a destination local group that possesses the entire membership list of the source local group (see the notes on local group cloning under the clonepr.vbs description above) and that contains the source local group SID in its sIDHistory. When a resource server with ACLs referencing the source local group is migrated to the destination domain, users that had access to the pre-migration resource can still access the resource after migration because they are members of the destination local group. Their membership in the destination local group results in an access token in the destination resource domain that contains the destination local group SID and sIDHistory, the latter of which grants access to the resource protected with references to the source local group.

In order to preserve the entire membership list of the local group, the destination domain must trust all account domains in which member accounts reside. Without such trusts, the members cannot be added to the destination local group. It makes sense to create these trusts anyway, since without them, the members would be unable to access resources in the destination domain. If, by chance, the local group is cloned before all required trusts have been established (and therefore, members from the untrusted domains are not added to the local group clone membership), then the administrator can fix this problem by establishing the missing trusts and re-cloning the affected local groups (which will update the membership list without adding duplicates).

Security Considerations and Requirements

ClonePrincipal performs a highly security-sensitive function by adding the primary account SID of an existing security principal to the sIDHistory of a principal in a domain in a different forest, effectively granting to the latter access to all resources accessible to

the former. This function is performed by the underlying DsAddSidHistory API, which enforces the requirements described in this section. See the Windows 2000 Software Developers Kit (SDK) reference page for DsAddSidHistory for a detailed description of the security measures and threat model associated with this API, and thus with the ClonePrincipal tool.

Authorization Requirements

ClonePrincipal requires the caller to have administrator privileges in both the source and destination domains. Specifically, the caller must be a member of the Domain Administrators group in the destination domain. (A hard-coded check for this membership is performed.) Additionally, the caller must be a member of the Administrators group in the source domain.

Domain and Trust Requirements

ClonePrincipal requires that the destination domain be Windows 2000 native mode, since only this domain type supports the sIDHistory attribute. The source domain may be either Windows NT 4.0 or Windows 2000, mixed or native mode. The source and destination domains may not be in the same forest. (Windows NT 4.0 domains are by definition not in a forest.) This inter-forest constraint ensures that duplicate SIDs (whether appearing as primary SIDs or sIDHistory values) are not created in the same forest.

ClonePrincipal requires an external trust from the source domain to the destination domain (source trusts destination) to support secure communications to the source domain using credentials inherited from the caller in the destination domain.

The ability for a clone (destination account) to access resources available to the source account requires that the applicable resource domains trust the destination domain.

Destination Domain Controller Requirements

ClonePrincipal must be run on the console of a domain controller in the destination domain. This tool cannot be run on a remote workstation.

Source Domain Controller Requirements

ClonePrincipal requires that the domain controller selected as the target for operations in the source domain be the primary domain controller (PDC in Windows NT 4.0 domains, or PDC Emulator in Windows 2000 domains). Although ClonePrincipal does not change the state of the source accounts it is cloning, source domain auditing is generated by write operations, thus the PDC (the only domain controller that supports writes) is required in Windows NT 4.0 source domains. Although Windows 2000 source domains may have more than one writable domain controller, the PDC-only restriction ensures that ClonePrincipal audits are generated on a single machine, making it easier and more efficient to monitor for audit events related to this call.

In Windows NT 4.0 source domains, the PDC (target of operations in the source domain) must be running Service Pack 4 (SP4) or later.

The TcpipClientSupport registry value must be created and set on the source domain controller (for both Windows NT 4.0 and Windows 2000 source domain controllers). Setting this value enables RPC calls over TCP transport. This is required because, by default, SAM RPC interfaces are remotable only on the named pipes transport. Using named pipes results in a credential management system that is suitable for interactively logged-on users making networked calls, but is not really flexible for a system process making network calls with user-supplied credentials. RPC over TCP is more suitable for that purpose. Setting this value does not diminish the security of the system in any way.

A new local group, <*SrcDomainName*>*$$$*, must be created in the source domain for auditing purposes (see details under "Auditing" later in this document).

Additional information and setup instructions follow.

Auditing

ClonePrincipal operations (actually, the underlying DsAddSidHistory operations) are audited to ensure that both source and destination domain administrators are able to detect when this function has been run. Auditing is mandatory in both the source and destination domains. In the destination domain, a unique audit event (Event Category "Account Management," Event Description "Add SID History") is generated for each successful (Event ID 718) or failed (Event ID 719) DsAddSidHistory operation. The audit events contain the source account name and SID and the target account name and SID, in addition to the usual caller information.

