OpenAFS for Windows Release Notes

OpenAFS for Windows Release Notes

Abstract

This document provides a series of usage notes regarding the OpenAFS for Windows client, supported platforms, contribution information, debugging techniques, and a reference to supported Windows registry values.

Table of Contents

Preface

The Andrew File System (AFS) is a globally-accessible location-independent file system that uses local caching to increase its performance. An AFS client accesses files anonymously or authenticated via Kerberos. The global AFS is partitioned into cells. Each AFS cell is a collection of AFS volumes that are administered by a common entity. AFS cells can be administered by a department even when the associated Kerberos authentication realms are managed by a much larger organization. AFS clients and servers take advantage of Kerberos cross-realm authentication to permit authenticated access by entities located outside the local realm. Authorization is enforced by the use of directory level access control lists of individual or group identities.

The AFS volume is a tree of files and sub-directories. AFS volumes are created by administrators and are joined to an AFS cell via the use of a mount point. Once a volume is created, users can create files and directories as well as mount points and symlinks within the volume without regard for the physical location of the volume. Administrators can move the volume to another server as necessary without the need to notify users. In fact, the volume move can occur while files in the volume are in use.

AFS volumes can be replicated to read-only copies. When accessing files from a read-only replica, clients will read all of the data from a single replica. If that replica becomes unavailable, the clients will failover to any replica that is reachable. Users of the data are unaware of where the replicas are stored or which one is being accessed. The contents of the replicas can be updated at any time by *releasing* the current contents of the source volume.

OpenAFS for Windows (OAFW) provides AFS client access on Microsoft Windows operating systems. It strives to maintain transparency such that the user is unaware of the distinction between the use of AFS and Microsoft Windows file shares. OAFW can be part of a single sign-on solution by allowing credentials for a Kerberos principal to be obtained at logon and for that principal to be used to obtain AFS tokens for one or more cells. OAFW is implemented as a native installable file system and maintains the portability of file paths by its use of the \\AFS UNC server name.

OpenAFS is the product of an open source development effort begun on 1 November 2000. OpenAFS is maintained and developed by a group of volunteers with the support of the end user community. When OpenAFS is used as part of your computing infrastructure, please contribute to its continued growth.

Chapter 1. Installer Options

OpenAFS can be installed either as a new installation or an upgrade from previous versions of either OpenAFS for Windows or the former IBM AFS 3.6 for Windows. Installers are provided as Windows Installer packages (.msi) [http://msdn.microsoft.com/en-us/library/cc185688%28v=vs.85%29.aspx] that are built using the open source WiX Toolkit [http://wix.sourceforge.net/]. The MSI can be customized for organizations via the use of MSI Transforms [http://msdn.microsoft.com/en-us/library/aa367447%28v=vs.85%29.aspx] (see MSI Deployment Guide [40])

Chapter 2. System Requirements

2.1 Supported Operating Systems

- Microsoft Windows XP Home SP2 and SP3
- Microsoft Windows XP Professional SP2 and SP3
- Microsoft Windows XP 64 SP1 and SP2
- Microsoft Windows 2003 Server SP1 and SP2 (32-bit and 64-bit Intel)
- Microsoft Windows 2003 R2 Server (32-bit and 64-bit Intel)
- Microsoft Windows Vista (32-bit and 64-bit Intel)
- Microsoft Windows 2008 Server (32-bit and 64-bit Intel)
- Microsoft Windows 7 (32-bit and 64-bit Intel)
- Microsoft Windows 2008 Server R2 (64-bit Intel)
- Microsoft Windows 8 (32-bit and 64-bit Intel)
- Microsoft Windows Server 2012 (64-bit Intel)

2.1.1 Unsupported Operating Systems

- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows 98 OSR2
- Microsoft Windows ME
- Microsoft NT
- Microsoft Windows 2000 Workstation
- Microsoft Windows 2000 Server
- Microsoft Windows XP Home (pre-SP2)
- Microsoft Windows XP Professional (pre-SP2)
- Microsoft Windows 8 RT (ARM)

Older releases of OpenAFS are available for download if unsupported operating systems must be used. The last version of OpenAFS with support for Win9x is 1.2.2b. The last version with support for Windows NT 4.0 is 1.2.10. The last version to support Windows 2000 and XP SP1 is 1.6.1.

2.2 Disk Space

Up to 60mb required for the OpenAFS binaries plus 100MB for the default AFSCache file. The size of the AFSCache file may be adjusted via the Registry after installation. The maximum cache size for 32-bit Windows is approximately 1.2GB. On 64-bit Windows there is no enforced limit on the cache size.

2.3 Additional Software Packages

Heimdal [https://www.secure-endpoints.com/heimdal] or MIT Kerberos for Windows [http://web.mit.edu/kerberos/dist/index.html] 3.2.x if Kerberos v5 authentication support is desired. Heimdal is preferred over MIT Kerberos as it will provide OpenAFS the ability to offer enhanced capabilities in future releases. For 64-bit Windows installations, the 64-bit version of Kerberos for Windows is required. For 32-bit Windows installations, the 32-bit version of Kerberos for Windows is required. See 3.2 Kerberos v5 Requirements for additional details.

Chapter 3. Operational Notes

3.1. Unicode Support

Starting with the 1.5.50 release of OpenAFS for Windows, each of the AFS Client Service, the AFS Explorer Shell Extension, and the command-line tools are Unicode enabled. No longer is OpenAFS restricted to accessing file system objects whose names can be represented in the locale specific OEM code page. This has significant benefits for end users. Most importantly it permits non-Western languages to now be used for file system object names in AFS from Microsoft Windows operating systems. Now that Unicode names are supported, Roaming User Profiles [http://en.wikipedia.org/wiki/Roaming_user_profile] and Folder Redirection [http://en.wikipedia.org/wiki/Folder_redirection] will no longer fail when a user attempts to store an object with a name that cannot be represented in the OEM code page.

Unicode names are stored in AFS using UTF-8 encoding. UTF-8 is supported as a locale on MacOS X, Linux, Solaris, and most other operating systems. This permits non-Western object names to be exchanged between Microsoft Windows and other operating systems. The OpenAFS for Windows client also implements Unicode Normalization [http://en.wikipedia.org/wiki/Unicode_normalization] as part of the name lookup algorithm. This is necessary because Unicode does not provide a unique representation for each input string. The use of normalization permits a file system object name created on MacOS X to be matched with the same string entered on Microsoft Windows even though the operating system's choice of representation may be different.

It is important to note that AFS file servers are character-set agnostic. All file system object names are stored as octet strings without any character set tagging. If a file system object is created using OEM Code Page 858 and then interpreted as UTF-8 it is likely that the object name will appear to be gibberish. OpenAFS for Windows goes to great lengths to ensure that the object name is converted to a form that will permit the user to rename the object using Unicode. Accessing UTF-8 names on UNIX systems that have the locale set to one of the ISO Latin character sets will result in the UTF-8 strings appearing to be gibberish.

UNIX AFS can not perform Unicode Normalization for string comparisons. Although it is possible to store and read Unicode object names, it is possible that a user may not be able to open an object by typing the name of the object at the keyboard. GUI point and click operations should permit any object to be accessed.

3.1.1. Interoperability with MacOS X

MacOS X uses UTF-8 Normalization Form D (NFD) whereas Microsoft Windows and most other applications use UTF-8 Normalization Form C (NFC). The difference is that in NFD Unicode character sequences containing diacritical marks are decomposed whereas in NFC the Unicode character sequences use combined characters whenever possible. Whereas Microsoft Windows can display and manipulate files stored using NFD, MacOS X Finder does have trouble with filenames that are NFC encoded. All file names stored by the OpenAFS Windows client use NFC.

3.2. Requirements for Kerberos v5 Authentication

The OpenAFS distribution ships with its own implementation of Kerberos v4 and although it is Kerberos v5 capable, it relies on third-party Kerberos v5 libraries. The OpenAFS 1.4 series (and later) integrates with

Heimdal [https://www.secure-endpoints.com/heimdal] or MIT Kerberos for Windows [http://web.mit.edu/kerberos/] 2.6.5 and above. OpenAFS Kerberos v5 capable functionality includes integrated logon, the AFS Authentication Tool, the Network Identity Manager AFS provider, and the aklog command. These tools provide support for Kerberos v5 authentication including acquisition and automatic renewal of AFS tokens as well as support for single sign-on via the Microsoft Windows Kerberos Logon Service.

The recommended versions of Heimdal [https://www.secure-endpoints.com/heimdal] and MIT Kerberos for Windows [http://web.mit.edu/kerberos/] are distributed by Secure Endpoints Inc. [https://www.secure-endpoints.com/]. As of this writing, the Secure Endpoints Inc. distribution provides 64-bit Windows support which is unavailable from MIT. KFW 3.2.2 includes Network Identity Manager 1.3.1 which integrates with the AFS Provider installed as part of OpenAFS for Windows. The most recent version of Network Identity Manager is version 2.1 which is available as an independent upgrade to MIT Kerberos for Windows. Heimdal does not include a version of Network Identity Manager.