Unique "Add SID History" audit events are not available on Windows NT 4.0 systems. To generate audit events that unambiguously reflect use of DsAddSidHistory against the source domain, an update is made to a special group, whose name is the unique identifier in the audit log. A local group, <*SrcDomainName*>*$$$*, whose name is composed of the source domain's NetBIOS name followed by three DOLLAR SIGN ($) symbols (ASCII code = 0x24 and Unicode = U+0024), must be created on the source domain controller prior to calling ClonePrincipal. Each source user, global group, and universal group that is a target of this operation is added to the membership of, and then removed from the membership of this special local group. This generates Local Group Member Add (Event ID 636) and Member Delete (Event ID 637) audit events in the source domain, which can be monitored by searching for events referencing the special group name, <*SrcDomainName*>*$$$*.

Note: ClonePrincipal operations on *local* groups in a Windows NT 4.0 or Windows 2000 mixed mode source domain cannot be audited because, since local groups cannot be made members of another local group (local group nesting is not supported in Windows NT 4.0 or mixed mode), the source local group cannot be added to the special *SrcDomainName$$$* group. Lack of source domain auditing for local group clones does not present a vulnerability to the source domain because access to source domain resources is not affected by cloning of source local groups. Adding the SID of a source local group to a destination local group does not grant any additional users access to

source resources protected by that local group. Membership in the destination local group does not grant a user access to source resources; it only grants access to servers in the destination domain that have been migrated from the source domain and thus may have resources protected by the source local group's SID.

See below for instructions on how to enable auditing.

Threat Model

The following table identifies the threats that should be considered for an operation like ClonePrincipal, or more precisely, for the underlying DsAddSidHistory operation. The potential threats are described in the left-hand column. The security measures that address each threat are described in the right-hand column.

Threat	Security Measures
"Man in the Middle" Attacker intercepts the "lookup SID of source object" return call, replacing the source object's SID with an arbitrary SID of his choosing for insertion into a target object's SIDhistory.	The "lookup SID of source object" is an authenticated RPC (using the caller's administrator credentials) with packet integrity message protection, which ensures that the return call cannot be modified without detection. The destination domain controller creates a unique "AddSidhistory" audit event that reflects the SID added to the destination account sIDHistory.
"Trojan Source Domain" An attacker creates a "Trojan" source domain (on a private network) that has the same domain SID and some of the same account SIDs as the legitimate source domain. He then attempts to run ClonePrincipal in a destination domain to "steal" the SID of a source account, without the need for the real source domain Administrator credentials and without leaving an audit trail in the real source domain. His method for creating the Trojan source domain could be one of the following: 1. Steal a copy (BDC backup tape) of the source domain SAM. 2. Create a new domain, altering the domain SID on disk to match the legitimate source domain SID. Then create enough users to instantiate an account with the desired SID.	Although there are many ways for an attacker to retrieve or create a desired "source object SID," he cannot use it to update an account's sIDHistory without being a member of the destination Domain Administrators group. Because the check (on the destination domain controller) for Domain Administrator membership is hard coded, there's no way to do a disk modification to change the access control information protecting this function. The attacker's attempt to clone a "Trojan" source account will be audited in the destination domain. This attack is mitigated by reserving membership in the Domain Administrators group for only very highly trusted individuals.