With Heimdal or Kerberos for Windows installed, the OpenAFS for Windows client can perform authentication to AFS services using Kerberos v5 service tickets as AFS tokens. When a Kerberos v5 derived AFS token is used, all of the AFS Volume Location and File Servers within the authenticated cell must support Kerberos v5. If a Kerberos v5 based token is presented to an AFS server that does not support them, the server will be unable to respond to the client. Attempts to access AFS volumes stored on such a server will fail with the Windows STATUS_NO_KERB_KEY (0xC0000322L) error. Kerberos v5 based tokens are supported by OpenAFS revisions 1.2.8 or later. IBM AFS 3.6 servers do not support Kerberos v5.

3.2.1. Active Directory

Microsoft Windows Active Directory can be used as a Kerberos v5 KDC in conjunction with OpenAFS.

- There are two things to consider when using an Active Directory as the Kerberos realm that issues the AFS service ticket. First, the Kerberos v5 tickets issued by Active Directory can be quite large when compared to tickets issued by traditional UNIX KDCs due to the inclusion of Windows specific authorization data (the Microsoft PAC) [http://msdn.microsoft.com/en-us/library/cc237917%28v=prot.10%29.aspx]. If the issued tickets are larger than 344 bytes, OpenAFS 1.2.x servers will be unable to process them and will issue a RXKADBADTICKET error. OpenAFS 1.4 (and beyond) servers can support the largest tickets that Active Directory can issue.

- Second, the Kerberos v5 tickets issued by Windows 2003 Active Directory are encrypted with the DES-CBC-MD5 encryption type (enctype). OpenAFS 1.2.x servers only support the DES-CBC-CRC enctype. As a result, OpenAFS 1.2.x servers cannot process the resulting Kerberos v5 tokens. Windows 2000 Active Directory issues tickets with the DES-CBC-CRC enctype. Windows Server 2008 R2 Active Directory domain by default disables use of DES-CBC-MD5 and it must be enabled.

  Microsoft has documented in Knowledge Base article 832572 [http://support.microsoft.com/kb/832572/] a new NO_AUTH_REQUIRED flag that can be set on the account mapped to the AFS service principal. When this flag is set, the PAC authorization data will not be included in the ticket. Setting this flag is recommended for all accounts that are associated with non-Windows services and that do not understand the authorization data stored in the PAC. This flag cannot be used if AFS service tickets are obtained via cross-realm using an Active Directory user principal.

  Note that an Active Directory computer object cannot be used for the afs service principal. A user object must be used.

- Starting with Windows 7 and Windows Server 2008 R2, Microsoft has disabled the single DES encryption types,TechNet: Changes in Kerberos Authentication [http://technet.microsoft.com/en-us/library/dd560670(WS.10).aspx]. DES must be enabled via Group Policy in order for Active Directory

to be used as a KDC for OpenAFS. Enable weak encryption becuase of AFS... Start > Administrative Tools > Group Policy Management Expand Forest > Domains > (domain name) > Group Policy Objects > Default Domain Policy Right-click "Default Domain Policy" and select "Edit" Expand "Computer Configuration" > "Policies" > "Windows Settings" > "Security Settings" > "Local Policies" > "Security Options" Double click "Network security: Configure encryption types allowed for Kerberos" Select "Define this policy setting", then select "DES_CBC_CRC" and all the others... Press "OK"

3.2.2. The krb524 Service is no longer supported

Before there was native support for Kerberos v5 derived AFS tokens, the krb524 service was used to convert a Kerberos v5 service ticket into a Kerberos v4 service ticket that could in turn be used to construct an AFS authentication token. As of OpenAFS 1.2.8 [14 December 2002], support was added to allow the immediate use of Kerberos v5 tickets as AFS (2b) tokens. This is the first building block necessary to break away from the limitations of Kerberos v4 with AFS. By using Kerberos v5 directly the security holes inherent in Kerberos v4 cross-realm are avoided. Use of cryptographically stronger algorithms for authentication and encryption become a possibility.

Another reason for using Kerberos v5 directly is because the krb524 service runs on port (4444/udp), which has increasingly been blocked by Internet service providers. The port was used to spread a worm which attacked Microsoft Windows in the Summer of 2003. When the port is blocked users find that they are unable to authenticate.

Replacing the Kerberos v4 ticket with a Kerberos v5 ticket is a win in all situations except when the cell name does not match the realm name and the principal names placed into the ACL's are not the principal names from the Kerberos v5 ticket. Unfortunately, some organizations have AFS cell names and Kerberos realm names which differ by more then just typographic case and rely the krb524d service to map the Kerberos v5 client principal name from realm FOO to a Kerberos v4 principal in realm BAR. This allows user@FOO to appear to be user@bar for the purposes of accessing the AFS cell.

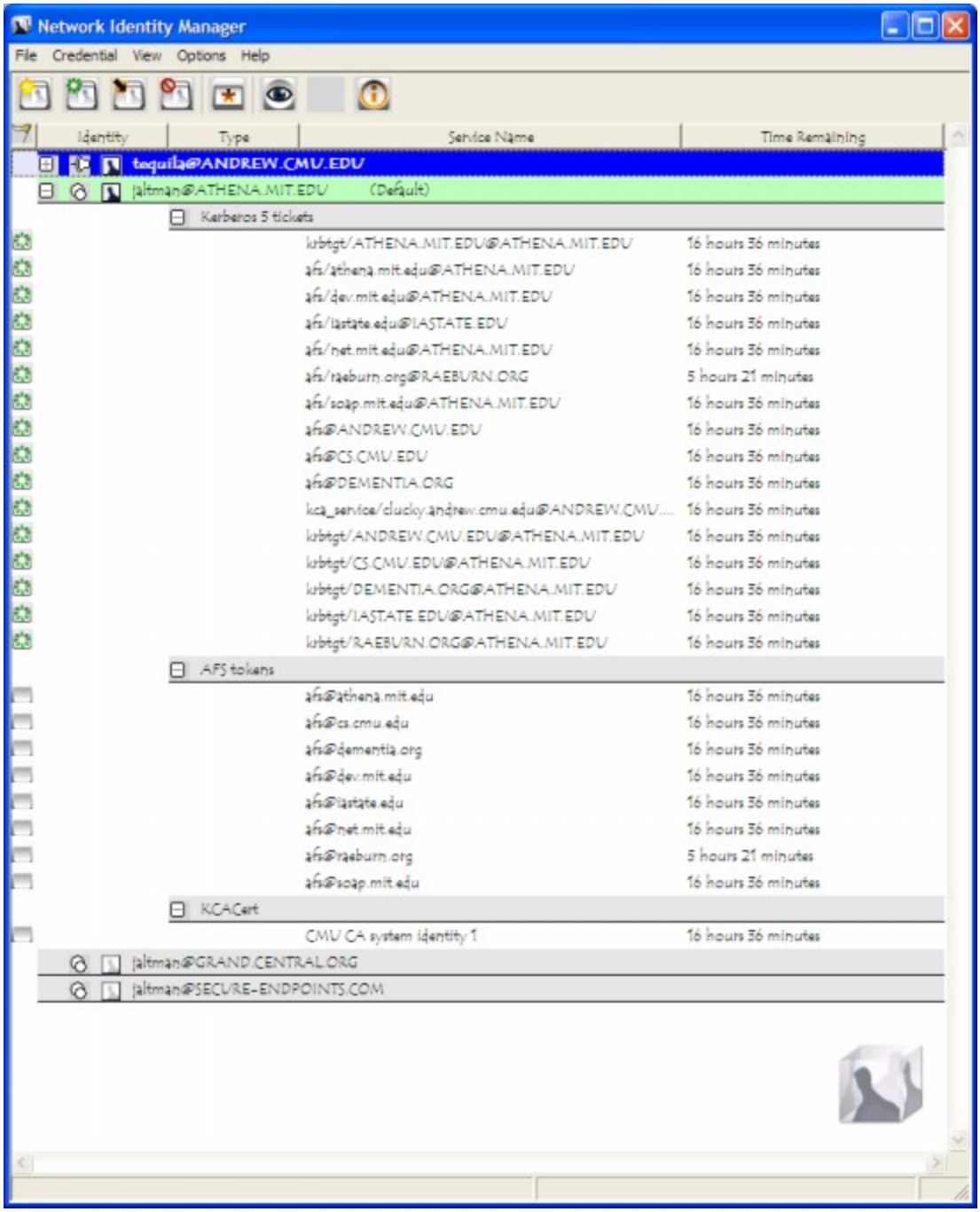
To support this mode of operation OpenAFS for Windows versions 1.4.x through 1.6.x supported a registry value, Use524 [84], that forced the use of krb524d within the AFS Authentication Tool and during integrated logon. Previous revisions of this documentation advised that this option should only be used by individuals until such time as their organizations transitioned away from the krb524 service.

Over the last few years all Kerberos distributions have removed support for Kerberos v4. As a result, OpenAFS can no longer support the krb524d service and the functionality has been removed from the source tree for the 1.7.x release.

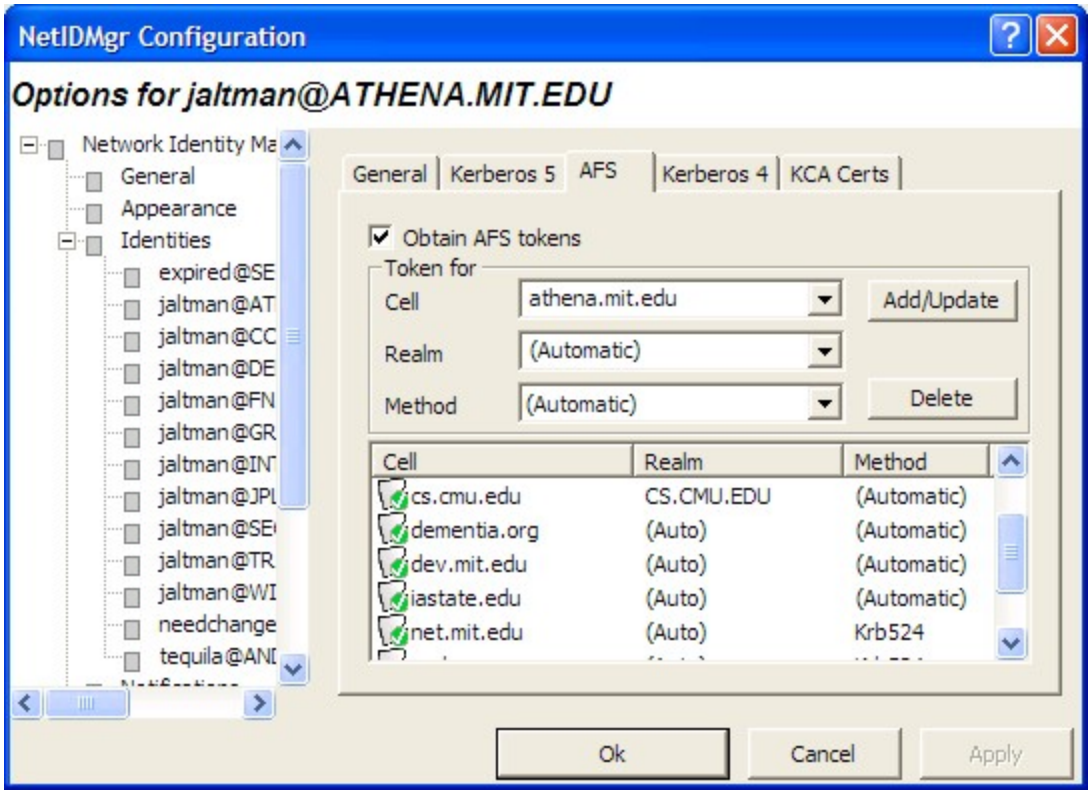
As an alternative, sites should be aware that the OpenAFS 1.4.x servers permit the use of a secondary realm name that can be treated as equivalent to the cell name for authentication. This functionality can be used to avoid the need for the krb524 service if and only if both realms are managed by the same administrative entity.

3.2.3. Network Identity Manager Provider

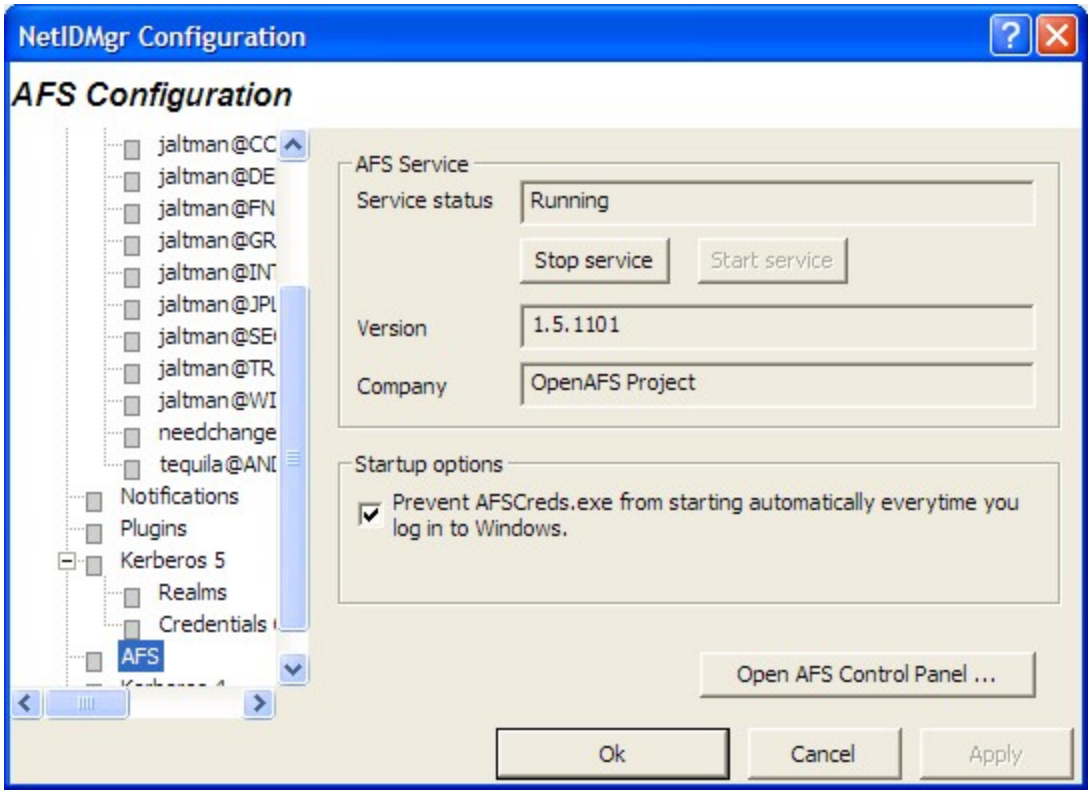
As of release 1.5.9, OpenAFS for Windows includes a Network Identity Manager Provider for obtaining AFS tokens. This plug-in is a contribution from Secure Endpoints Inc. [https://www.secure-endpoints.com/] Network Identity Manager is a multiple identity credential management tool that ships with MIT Kerberos for Windows [http://web.mit.edu/kerberos/] version 3.0 and above. The OpenAFS plug-in requires Heimdal [https://www.secure-endpoints.com/heimdal] or MIT Kerberos for Windows [http://web.mit.edu/kerberos/] version 3.1 or above.



The Network Identity Manager replaces the former KFW 2.6.x ticket manager, "Leash", and when combined with the OpenAFS Provider it can be used as a replacement for the AFS Authentication Tool (afscreds.exe). Unlike both Leash and the AFS Authentication Tool, Network Identity Manager with the OpenAFS Provider can easily acquire and renew AFS tokens for multiple cells from one or more Kerberos v5 identities.



The AFS configuration panel for each Kerberos v5 identity is used to configure which cells credentials should be obtained for and how they should be obtained. If the cell to realm mapping cannot be automatically determined, it can be explicitly specified. If the cell does not support Kerberos v5 tickets as tokens, then a krb524 service can be configured.



The OpenAFS Provider configuration panel can be used to check the status of the AFS Client Service and its version. An optional checkbox is provided that will prevent the AFS Authentication Tool from being started by Windows after login. A shortcut to the OpenAFS Control Panel is also provided.

As of OpenAFS 1.5.66, the Network Identity Manager OpenAFS Provider displays the same AFS Lock notification icon generated by the AFS Authentication Tool. The AFS Lock can be used to determine if:

- one or more AFS tokens are valid
- no AFS tokens are present but the AFS service is running
- the AFS Service is not running
- the AFS Service is running but there is a communication error preventing access to \\AFS

3.2.4. Heimdal 1.5, MIT 4.x, and Weak Encryption Types

Just as Microsoft disabled the use of Weak Encryption Types in Windows 7 and Windows Server 2008 R2, Heimdal and MIT have disabled the use of weak encryption types in their latest releases. In order to use Heimdal 1.5 or MIT Kerberos 1.9 or later with OpenAFS, the weak encryption types including DES-CBC-CRC and DES-CBC-MD5 must be enabled. In Heimdal, this is performed by adding "allow_weak_crypto = true" to the [libdefaults] section of the %SystemRoot%\ProgramData\Kerberos\krb5.conf file. In MIT KFW 4.x, this is performed by adding "allow_weak_crypto = true" to the [libdefaults] section of the %SystemRoot%\ProgramData\MIT\Kerberos5\krb5.ini file.

Futures versions of OpenAFS will not have this requirement.

3.3. The Former use of the Microsoft Loopback Adapter by the OpenAFS Client Service

This section is preserved for those sites that may want to manually configure the OpenAFS Client Service to run as an SMB Gateway to AFS instead of using the native IFS file system redirector driver. When the IFS driver is active, the Microsoft Loopback Adapter is ignored. The OpenAFS 1.7.x installer will not install a Microsoft Loopback Adapter by default nor will it remove one if already present on the machine.

The Microsoft Loopback Adapter (MLA) is installed with a name "AFS" and a pre-assigned IP address of 10.254.254.253. The MLA is bound to the "Client for Microsoft Networks" service and not bound to the "File and Printer Sharing for Microsoft Networks" service. If the MLA is unbound to "Client Microsoft Networks", the OpenAFS Client Service will become inaccessible when the machine is disconnected from the network. If the MLA is bound to "File and Printer Sharing ..." there will be a service type collision between the "AFS" SMB Service and the local machine's File Sharing Service. This will result in the OpenAFS client service becoming inaccessible and the "NET VIEW \\AFS" command will return a "System Error 52" message. To correct the problem:

- stop the AFS Client Service
- bind the "Client for Microsoft Networks" to the MLA
- unbind "File and Printer Sharing for Microsoft Networks" from the MLA
- disable and then re-enable the MLA
- start the AFS Client Service

When the MLA is not installed the NETBIOS name published by the OpenAFS SMB server must be unique in order to avoid name conflicts on public network. The unique name will take the form "*MACHINE*-AFS". One of the benefits of using the MLA is that the NETBIOS name does not have to be published on any adapter other than the MLA. Therefore the chosen name is no longer required to be globally unique. Instead the NETBIOS name associated with the AFS Client Service is simply "AFS" and portable UNC paths of the form \\AFS\cellname\path can now be used on all machines.