3. Create a BDC replica (this requires source domain Administrator credentials). Take the replica to a private network for attack.	
"On-disk Modification of SIDhistory" A sophisticated attacker with Domain Administrator credentials and with physical access to a domain controller in the destination domain could modify an account's sIDHistory value on disk.	This attack is not enabled by use of ClonePrincipal or DsAddSidhistory. This attack is mitigated by preventing physical access to domain controllers to all but highly trusted administrators.
"Resources Vulnerable to Stolen SIDs" If an attacker has succeeded in using one of the methods described here to modify his account sIDHistory, and if the resource domains of interest trust his account domain, then he can get unauthorized access to the stolen SID's resources, potentially without leaving an audit trail in the account domain from which the SID was stolen.	Resource domain administrators protect their resources by setting up only those trust relationships that make sense from a security perspective. Use of ClonePrincipal and DsAddSidHistory is restricted to Domain Administrators (who already have broad powers) in the trusted target domain.
"Rogue Target Domain" An attacker creates a Windows 2000 domain with an account whose SIDhistory contains a SID he has stolen from a source domain. He uses this account for unauthorized access to resources.	The attacker requires Administrator credentials for the source domain in order to use ClonePrincipal / DsAddSidhistory, and will leave an audit trail on the source domain controller. The Rogue Target domain will only gain unauthorized access in other domains that trust the Rogue domain, requiring Administrator privileges in those resource domains.
"Patch Code to Remove Protections" A highly sophisticated rogue administrator or attacker with physical access to the Directory Service code could patch it to: 1. Remove the check for Domain Administrator in the code. 2. Change the call to the source domain controller that retrieves the SID to an unaudited LookupSidFromName. 3. Remove audit log calls.	Someone with physical access to the DS code and sophisticated enough to patch code already has the capability of arbitrarily modifying the sIDHistory attribute of an account. ClonePrincipal and DsAddSidhistory do not increase the vulnerability.

ClonePrincipal Setup and Configuration

This section describes the configuration steps required to setup the environment prior to running ClonePrincipal.

Set the TcpipClientSupport Registry Value

On the source primary domain controller, create the following registry DWORD value, setting the value name to TcpipClientSupport and the value data to **1**:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\TcpipClientSupport

Then, reboot the source domain controller. This registry value makes the Security Account Manager (SAM) listen on the TCP transport. ClonePrincipal will fail if this registry value isn't set on the source domain controller.

Enable Auditing in the Source and Destination Domains

In a Windows 2000 domain (for both source and destination domains):

1 In the Active Directory Users and Computers MMC snap-in, select the destination domain "Domain Controllers" container.
2 Right click on **Domain Controllers** and choose **Properties**.
3 Click on the **Group Policy** tab.
4 Select the **Default Domain Controllers Policy** and click **Edit**.
5 Under Computer Configuration\Windows Settings\Security Settings\Local Policies\ Audit Policy, double-click on audit account management.
6 In the Audit Account Management window, select both Success and Failure auditing. Policy updates take effect after a reboot, or after waiting up to 15 minutes for a refresh to take place.

In a Windows NT 4.0 source domain:

1 In User Manager for Domains, click the **Policies** menu and select **Audit**.
2 Select **Audit These Events**
3 For User and Group Management, check Success and Failure.

In the source domain (for both Windows NT 4.0 and Windows 2000):

1 In **User Manager for Domains**, click the **User** menu and select **New Local Group**.
2 Enter a group name composed of the source domain NetBIOS name appended with three dollar symbols, for example, REDMOND$$$.
3 Edit text in the description field to indicate that this group is used to audit use of cloning operations. For example, create a description like "Audit group for clone operations."
4 Make sure there are no members entered for the group. Click **OK**.

ClonePrincipal will fail if source and destination auditing aren't enabled as described here.

Set up Trust from Source to Destination

Establish a trust from the source domain to the destination domain (which must be in a different forest).

Register clonepr.dll

This setup step is not normally required since the Support Tools setup procedure used to install ClonePrincipal will register clonepr.dll. If installing ClonePrincipal manually, register clonepr.dll by running the following command in the directory in which clonepr.dll resides:
regsvr32 clonepr.dll

Syntax and Semantics

This section describes how to use the ClonePrincipal COM object and sample scripts, detailing the syntax and semantics for their use.

ICloneSecurityPrincipal Programmer's Reference

Note: This COM object and programmer's reference are preliminary and subject to change.

Clonepr.dll contains a COM object implementing the ICloneSecurityPrincipal interface. Scripts and applications calling these methods must be run on the console of the destination domain controller. They do not support running on a remote workstation.

```
interface ICloneSecurityPrincipal : IDispatch
{
        HRESULT
        Connect(
                [in] BSTR  srcDomainController,
                [in] BSTR  srcDomain,
                [in] BSTR  dstDomainController,
                [in] BSTR  dstDomain);

        HRESULT
        CopyDownlevelUserProperties(
                [in] BSTR  srcSamName,
                [in] BSTR  dstSamName,
                [in] LONG flags);

        HRESULT
        AddSidHistory(
                [in] BSTR  srcPrincipalSamName,
                [in] BSTR  dstPrincipalSamName,
                [in] LONG flags);
}
```

This COM object may be used to support a clone of a security principal (for example a user or group) from a source domain into a destination Windows 2000 domain in a different forest, such that the destination object possesses the same Windows NT 4.0 attributes and network access as the source object.