3.4. Using Freelance (Dynamic Root) Mode to Improve Mobility

Historically, when the OpenAFS Client Service starts it must mount the "root.afs" volume of the default cell. The "root.afs" volume contains the set of mount points to the "root.cell" volumes of various cells the administrator of the default cell believes should be accessible. If the "root.afs" volume is inaccessible when the client service starts, the service will terminate unexpectedly. Since many users now use laptops or otherwise operate in disconnected environments in which a Virtual Private Network (VPN) connection may be required to access the cell's servers, it is often the case that the "root.afs" volume is unreachable and the OpenAFS Client Service can not successfully start.

Freelance mode dynamically constructs a fake "root.afs" volume from mount points and symlinks stored in the local registry. This permits the OpenAFS Client Service to operate in these environments.

The content of the fake "root.afs" volume is dynamically generated as cells are accessed. When the fake "root.afs" volume is initially constructed it will only contain two mount points: a *regular path* and *read-write path* mount point used to access the "root.cell" volume of the default AFS cell. Any attempt to access a valid cell name will result in a new mount point being created in the fake "root.afs" volume. If the

cellname begins with a "." the mount point will be a *read-write path*; otherwise the mount point will be a *regular path*. These mount points are preserved in the registry at key:

HKLM\SOFTWARE\OpenAFS\Client\Freelance

Additional mount points may be manually created using the "fs mkmount" command. Mount points may be removed using the "fs rmmount" command.

>fs mkmount \\AFS\athena.mit.edu root.cell athena.mit.edu

>fs mkmount \\AFS\.athena.mit.edu root.cell athena.mit.edu -rw

>fs rmmount \\AFS\athena.mit.edu

>fs rmmount \\AFS\.athena.mit.edu

Symlinks may also be created within the Freelance "root.afs" volume.

>symlink make \\afs\link \\afs\athena.mit.edu\user\j\a\jaltman

>symlink list \\afs\link

'\\afs\link' is a symlink to 'athena.mit.edu\user\j\a\jaltman'

>symlink rm \\afs\link

The symlinks are stored in the registry at:

HKLM\SOFTWARE\OpenAFS\Client\Freelance\Symlinks

NET VIEW \\AFS can be used to browse all of the entries from the command line.

3.5. Locating AFS Volume Database Servers via DNS

The OpenAFS for Windows client will use DNS SRV records and DNS AFSDB records to discover the location of AFS Volume Database servers when entries for the cell are not present in either the client's CellServDB registry store or file (\%ALLUSERSPROFILE%\OpenAFS\Client\CellServDB or \%PROGRAMFILES%\OpenAFS\Client\CellServDB). Also see Registry Configuration for AFS Volume Database Servers.

3.6. Obtaining AFS Tokens as a Integrated Part of Windows Logon

OpenAFS for Windows installs a WinLogon Authentication Provider to provide Single Sign-On functionality (aka Integrated Logon.) Integrated Logon can be used to obtain AFS tokens when the Windows username and password match the username and password associated with the default cell's Kerberos realm. For example, if the Windows username is "jaltman" and the default cell is "your-file-system.com", then Integrated Logon can be successfully used if the windows password matches the password assigned to the Kerberos principal "jaltman@YOUR-FILE-SYSTEM.COM". The realm "YOUR-FILE-SYSTEM.COM" is obtained by performing a domain name to realm mapping on the hostname of one of the cell's Volume Database servers.

Integrated Logon is required if roaming user profiles are stored within the AFS file system. OpenAFS does not provide tools for synchronizing the Windows and Kerberos user accounts and passwords. Integrated Logon can be enabled or disabled via the LogonOptions [78] registry value.

When Heimdal or KFW is installed, Integrated Logon will use it to obtain tokens using Kerberos v5. If you must use the *deprecated* kaserver for authentication instead of Kerberos v5, the use of KFW can be disabled via the EnableKFW [83] registry value.

Integrated Logon will not transfer Kerberos v5 tickets into the user's logon session credential cache. This is no longer possible on Vista and Windows 7.

Integrated Logon does not have the ability to cache the username and password for the purpose of obtaining tokens if the Kerberos KDC is inaccessible at logon time.

Integrated Logon supports the ability to obtain tokens for multiple cells. For further information on how to configure this feature, read about the TheseCells [81] registry value.

Depending on the configuration of the local machine, it is possible for logon authentication to complete with one of the following user account types:

- Local Machine Account (*LOCALHOST* domain)
- Domain or Forest Account
- Domain or Forest Account NETBIOS-compatible name
- Kerberos Principal mapped to a local or domain or forest account

For each "domain" context, the following properties are configurable:

- Obtain AFS Tokens at Logon
  - Yes
  - No
- Alternate Kerberos Realm Name - combined with the username to construct a Kerberos principal
- TheseCells - A list of cell names other than the workstation cell for which tokens should be obtained
- Fail Logons Silently
  - Yes
  - No
- Logon Script to Execute
- Logon Retry Interval
- Logon Sleep between Failure Interval

Within a "domain" context it is often desireable to apply alternate rules for a particular user. The rules can include a username substitution.

- Obtain AFS Tokens at Logon
  - Yes

    - No
- Alternate User Name
- Alternate Kerberos Realm Name - combined with the username to construct a Kerberos principal
- TheseCells - A list of cell names other than the workstation cell for which tokens should be obtained
- Fail Logons Silently
    - Yes
    - No
- Logon Script to Execute
- Logon Retry Interval
- Logon Sleep between Failure Interval

The configuration hierarchy is specified in the registry under the HKLM\SYSTEM\CurrentControlSet \Services\TransarcAFSDaemon\NetworkProvider\Domain key. For example:

- ...\NetworkProvider\Domain\LOCALHOST\
- ...\NetworkProvider\Domain\LOCALHOST\Administrator\
- ...\NetworkProvider\Domain\AD\
- ...\NetworkProvider\Domain\AD.EXAMPLE.ORG\

From the perspective of configuration, the Full domain name and the NETBIOS-compatibility name are separate entities.

3.7. AFS Authentication Tool Command Line Options

The AFS Authentication Tool (afscreds.exe) has been deprecated in favor of Network Identity Manager. afscreds.exe is no longer installed by default in the OpenAFS 1.7 release series. The following information is for historical reference.

The AFS Authentication Tool (afscreds.exe) supports several command line options:

- -A = autoinit
- -E = force existing afscreds to exit
- -I = install startup shortcut
- -M = renew drive maps (deprecated)
- -N = IP address change detection
- -Q = quiet mode. do not display start service dialog if afsd_service is not already running
- -S = show tokens dialog on startup

- -U = uninstall startup shortcut
- -X = test and do map share (deprecated)
- -Z = unmap drives (deprecated)

autoinit will result in automated attempts to acquire AFS tokens when afscreds.exe is started. afscreds.exe will attempt to utilize tickets stored in the MSLSA credentials cache; any existing CCAPI credentials cache; and finally display an Obtain Tokens dialog to the user. When used in combination with IP address change detection, afscreds.exe will attempt to acquire AFS tokens whenever the IP address list changes and the Kerberos KDC is accessible.

The renew drive maps option is used to ensure that the user drive maps constructed via the OpenAFS tools (not NET USE) are re-constructed each time afscreds.exe is started.

By default afscreds.exe is configured by the OpenAFS.org installers to use "-A -N -M -Q" as startup options. Currently, there is no user interface to change this selection after install time although these options may be altered via the registry on either per machine or per user basis. See AfscredsShortcutParams [84] in Appendix A.

Due to conflicts with Vista and Windows 7 User Account Control, the Drive Letter Mount and Advanced tabs of the AFS Authentication Tool were disabled beginning with the 1.5.66 release.

3.8. The "AFS Client Admins" Authorization Group

The OpenAFS for Windows client supports a local Windows authorization group named "AFS Client Admins". This group is used in place of the "Administrators" group to determine which users are allowed to modify the AFS Client Service configuration via the AFS Control Panel (afs_config.exe) or fs.exe command line tool. The following fs.exe commands are now restricted to members of the "AFS Client Admins" group:

- checkservers with a non-zero timer value
- setcachesize
- newcell
- sysname with a new sysname list
- exportafs
- setcell
- setserverprefs
- storebehind
- setcrypt
- cscpolicy
- trace
- minidump

The creation or removal of mount points and symlinks in the Freelance "root.afs" volume are also restricted to members of the "AFS Client Admins" group.

The initial membership of the "AFS Client Admins" group when created by the installer is equivalent to the local "Administrators" group. If a user is added to the "Administrators" group after the creation of the "AFS Client Admin" group, that user will not be an AFS Client Administrator. Only users that are members of the "AFS Client Admins" group are AFS Client Administrators. The local "SYSTEM" account is an implicit member of the "AFS Client Admins" group.

Setting the default sysname for a machine should be done via the SysName registry value [55] and not via "fs sysname".