ICloneSecurityPrincipal::Connect

Establishes connections to the source and destination domain controllers preparatory to performing a clone operation.

```
HRESULT
Connect(
        [in] BSTR  srcDomainController,
        [in] BSTR  srcDomain,
        [in] BSTR  dstDomainController,
        [in] BSTR  dstDomain);
```

Parameters

srcDomainController

[in] Specifies the NetBIOS name of the primary domain controller (PDC for Windows NT 4.0, PDC Emulation for Windows 2000 source domains) in the domain from which accounts are to be cloned. If this parameter is NULL, then *srcDomain* must be specified, and the primary domain controller for that domain will be selected.

The secure connection to the source domain controller requires that the value of the entry in the following registry subkey is set to REG_DWORD=1 on the source domain controller (and that the domain controller is subsequently rebooted):

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\TcpipClientSupport

If the value of the entry in this registry subkey isn't set correctly, clone operations will return E_HANDLE.

srcDomain

[in] Specifies the NetBIOS name of the domain from which accounts are to be cloned. If this parameter is NULL, then *srcDomainController* must not be NULL.

dstDomainController

[in] Specifies the NetBIOS name of a writable domain controller for the domain in which the target clone objects are to be created. If this parameter is NULL, then *dstDomain* must be specified, and a suitable domain controller for that domain will be selected.

dstDomain

[in] Specifies the DNS name of the domain in which the target clone objects are to be created. If this parameter is NULL, then *dstDomainController* must not be NULL.

Remarks

Connections established to the source and destination domain controllers remain open until Connect is invoked again (in which case the existing connections are closed, and new connections are made), or until the final reference to the object is release by calling Release.

An external trust from the source domain to the destination domain must exist.

Auditing of Account Management Success and Failure events must be enabled in the source and destination domains. A local group named *<SrcDomainName>$$$* must exist in the source domain for auditing purposes.

The source domain may be a Windows NT 4.0 (Service Pack 4 or later) or Windows 2000 mixed or native mode domain. The destination domain must be a Windows 2000 native mode domain. If the source domain is a Windows 2000 domain, it will be considered as though it were a Windows NT 4.0 domain. Any information specific to Windows 2000 security principals will not be preserved by any of the clone operations.

The source domain must not be in same forest as destination domain.

See the "ClonePrincipal Setup and Configuration" section of this document for additional information.

Return Values

Returns S_OK if successful or **E_INVALIDARG** or an ADSI or another standard COM error code otherwise.

Requirements

Windows NT/Windows 2000: Requires Windows 2000.
Windows 95/98: Unsupported.
Windows CE: Unsupported.
Header: Declared in clonepr.h
Unicode: Implemented as Unicode and ANSI versions on Windows NT or Windows 2000.

ICloneSecurityPrincipal::AddSidHistory

Copies the SID of a security principal in a source domain into the sIDHistory of a principal in a destination domain in a different forest.

```
HRESULT
AddSidHistory(
        [in] BSTR  srcPrincipalSamName,
```

```
[in] BSTR  dstPrincipalSamName,
[in] LONG flags);
```

Parameters

srcPrincipalSamName

[in] source object SAM name relative to the source domain established with a prior call to ICloneSecurityPrincipal::Connect.

dstPrincipalSamName

[in] destination object SAM name relative to the destination domain established with a prior call to ICloneSecurityPrincipal::Connect.

flags

[in] Unused, reserved for future use

Remarks

Connections to the source and destination domain controllers must have already been established to with a prior call to ICloneSecurityPrinicpal::Connect.
The SID of *SrcPrincipal* must not already exist in the destination forest, either as a primary account SID or in the sIDHistory of an account. The exception is that AddSidhistory will not complain when attempting to add a SID to a sIDHistory that already contains the identical SID. This behavior allows AddSidhistory to be run multiple times with identical input, resulting in success and a consistent end state, for tool-developer ease-of-use.
SrcPrincipal and *DstPrincipal* must be of one of the following types:

- User
- Security Enabled Group, including:
  - "Shared" Local Group (shared by all domain controllers)
  - Global Group
  - Domain Local Group (Windows 2000 native mode only)
  - Universal Group (Windows 2000 native mode only)

The object types of *SrcPrincipal* and *DstPrincipal* must match, specifically:

- If *SrcPrincipal* is a User, *DstPrincipal* must be a User.
- If *SrcPrincipal* is a Local or Domain Local Group, *DstPrincipal* must be a Local or Domain Local Group
- If *SrcPrincipal* is a Global or Universal Group, *DstPrincipal* must be a Global or Universal Group

SrcPrincipal and *DstPrincipal* may not be one of the following types:

- Computer (workstation or domain controller)
- Inter-Domain Trust
- Account with a well known SID (for example, local Administrators, Users, and Power Users). Well known SIDs are identical in every domain

If *SrcPrincipal* has a well known relative identifier (RID) (for example, Domain Administrators), then *DstPrincipal* must possess the same well known RID in order for AddSidhistory to succeed.

Return Values

Returns S_OK if successful or E_INVALIDARG or an ADSI or another standard COM error code otherwise.

Requirements

Windows NT/Windows 2000: Requires Windows 2000.
Windows 95/98: Unsupported.
Windows CE: Unsupported.
Header: Declared in clonepr.h
Unicode: Implemented as Unicode and ANSI versions on Windows NT and Windows 2000.

ICloneSecurityPrincipal::CopyDownlevelUserProperties

Copies the Windows NT 4.0 attributes of a user account.

```
HRESULT
(CopyDownlevelUserProperties(
        [in] BSTR  srcSamName,
        [in] BSTR  dstSamName,
        [in] LONG flags);
```

Parameters

srcSamName

[in] source object SAM name relative to the source domain established with a prior call to ICloneSecurityPrincipal::Connect.

dstSamName

[in] destination object SAM name relative to the destination domain established with a prior call to ICloneSecurityPrincipal::Connect. If an account in the destination domain is already assigned this SAM account name, then that object must be an instance of the User Active Directory object class, or a class derived from User.

If either of those conditions is not met, an error is returned.

flags

[in] Unused, reserved for future use.

Remarks

Connections to the source and destination domain controllers must have already been established to with a prior call to ICloneSecurityPrincipal::Connect.

The following fields from the source user will be copied to the destination user:

- Full name
- Description
- Password/Account Flags: Password never expires; User cannot change password; Account locked out.
- User Profile: Profile path; Logon script.
- Home directory: Local path; Connect drive and location.
- Dial-in Properties (these will be converted to Windows 2000 formats).
- File and Print for Netware Properties.
- Terminal Server Properties.
- Other third-party Application Properties.

The following account control flags will be set on the destination object:

- User must change password at next logon.
- Account Disabled.

1 NULL password

Return Values

Returns S_OK if successful; or E_INVALIDARG, or an ADSI, or another standard COM error code otherwise.

Requirements

Windows NT/Windows 2000: Requires Windows 2000.
Windows 95/98: Unsupported.
Windows CE: Unsupported.
Header: Declared in clonepr.h
Unicode: Implemented as Unicode and ANSI versions on Windows NT or Windows 2000.

Sample Scripts Syntax

This section defines the command line usage of the sample scripts. These scripts must be run on the console of the destination domain controller. They cannot be run from a remote workstation.

cscript **sidhist.vbs** /srcdc:*SrcDC* /srcdom:*SrcDomain* /srcsam:*SrcSamName* /dstdc:*DstDC* /dstdom:*DstDomain* /dstSam:*DstSamName*

cscript **clonepr.vbs** /srcdc:*SrcDC* /srcdom:*SrcDomain* /srcsam:*SrcSamName* /dstdc:*DstDC* /dstdom:*DstDomain* /dstSam:*DstSamName* /dstDN:*DstDN*

cscript **cloneggu.vbs** /srcdc:*SrcDC* /srcdom:*SrcDomain* /dstdc:*DstDC* /dstdom:*DstDomain* /dstOU:*DstOU*

cscript **clonelg.vbs** /srcdc:*SrcDC* /srcdom:*SrcDomain* /dstdc:*DstDC* /dstdom:*DstDomain* /dstOU:*DstOU*

cscript **clonegg.vbs** /srcdc:*SrcDC* /srcdom:*SrcDomain* /dstdc:*DstDC* /dstdom:*DstDomain* /dstOU:*DstOU*

Sample Scripts Command Line Options
This section describes the options for use on the sample scripts command lines specified previously.