3.9. OpenAFS Support for UNC Paths

The OpenAFS client supports UNC paths everywhere. UNC paths provide a canonical name for resources stored within AFS. UNC paths should be used instead of drive letter mappings whenever possible. This is especially true when specifying the location of roaming profiles and redirected folders.

Power users that make extensive use of the command line shell, cmd.exe, should consider using JP Software's Take Command command processor. Unlike cmd.exe, the JPSoftware shells fully support UNC paths as the current directory. JPSoftware added special recognition for OpenAFS to its command shells starting in version 7.0. AFS paths can be entered in UNIX notation (e.g., /afs/openafs.org/software), space utilization reports the output of the volume status for the specified path, and many AFS specific functions and variables have been added to the command language. Take Command 14.03 includes support for OpenAFS IFS Reparse Points and Hard Links.

JPSoftware's web site is http://www.jpsoft.com [http://www.jpsoft.com/].

Microsoft PowerShell 1.0, 2.0 and 3.0 also support UNC paths as the current directory.

The Cygwin environment also supports UNC paths as the current directory. Enter AFS paths with two leading forward slashes: //afs/grand.central.org/software/openafs/. As of Cygwin 1.7.18-1, AFS Symbolic Links are supported natively by the Cygwin environment.

3.10. aklog.exe

The OpenAFS Client ships with its own version of aklog.exe which should be used in preference to those obtained by other sources. The OpenAFS aklog.exe supports Kerberos v5 as well as the ability to auto-generate AFS IDs within foreign PTS databases.

```
Usage: aklog [-d] [[-cell | -c] cell [-k krb_realm]]
             [[-p | -path] pathname]
             [-noprdb] [-force]
             [-5]

             -d = output debugging information.
             cell = zero or more cells for which tokens will be obtained
             krb_realm = the kerberos realm of the cell.
             pathname = the directory for which authentication is required
             -noprdb = don't try to determine AFS ID.
             -5 = use Kerberos v5.
                  (only Kerberos v5 is available)
             No commandline arguments means authenticate to the local cell.
```

3.11. OpenAFS Servers on Windows are Unsupported

The AFS Server functionality provided as part of the OpenAFS install package might work but should be considered highly experimental. It has not been thoroughly tested. Any data which would cause pain if lost should not be stored in an OpenAFS Server on Windows.

Known issues include lack of support for power management and dynamic network configuration. Salvager is also known to crash.

3.11.1. OpenAFS Server Installation

When the OpenAFS Server is installed, the TransarcAFSServer service (bosctlsvc.exe) will be installed and configured. The TransarcAFSServer service will auto-start the traditional AFS bos server. The former AFS Server Configuration wizard makes assumptions that no longer hold true and it has therefore been disabled. However, following the instructions for installing the AFS Servers on UNIX it is possible to properly configure the AFS Servers on Microsoft Windows. The AFS Server binaries, configuration files, and log files are installed under %Program Files%\OpenAFS\Server. kaserver has been deprecated and its use is strongly discouraged. [http://www.openafs.org/no-more-des.html] Instead, Active Directory or some other Kerberos v5 KDC should be used in its place.

3.11.2. Using the AFS Client Service when the Server is installed

A few notes on the usage of the AFS Client Service if it is going to be used with the OpenAFS AFS Server:

- Freelance mode should be disabled when the AFS Client Service is installed on the same machine as the AFS Server,. Otherwise, it will be impossible to manipulate the contents of the root.afs volume for the hosted cell without constructing an explicit mountpoint to the root.afs volume from another volume.

- The AFS Server and related tools only support the built in kaserver (Kerberos IV). If kaserver is being used, MIT Kerberos for Windows [http://web.mit.edu/kerberos/] should not be installed or must be disabled via the EnableKFW [83] registry value.

- The AFS Servers are not aware of power management events nor are they aware of network configuration changes. It is strongly advised that the AFS servers be installed only on systems that will not be shutdown or suspended unexpectedly. An inadvertent shutdown will corrupt volume data.

3.12. OpenAFS Debugging Symbols and Checked Builds

The OpenAFS for Windows installers include Debugging Symbol files which should be installed if you are experiencing problems and need to send crash reports or view OpenAFS stack traces within Process Explorer or Debug View. This is true for both the release and the debug versions of the installers. The difference between the release and debug versions are:

- whether or not the binaries were compiled with optimization (release: yes, debug: no)

- whether or not the debug symbols are installed by default (release: no, debug: yes)
- whether or not *fs trace* logging is turned on by default (release: no, debug: yes)
- whether or not additional debug statements were compiled into the binaries (release: no, debug: yes)

3.13. Large File (64-bit) Support

As of release 1.5.3, OpenAFS for Windows supports files larger than 2GB. The maximum file size is now 16777216 terabytes when the AFS File Server supports large files. If the AFS File Server does not support 64-bit file sizes, then the maximum size of files stored on that server remains 2GB.

3.14. Encrypted AFS Network Communication

The OpenAFS for Windows installer by default activates a weak form of encrypted data transfer between the AFS client and the AFS servers. This is often referred to as "fcrypt" mode. Encrypted data transfer can be turned on or off with the "fs setcrypt" command. Transitions between "crypt" and "non-crypt" modes are logged to the Windows Application Event Log.

3.15. Authenticated SMB Access to the OpenAFS Client Service

This section is maintained for historical reference and those sites that are manually configuring the OpenAFS Service to act as an SMB gateway. This section does not apply when the OpenAFS IFS redirector driver is in use.

OpenAFS authenticates SMB connections using either NTLM or GSS SPNEGO (NTLM). In previous versions of OpenAFS, the SMB connections were unauthenticated which opened the door for several attacks which could be used to obtain access to another user's tokens on shared machines.

When GSS SPNEGO attempts a Kerberos v5 authentication, the Windows SMB client will attempt to retrieve service tickets for "cifs/afs@REALM" (if the loopback adapter is in use) or "cifs/machine-afs@REALM" (if the loopback adapter is not being used). It is extremely important that this service principal not exist in the KDC database as the Kerberos authentication must fail allowing automatic fallback to NTLM. When NTLM is used a special local authentication mode will be used that does not require access to the user's password. Instead, Windows will internally recognize the request as coming from a local logon session.

It should also be noted that because Kerberos v5 authentication cannot be used, it is not possible to digitally sign the SMB communications. On systems that require Digital Signing of SMB Client connections, access to \\AFS will fail with a connection error.

3.16. IBM AFS INI Files Replaced By Windows Registry Keys

IBM AFS and OpenAFS 1.2 Windows clients stored configuration data in Windows .INI files. This OpenAFS client does not use Windows .INI files for the storage of configuration data. All settings are stored in the registry (see Appendix A). The CellServDB file is now stored in either the %ALLUSERSPROFILE%\Application Data\OpenAFS\Client directory (aka \ProgramData

\OpenAFS\Client on Vista\Win7\2008) or the %PROGRAMFILES%\OpenAFS\Client directory. The CellServDBDir [68] registry value or the AFSCONF environment variable can be used to specify an alternative location.

For users converting from IBM AFS clients, during installation OpenAFS will relocate the contents of the "afsdcell.ini" file to the new CellServDB file. OpenAFS will also import the contents of the "afs_freelance.ini" file to the Windows registry. OpenAFS will not process the contents of the "afsddbmt.ini".

3.17. Microsoft Windows Internet Connection Firewall

The OpenAFS Client is compatible with the Internet Connection Firewall that debuted with Windows XP SP2 and Windows 2003 SP1 and the Advanced Firewall that was introduced with Windows Vista. The Internet Connection Firewall will be automatically configured to allow the receipt of incoming callback messages from the AFS file server. In addition, if the OpenAFS Service is manually configured to behave as an SMB Gateway, the appropriate *Back Connection* registry entries are added to allow SMB authentication to be performed across the Microsoft Loopback Adapter.

3.18. Browsing AFS from the Explorer Shell and Office

The OpenAFS Client Service implements the CIFS Remote Admin Protocol and the Microsoft RPC SVRSVC and WKSSVC services which allows Explorer to browse server and share information. This significantly enhances the interoperability of AFS volumes within the Explorer Shell and Microsoft Office applications.

3.19. Byte Range Locking

Many applications on Windows (e.g. Microsoft Office) require the use of byte range locks applied to a file either to protect against simultaneous file access or as a signaling mechanism. OpenAFS for Windows release 1.5 (or greater) implements byte range locking within the CIFS-AFS gateway server. This support for byte range locking obtains AFS' advisory file server locks to simulate Microsoft Windows mandatory locks. When an application opens a file, a lock will be obtained from AFS indicating that the file is in use. If the lock is a write lock, access to the file will be restricted to other applications running on the same machine as the first application to request the lock. Applications running on other machines will see the AFS full file lock and will be unable to access the file.

Most Windows applications and Windows itself opens files in shared read mode. When this is done, a read lock is applied to the file. This does not prevent shared read access across multiple machines but is used to ensure that no one writes to the file while it is in use.

As the CIFS-AFS gateway server attempts to implement Windows lock semantics on top of AFS lock semantics it is important to understand how AFS file locks work. In Windows there are no special privileges associated with obtaining file locks. If you can read or execute a file, then you can obtain shared and exclusive locks. In general, a Windows shared lock equates to an AFS read lock and a Windows exclusive lock equates to an AFS write lock. In AFS if you can write to a file, then you can obtain a write lock. However, in AFS if you can read a file it does not mean that you can obtain a read lock on it. The ability to obtain read locks is granted only if you have the lock (or ‘k') privilege. This behavior is required in order to allow anonymous users to read files while preventing them from being able to deny access to the files

to other users. *OpenAFS 1.4.0 and earlier as well as all IBM AFS file servers have an implementation bug that prevents users with write privileges from being able to obtain locks without the lock privilege.* When AFS serves data out of read-only volumes the file server will deny all requests for read and write locks because the contents of the volume cannot be changed by the client.

Since Microsoft Windows applications almost always attempt to obtain a temporary exclusive lock when accessing files the OpenAFS Client's CIFS-AFS gateway implements the following semantics in order to reduce the inconvenience on end users.

- If the file is located on a read-only volume and the application requests a shared lock, the CIFS-AFS server will grant the lock request without asking the AFS file server.
- If the file is located on a read-only volume and the application opens the file with write access and requests an exclusive lock, the CIFS-AFS server will refuse the lock request and return a read only error.
- If the file is located on a read-only volume and the application opens the file with only read access and requests an exclusive lock, the CIFS-AFS server will fulfill the lock request with a read lock.
- If the file is located on a read-write volume and the application requests an exclusive lock, the CIFS-AFS server will request a write lock from the AFS file server. If granted by the file server, then the CIFS-AFS server will grant the lock request. If the request is denied due to an access denied error and the user has the lookup, read and lock privileges and the file was opened for read only access, then the CIFS-AFS server will request a read lock from the file server. If the request is denied due to an access denied error and the user has the lookup and read privileges but not the lock privilege, then the CIFS-AFS server will grant the request even though the AFS file server said ‘no'. If the user does not have at least those permissions, the CIFS-AFS server will deny the request.
- If the file is located on a read-write volume and the application requests a shared lock, the CIFS-AFS server will request a read lock from the AFS file server. If granted by the file server, then the CIFS-AFS server grants the lock request. If the request is denied due to an access denied error and the user has the lookup and read privileges but not the lock privilege, then the CIFS-AFS server will grant the request even though the AFS file server said ‘no'. If the user does not have at least those permissions, the CIFS-AFS server will deny the request.
- If multiple processes on the same machine attempt to access the same file simultaneously, the CIFS-AFS server will locally manage the granted locks and all processes will share a single lock on the AFS file server.
- If the CIFS-AFS server is unable to renew the AFS file server locks, then it will invalidate the associated file handles. This is the same behavior that an application will experience if it was using a Windows File Share and the connection was broken. Invalidating the file handles prevents subsequent data corruption from taking place.

If you wish to disable the acquisition of locks from the file server, this can be performed using the EnableServerLocks [63] registry value.

3.20. Automatic Discarding of AFS Tokens at Logoff

This section does not apply unless the OpenAFS Service is manually configured as an SMB Gateway.

The OpenAFS Client will automatically forget a user's tokens upon Logoff unless the user's profile was loaded from an AFS volume. In this situation there is no mechanism to determine when the profile has been successfully written back to the network. It is therefore unsafe to release the user's tokens. Whether or

not the profile has been loaded from the registry can be determined for Local Accounts, Active Directory accounts and NT4 accounts.

If there is a need to disable this functionality, the LogoffPreserveTokens [53] registry value can be used. (see Appendix A.)

3.21. Windows Terminal Server installations

The OpenAFS Servers should not be installed on a machine with Terminal Server installed. The OpenAFS Client is fully compatible with Terminal Services.

3.22. Hidden Dot Files

AFS is a UNIX native file system. The OpenAFS client attempts to treat the files stored in AFS as they would be on UNIX. File and directory names beginning with a "." are automatically given the Hidden attribute so they will not normally be displayed. This behavior can be altered via the HideDotFiles [57] registry value.

3.23. Status Cache Limits

The Status Cache (AFS Configuration Control Panel: Advanced Page) is defined to have a maximum number of entries. Each entry represents a single file or directory entry accessed within the AFS file system. When the maximum number of entries are allocated, entries will begin to be reused according to a least recently used (LRU) algorithm. If the number of files or directories being accessed repeatedly by your applications is greater then the maximum number of entries, your host will begin to experience thrashing of the Status Cache and all requests will result in network operations.

If you are experiencing poor performance try increasing the maximum number of Status Cache entries. Each entry requires approximately 1.2K. The default number of Status Cache entries is 10,000. This can be adjusted using the Stats [52] registry value.

3.24. NETBIOS over TCP/IP must be enabled

This section only applies when the IFS client mode is disabled.

"Netbios over TCP/IP" must be active on the machine in order for communication with the AFS Client Service to succeed. If "Netbios over TCP/IP" is disabled on the machine, then communication with the AFS Client Service will be impossible. If you are using the Microsoft Loopback Adapter, configure the "Netbios over TCP/IP" setting for the adapter.

3.25. OpenAFS binaries are digitally signed

The OpenAFS Client Service and related binaries distributed by OpenAFS.org are digitally signed by "Secure Endpoints Inc." or "Your File System Inc.". The OpenAFS Client Service will perform a run-time verification check to ensure that all OpenAFS related DLLs loaded by the service match the same file version number and were signed by the same entity. This check has been added to prevent the stability problems caused when more than one AFS installation is present on a machine simultaneously. Many hours of support time have been wasted tracking down problems caused by the mixture of files from different releases.

Appendix A documents the " VerifyServiceSignature [68]" registry value which can be used to disable the signature check. The file version check cannot be disabled.

3.26. Maximum Size of the AFSCache File

The maximum cache size on 32-bit Windows is approximately 1.2GB. This is the largest contiguous block of memory in the 2GB process address space which can be used for constructing a memory mapped file. Due to fragmentation of the process space caused by the loading of libraries required by the digital signature verification code, any attempt to specify a cache size greater then 700MB will result in the automatic disabling of the signature check. Significantly larger cache sizes can be used on 64-bit Windows.

On 32-bit systems that have Apple Bonjour 1.0.6 installed, the maximum cache size is further constrained due design flaw in the Apple mdnsNSP.dll which is injected into all processes that use network services. On these systems the maximum is approximately 512MB.

3.27. Filename Character Sets

This section describes functionality and concerns related to pre-1.5.50 releases of OpenAFS for Windows. This release stores all file names on the file servers as Unicode encoded using UTF-8.

OpenAFS for Windows implements an SMB server which is used as a gateway to the AFS filesystem. Because of limitations of the SMB implementation in pre-1.5.50 releases, Windows stored all files into AFS using OEM code pages such as CP437 (United States) or CP850 (Western Europe). These code pages are incompatible with the ISO Latin-1 or Unicode (UTF-8) character sets typically used as the default on UNIX systems in both the United States and Western Europe. Filenames stored by OpenAFS for Windows were therefore unreadable on UNIX systems if they include any of the following characters:

[Ç] 128 08/00 200 80 C cedilla

[ü] 129 08/01 201 81 u diaeresis

[é] 130 08/02 202 82 e acute

[â] 131 08/03 203 83 a circumflex

[ä] 132 08/04 204 84 a diaeresis

[à] 133 08/05 205 85 a grave

[å] 134 08/06 206 86 a ring

[ç] 135 08/07 207 87 c cedilla

[ê] 136 08/08 210 88 e circumflex

[ë] 137 08/09 211 89 e diaeresis

[è] 138 08/10 212 8A e grave

[ï] 139 08/11 213 8B i diaeresis

[î] 140 08/12 214 8C i circumflex

[ì] 141 08/13 215 8D i grave

[Ä] 142 08/14 216 8E A diaeresis

[Å] 143 08/15 217 8F A ring

[É] 144 09/00 220 90 E acute

[æ] 145 09/01 221 91 ae diphthong

[Æ] 146 09/02 222 92 AE diphthong

[ô] 147 09/03 223 93 o circumflex

[ö] 148 09/04 224 94 o diaeresis

[ò] 149 09/05 225 95 o grave

[û] 150 09/06 226 96 u circumflex

[ù] 151 09/07 227 97 u grave

[ÿ] 152 09/08 230 98 y diaeresis

[Ö] 153 09/09 231 99 O diaeresis

[Ü] 154 09/10 232 9A U diaeresis

[ø] 155 09/11 233 9B o slash

[£] 156 09/12 234 9C Pound sterling sign

[Ø] 157 09/13 235 9D O slash

[×] 158 09/14 236 9E Multiplication sign

[ƒ] 159 09/15 237 9F Florin sign

The pre-1.5.50 OpenAFS Client provided an optional registry value, StoreAnsiFilenames [69], that could be set to instruct OpenAFS to store filenames using the ANSI Code Page instead of the OEM Code Page. The ANSI Code Page is a compatible superset of Latin-1. This setting is not the default setting because making this change would prevent OpenAFS for Windows from being able to access filenames containing the above characters which were created without this setting.

All versions of OpenAFS for Windows 1.5.50 and above exchange file names with Microsoft Windows using the Unicode character set. All file names are read from and stored to AFS file servers using the UTF-8 encoding of Unicode. As a result the StoreAnsiFilenames [69] option is no longer supported.

3.28. Character Set Issues with Roaming Profiles

This section describes functionality and concerns related to pre-1.5.50 releases of OpenAFS for Windows. This release stores all file names on the file servers as Unicode encoded using UTF-8.