Note: Use quotation marks (“”) to enclose options that contain spaces. Both the option flag, for example “/dstou” and value must be inside the quotation marks. See example below.

/srcdc:*SrcDC*. NetBIOS name of the source domain controller (without leading \\).

/srcdom:*SrcDomain*. NetBIOS name of the source domain.

/srcsam:*SrcSamName*. SAM name (relative to *SrcDomain*) of the source object.

/dstdc:*DstDC*. NetBIOS name of the destination domain controller (without leading \\).

/dstdom:*DstDomain* – DNS name of the destination domain.

/dstsam:*DstSamName*. SAM name (relative to *DstDomain*) of the destination object.

/dstdn:*DstDN*. Full distinguished name of the destination object, for example CN=user1,OU=sales,OU=employees,DC=microsoft,DC=com. The parent container implied by this distinguished name, for example OU=Sales,OU=employees,DC=microsoft,DC=com, must already exist.

/dstou:*DstOU*. Full Distinguished Name of the destination organizational unit for clones, for example OU=sales,OU=employees,DC=microsoft,DC=com. The destination organizational unit must already exist.

Example Command Line Usage
This section provides examples of usage of the sample scripts.

Consider an example where the source domain is “hb-acct,” the source domain controller is “hb-acct-dc,” the destination domain is “microsoft.com,” and the destination domain

controller is “microsoft-dc.” In the clonepr.vbs example, the source account, “user1” is being cloned into a destination OU=employees. In the clonegg.vbs example, the destination organizational unit is the Users container, CN=Users.

Example clonepr.vbs usage:
cscript clonepr.vbs /srcdc:hb-acct-dc /srcdom:hb-acct /srcsam:user1
/dstdc:microsoft-dc /dstdom:microsoft.com /dstsam:user1
“/dstdn:CN=user1,OU=employees,DC=microsoft,DC=com”

Example clonegg.vbs usage:
cscript clonegg.vbs /srcdc:hb-acct-dc /srcdom:hb-acct
/dstdc:microsoft-dc /dstdom:microsoft.com “/dstou:CN=Users,DC=microsoft,DC=com”

Logging and Problem Diagnosis

This section describes:

1 how to collect the information logged by ClonePrincipal,
2 what information should be reported to a problem investigator or support technician,
3 errors generated by underlying DsAddSidhistory operation, and
4 known issues.

How to Investigate a Problem

The log output of the ClonePrincipal COM object in clonepr.dll is always captured in the following file:

%windir%\debug\clonepr.log

This file contains output when any of the sample scripts or a custom script using the ClonePrincipal COM object is run.

Additional progress and error reporting is sent to “stdout” by the sample scripts. It is recommended that stdout be redirected to a file for every run of a script so that the output is captured in case of an error. Sample script log output may be redirected to a file with a command line such as the following:

cscript *script.vbs optionlist* > scriptname.out

In case a problem requires investigation by a support technician, the following information will be required:

- The Windows 2000 version running on the destination domain controller on which the script is running.
- The version time stamp of the script.
- The %windir%\debug\clonepr.log file (this is clonepr.dll’s log file).
- Log output from the VBS script.
- The command line parameters used to invoke the script.
- The line number of the script where the failure occurred. This can be determined by stepping through the scripts with the Microsoft script debugger.

Common Problems include:

- Incomplete setup/prerequisites for the tools (registry key and auditing, for example).
- Incorrect command line parameters.

DsAddSidHistory Errors

Following is a partial list of DsAddSidHistory errors that are returned by ClonePrincipal, along with an explanation of what might have caused the problem.

ERROR_DS_DESTINATION_AUDITING_NOT_ENABLED
The operation requires that destination domain auditing be enabled for Success and Failure auditing of account management operations.

ERROR_DS_UNWILLING_TO_PERFORM
It might be that the user account is not one of UF_NORMAL_ACCOUNT, UF_WORKSTATION_TRUST_ACCOUNT, or UF_SERVER_TRUST_ACCOUNT.

It might be that the source principal has a well known SID.