There is a known issue with storing Windows Roaming Profiles when the profile contains either directories or files with names which cannot be represented in the local OEM character set. In this case, attempts to write the profile back to AFS will fail during the character set conversion. The pre-1.5.50 OpenAFS Client's CIFS gateway did not support UNICODE. To avoid this problem some sites run custom logoff scripts (assigned by group policy) which rename all files to use only the supported characters for the locale.

Versions of OpenAFS for Windows 1.5.50 and above do not suffer from these issues.

3.29. The AFSCache File

The AFS Cache file is stored by default at %TEMP%\AFSCache in a persistent file marked with the Hidden and System attributes. The persistent nature of the data stored in the cache file improves the performance of OpenAFS by reducing the number of times data must be read from the AFS file servers.

The performance of the AFS Client Service is significantly affected by the access times associated with the AFSCache paging file. When given the choice, the AFSCache file should be placed on a fast disk, preferably formatted NTFS and using a Solid State Disk, the file should not be compressed and should consist of as few fragments as possible. Significant performance gains can be achieved by defragmenting the AFSCache file with SysInternal's Contig utility while the AFS Client Service is stopped.

3.30. Restricting OpenAFS Client Service Start and Stop

A command line tool, afsdacl.exe, can be used to restrict the ability to start and stop the OpenAFS Client Service.

```
afsdacl : Set or reset the DACL to allow starting or stopping
          the afsd service by any ordinary user.

Usage: afsdacl [-set | -reset] [-show]
       -set   : Sets the DACL
       -reset : Reset the DACL
       -show  : Show current DACL (SDSF)
```

3.31. The @sys Name List

The default @sys name list in the OpenAFS Client is set to "x86_win32 i386_w2k i386_nt40" for 32-bit x86 systems. The default is "amd64_win64" for amd 64-bit versions of Windows.

The IFS redirector driver is aware of the process type. On 64-bit systems, there are two @sys name lists "SysName" which is used for the WOW64 environment and "SysName64" which is used for the 64-bit environment. If "SysName64" is not provided, "SysName" is used for all processes.

3.32. Symlinks to AFS UNC Paths

In OpenAFS, symlinks to AFS UNC paths, \\AFS[\all]\..., are treated the same as symlinks to /afs/... However, please use /afs/... as the Windows UNC form will not work on UNIX client.

The *symlink make* command will automatically translate \\AFS\... to /afs/... for you.

3.33. Cache Manager Debugging

The OpenAFS Client implements the Cache Manager Debugging RPC Interface. The CM debugger can be queried with cmdebug.exe [http://docs.openafs.org/Reference/1/cmdebug.html].

```
Usage: cmdebug -servers <server machine> [-port <IP port>] [-long] [-refcounts
```

```
                  [-callbacks] [-ctime] [-addrs] [-cache] [-cellservdb] [-help]
Where: -long        print all info
       -refcounts   print only cache entries with positive reference counts
       -callbacks   print only cache entries with callbacks
       -ctime       print human readable expiration time
       -addrs       print only host interfaces
       -cache       print only cache configuration
       -cellservdb  print only cellservdb info
```

3.34. Windows Logon Caching vs. Kerberos Logons

If you are a site which utilizes MIT/Heimdal Kerberos principals to logon to Windows via a cross-realm relationship with a multi-domain Windows forest, you must enable Windows logon caching unless the workstation is Windows Vista or Windows 7.

3.35. Initial Server Preferences

VLDB and File Server Preferences can now be provided initial values using registry keys. This is useful for managed machines in a Windows domain which are centrally located (e.g., in a computing lab.) See Appendix A for details on the " Server Preferences" keys.

3.36. File Timestamps and Daylight Saving Time

The OpenAFS Client reports timestamps on files stored in AFS in UTC all year round. In locales with daylight savings time, previous versions of AFS for Windows reported the time when DST is active as UTC+1. This was done to preserve the relative local time for the user. A file stored at 11:00am EST in January would be reported as having been stored at 11:00am EDT in June. Unfortunately, this has the negative side effect of changing the reported timestamp from 16:00UTC to 15:00UTC. Since Windows treats all file times in UTC, data synchronization applications which rely on the timestamp would believe that all files stored in AFS had changed.

It should be noted that UNIX based operating systems (such as Solaris) do not appear to report file times to applications in UTC. They do preserve the relative local time. This may confuse some users who are used to being able to compare the timestamp in an UNIX shell with the timestamp from the Windows explorer. During DST, these two times will no longer agree even though they are in fact representing the same moment in time.

3.37. Windows RPC client support must be installed

If the installer refuses to install and complains about an RPC configuration error, check to ensure that the following registry entries are present and that they refer to the dll "rpcrt4.dll":

HKLM "SOFTWARE\Microsoft\RPC\ClientProtocols" "ncacn_np"

HKLM "SOFTWARE\Microsoft\RPC\ClientProtocols" "ncacn_ip_tcp"

HKLM "SOFTWARE\Microsoft\RPC\ClientProtocols" "ncadg_ip_udp"

HKLM "SOFTWARE\Microsoft\RPC\ClientProtocols" "ncacn_http"

3.38. Generating Minidumps of the OpenAFS Client Service

OpenAFS 1.4 added a new command, "fs minidump". This command can be used at any time to generate a mini dump file containing the current stack of the afsd_service.exe process. This output can be very helpful when debugging the AFS Client Service when it is unresponsive to SMB/CIFS requests.

3.39. AFS Client Universally Unique Identifiers (UUIDs) vs. System Cloning or Virtual Machine Cloning

The OpenAFS Client implements Universally Unique Identifiers (UUIDs). They are used to provide the AFS file server with a method of identifying the client that is independent of IP address. This permits the AFS file server to track mobile clients or those behind Network Address Translators when they move from address to address or port to port. Tracking the client improves client performance by permitting callback state to be maintained across location changes. The UUID is generated when the AFSCache file is created and is maintained as long as the contents of the AFSCache file are valid. The UUID is stored in the AFSCache file.

When cloning machines that have Windows AFS client installed it is necessary to generate a new UUID for each client. This will be done automatically if the Windows Machine SID is re-generated using Microsoft SysPrep. If the SID is not being re-generated either the AFSCache file should be deleted or the command *fs uuid -generate* must be executed after the the clone is created. **Multiple AFS clients reporting the same UUID will not only result in horrible AFS client performance and cache inconsistencies, but they will also put a tremendous strain on the AFS file servers.**

For lab environments that wish to erase all cached data on each restart, the NonPersistentCaching [54] option will disable the use of the persistent cache file. As a side effect, a new UUID will be generated for the AFS client service on each restart.

[SMB only] When a Windows system is cloned, the Microsoft Loopback Adapter will be disabled in the cloned system. Administrators must re-install the Microsoft Loopback Adapter within the cloned environment. This can be automated by using the OpenAFS " *instloop.exe – i*" command. Instloop.exe can be extracted from the MSI installer by performing an administrative install via *msiexec.exe /a*.

3.40. Delayed Write Errors with Microsoft Office Applications

This section does not apply unless the OpenAFS Service is manually configured as an SMB Gateway.

Microsoft Office makes heavy use of asynchronous input/output methods for reading and writing to file streams. This can result in hundreds of requests being simultaneously queued for service by the CIFS client with a fixed timeout period. As the AFS CIFS server is local to the machine the Windows CIFS client believes that it can respond almost instantaneously to write requests as the actual writing to the AFS file server is performed by a background daemon thread. When the actual network bandwidth to the AFS file

server is slow and the file size is large it is possible for the CIFS client to time out the connection. When this happens a "delayed write error" will be reported to the user and the application may crash. The only workaround at the current time is to save first to a local disk and subsequently copy the file to AFS as copying a file with the explorer shell does not use asynchronous i/o.

The CIFS session timeout defaults to 45 seconds and can be increased by modifying the ConnDeadTimeout registry value [59].

Beginning with the 1.5.33 release, the performance characteristics of SMB Write Data operations can be adjusted. In prior releases all writes were performed using a restricted asynchronous store model in which only one asynchronous store operation per file can be performed at a time. The reason for this restriction is limit the amount of data the cache manager will accept without it having been written to the file server. If too much unwritten data is accepted, the file close operation will block until all of the unwritten data is output and this could trigger a CIFS client disconnect.

Prior to 1.5.33 the size of the asynchronous store was always equal to the chunk size which was often too large for low bandwidth connections. The asynchronous store size now defaults to 32KB and is configurable using the SMBAsyncStoreSize [69] registry value. Asynchronous store operations can also be disabled using the EnableSMBAsyncStore [69] registry value in which case all writes received by the cache manager block until the Rx StoreData operation completes.

During the first quarter of 2009 Microsoft added new functionality to the SMB Redirector which permits an extended timeout period to be used for an enumerated list of Netbios server names. This functionality was distributed in Service Pack 2 for Vista and 2008 and is incorporated into the RTM releases of Windows 7 and Server 2008 R2. Updates to Windows XP (KB916204), XP64 (KB969289), and Server 2003 (KB969289) were made available as hot fixes. When this support is available, the OpenAFS for Windows client 1.5.61 and above will raise the timeout period from 45 seconds to 10 minutes.