It might be that the source principal has a well known RID and is being added to a destination principal that has a different RID. Accounts with well known RIDs, like Administrators, in the source domain can only be assigned to the account with the same RID, for example, Administrators, in the destination domain.

ERROR_DS_SRC_OBJ_NOT_GROUP_OR_USER
The source object must be a group or user.

ERROR_DS_SRC_SID_EXISTS_IN_FOREST
The source object's SID already exists in the destination forest.

ERROR_DS_INTERNAL_FAILURE;
The directory service encountered an internal failure. This shouldn't happen.

ERROR_DS_MUST_BE_RUN_ON_DST_DC
For security reasons, the operation must be run on the destination domain controller. Specifically, the connection between the client and server (destination domain controller) requires 128-bit encryption when credentials for the source domain are supplied.

ERROR_DS_NO_PKT_PRIVACY_ON_CONNECTION
The connection between client and server requires packet privacy or better.

ERROR_DS_SOURCE_DOMAIN_IN_FOREST
The source domain may not be in the same forest as destination.

ERROR_DS_DESTINATION_DOMAIN_NOT_IN_FOREST
The destination domain must be in the forest.

ERROR_DS_MASTERDSA_REQUIRED
The operation must be performed at a master DSA (writable domain controller).

ERROR_DS_INSUFF_ACCESS_RIGHTS
Insufficient access rights to perform the operation. Most likely the caller is not a member of Domain Admins for the destination domain.

ERROR_DS_DST_DOMAIN_NOT_NATIVE
Destination domain must be in Windows 2000 native mode.

ERROR_DS_CANT_FIND_DC_FOR_SRC_DOMAIN
The operation couldn't locate a domain controller for the source domain.

ERROR_DS_OBJ_NOT_FOUND
Directory object not found. Most likely the Fully Qualified Domain Name (FQDN) of the destination principal could not be found in the destination domain.

ERROR_DS_NAME_ERROR_NOT_UNIQUE
Name translation: Input name mapped to more than one output name. Most likely the destination principal mapped to more than one FQDN in the destination domain.

ERROR_DS_SRC_AND_DST_OBJECT_CLASS_MISMATCH
The source and destination object must be of the same type.

ERROR_DS_OBJ_CLASS_VIOLATION
The requested operation did not satisfy one or more constraints associated with the class of the object. Most likely because the destination principal is not a user or group.

ERROR_DS_UNAVAILABLE
The directory service is unavailable. Most likely the ldap_openW() to the Windows 2000 source domain controller failed.

ERROR_DS_INAPPROPRIATE_AUTH
Inappropriate authentication. Most likely the ldap_bind_sW() to the Windows 2000 source domain controller failed.

ERROR_DS_SOURCE_AUDITING_NOT_ENABLED
The operation requires that source domain auditing be enabled for Success and Failure auditing of account management operations.

ERROR_DS_SRC_DC_MUST_BE_SP4_OR_GREATER
For security reasons, the Windows NT 4.0 source domain controller must be running Service Pack 4 or greater.

Known Issues

1 The function, HasWellKnownRid, in cloneggu.vbs, clonegg.vbs, and clonelg.vbs

may erroneously identify admin-created accounts as having well known RIDs if the account's RID is larger than 32K (which would occur when a domain has more than 32K objects). This is due to the scripts using the CInt function to convert the RID string to an integer prior to determining if it is well known.
Implications: As released, the scripts never call HasWellKnownRid; thus this problem would not be encountered. Only if the script is modified to "uncomment" the relevant section (as described in the Cloneggu.vbs section of this document) could the problem occur, in which case accounts with RIDs greater than 32K would be skipped and not cloned.
Workaround: Replace "CInt" with "CLng" in the following segment of the script's HasWellKnownRid function.

```
if CInt(ridString) < 1000 then
    HasWellKnownRid = True
    exit function
end if
```

2 An account expiration date, if set on an NT 4.0 source account, may differ by plus or minus 1 hour on the destination account if the expiration date is in Daylight Savings Time and the current date is not in Daylight Savings Time, or vice verse. Because expiration times are often set to midnight, it may appear that the destination account expiration date is off by a full day, when in fact it's only off by 1 hour. This problem does not occur with Windows 2000 source accounts.

3 The clonelg.vbs command line help (printed by running "cscript clonelg.vbs") erroneously lists "/srcsam" under Parameters. The Usage that is printed does not include "/srcsam" and is correct.