3.41. Global Drives (aka Service Drive Letters) are no longer supported by Microsoft

The Global Drive auto-mount feature has been deprecated due to the following Microsoft KB article.

http://msdn.microsoft.com/library/en-us/dllproc/base/services_and_redirected_drives.asp

The article says that services mounting drive letters are no longer supported by Microsoft and may act unpredictably. The experience other users have had is that if the connection to the OpenAFS CIFS/SMB server is terminated by the Windows CIFS client, the drive mapping may not be re-established until the machine is rebooted.

OpenAFS supports UNC paths and whenever possible applications should be modified to use UNC form \\AFS\<cellname>\<path> instead of drive letters.

Another problem with service mounted drive letters is that the drives are reported as local disk devices and cannot be resolved as being mapped to the \\AFS name space. As a result, AFS path ioctl operations will fail. The fs.exe and symlink.exe command line tools and the AFS Explorer Shell extension will not operate on service mounted drive letters.

3.42. 64-bit Microsoft Windows Installations

Although 64-bit Windows platforms support both 64-bit and 32-bit applications, the OpenAFS Service installed on the machine must be 64-bit. The 64-bit installer contains only 64-bit executables. In order to support 32-bit applications it is required that a separate 32-bit OpenAFS Tools set be installed. This is

especially true when the IFS redirector is in use as the 32-bit Network Provider DLL must be installed in order for 32-bit applications to access drive letters mapped to \\AFS.

OpenAFS on 64-bit Windows benefits from the lifting of the 2GB process memory restriction that is present in 32-bit Windows. Without this restriction the AFS Cache File can become arbitrarily large limited only by available disk space.

3.43. Known Issues with Microsoft Windows Vista, Windows 7, Server 2008 [R2], Windows 8 and Server 2012

Windows Vista, Windows 7, and Server 2008 [R2] implement User Account Control [http://www.microsoft.com/technet/windowsvista/library/0d75f774-8514-4c9e-ac08-4c21f5c6c2d9.mspx] (UAC), a new security feature that implements least user privilege. With UAC, applications only run with the minimum required privileges. Even Administrator accounts run applications without the "Administrator" access control credentials. One side effect of this is that existing applications that mix user and system configuration capabilities must be re-written to separate those functions that require "Administrator" privileges into a separate process space. Future updates to OpenAFS will incorporate the necessary privilege separation, until that time some functions such as the Start and Stop Service features of the AFS Authentication Tool and the AFS Control Panel will not work unless they are "Run as Administrator". When a Vista user account that is a member of the "Administrators" group is used to access the AFS Control Panel (afs_config.exe), the process must be "Run as Administrator". Otherwise, attempts to modify the OpenAFS configuration will appear to succeed but in reality will have failed due to Vista's system file and registry virtualization feature.

The help files provided with OpenAFS are in .HLP format. Windows Vista, Windows 7, Server 2008 [R2], Windows 8 and Server 2012 do not include a help engine for this format. [http://support.microsoft.com/kb/917607]

The following items only apply when the OpenAFS Service is manually configured as an SMB Gateway.

OpenAFS for Windows works with Microsoft Windows Vista, Windows 7 and Server 2008 [R2] from both the command prompt and the Explorer Shell. When performing an upgrade from earlier versions of Microsoft Windows the Microsoft Loopback Adapter (MSLA) will be uninstalled. OpenAFS should be re-installed after the Windows Upgrade installation to restore the MSLA configuration.

Due to a feature change in Windows Vista's Plug-n-Play network stack, during a standby/hibernate operation the MSLA is disabled just as any other hardware device would be. This causes the OpenAFS Client's network binding to be lost. As a result, it takes anywhere from 30 to 90 seconds after the operating system is resumed for access to the OpenAFS Client and the AFS file name space to be restored. Until the network bindings have been re-established, ticket managers and other tools will report that the "AFS Client Service may not have been started".

3.44. AFS Share Name Syntax Provides Direct Access to Volumes

Starting with the 1.5.21 release of OpenAFS for Windows, the following syntax can be used to access any volume in any cell without requiring the creation of a mount point.

\\AFS\<cell><mount point type><volume>\

Where <cell> can be either a full cell name or an unambiguous prefix, the <mount point type> is '#' for a normal mount point or '%' to force the use of a read-write volume, and <volume> is either a volume name or its ID number.

Examples include:

\\AFS\athena.mit.edu#user.jaltman\

\\AFS\athena%user.jaltman\

\\AFS\athena.mit.edu# 537235559\

3.45. Differences between Windows and UNIX *fs examine*

The OpenAFS for Windows version of "fs examine" provide two additional lines of output when compared to the UNIX implementation. These lines include the owner and group information for the file as well as the volume status. The Windows output will also indicate the type of object {File, Directory, Mountpoint, Symlink, ...} that was examined.

[C:\]fs examine \\afs\athena#user.jaltman

Directory \\afs\athena#user.jaltman (537235559.1.1) contained in cell athena.mit.edu

Owner jaltman (28180) Group jaltman (28180)

Volume status for vid = 537235559 named user.jaltman is

Current disk quota is 1500000

Current blocks used are 1244184

The partition has 151945877 blocks available out of 511163724

Volume is online

The object owner and group and UNIX mode information is not available on Microsoft Windows via any other method.

To set the owner use *fs chown -owner <user name or id> [-path <dir/file path>+] [-literal]*

To set the group use *fs chgrp -group <user name or id> [-path <dir/file path>+] [-literal]*

To set the UNIX mode use *fs chmod -mode <UNIX mode bits> [-path <dir/file path>+] [-literal]*

3.46. Literal evaluation of AFS Symlink and MountPoint objects via fs commands

Beginning with the 1.5.31 release, the fs commands *examine*, *flush*, *getuseraccess*, *whereis*, and *whichcell* provide a new command-line parameter, *-literal*. When specified, if the evaluated object is a symlink or a mountpoint the resulting output will describe the specified object and not the target of the symlink or mountpoint.

3.47. Out of Quota errors

Prior to the 1.5.31 release, out of quota errors were reported to the calling application as an out of space error. As of 1.5.31, an out of space error will indicate that the partition on which the volume is located is in fact out of space. Whereas an out of quota error indicates that the user does not have permission to allocate additional space.

3.48. Linked Cells

The 1.5.55 release adds support for linked cells as implemented in the Unix OpenAFS client. When two cells are linked, a volume lookup in one cell that fails is retried in the linked cell. This functionality can be used to implement:

- a test cell which provides substitutes for a subset of the volumes in the linked production cell
- a cell renaming
- a cell splitting
- a cell merger

Two cells are linked in the CellServDB file:

```
>cell-one cell-two #Description
...
>cell-two cell-one #Description
...
```

aklog and Network Identity Manager will automatically obtain tokens for the linked cell when tokens for the other cell is specified.

3.49 Registry Alternative to CellServDB File

Beginning with the 1.5.60 release, the [HKLM\SOFTWARE\OpenAFS\Client\CellServDB] registry key can be used to distribute Volume Database Server location information either as a supplement to the *CellServDB file* or as a substitute for it. The precedence order for lookups is: Registry, File, and then DNS.

3.50 Release Notes Converted to Windows HTML Help

Starting with the 1.5.60 release, this document, the OpenAFS Administrator Guide and the OpenAFS User Guide are provided in HTML Help format instead of raw HTML pages.

3.51. Support for Microsoft RPC Services: WKSSVC and SRVSVC

Beginning with the 1.5.62 release, the OpenAFS client supports named pipes and the Microsoft RPC Services WKSSVC and SRVSVC. This permits a significantly improved Netbios Server browsing

experience with both the *NET VIEW \\AFS* command and the Explorer Shell. No longer will Windows display truncated cell names as available network shares. The network share properties will also include the object type and and the target of symlinks and mount points.

3.52. Differences between Windows and UNIX *fs newcell*

The OpenAFS for Windows version of "fs newcell" prior to 1.5.74 behaved quite differently than its UNIX counterpart. Instead of adding cell server information for a new cell, the command simply caused the cache manager to destroy all of its cell server information and then reload it the next time the server list for a cell is needed. The UNIX version explicitly replaces the server list for a cell with a new list.

Beginning with the 1.5.75 release, the Windows version will continue to behave as prior versions did when no parameters are specified but will accept an extended UNIX command-line syntax as well. In addition to the UNIX parameters, the Windows "fs newcell" command accepts four new ones:

- [-fsport <cell's fileserver port>]

  Specifies an alternate port number at which the cell's file servers are listening

- [-vlport <cell's vldb server port>]

  Specifies an alternate port number at which the cell's volume location database servers are listening

- [-registry]

  Instructs the cache manager to save the cell server information to the registry database

- [-dns]

  Indicates that the cell server information should be obtained via DNS SRV or DNS AFSDB records

3.53. AFS Mount Points and Symlinks are Reparse Points

The AFS redirector driver represents all AFS mount points and AFS symlinks as reparse points within the file system name space using a Microsoft assigned tag value. Tools that are OpenAFS reparse point aware can create, query and remove AFS symlinks and mount points without requiring knowledge of AFS pioctls. The explorer shell will be able to delete a mount point or symlink as part of a recursive directory tree removal without crossing into the reparse point target.

The Explorer Shell displays Symlinks and Mount Points using overlay icons.

Beginning with 1.7.22, AFS Symlinks are represented as Microsoft Symlink reparse points instead of an OpenAFS specific reparse point. Symlinks can now be created using the Win32 CreateSymbolicLink [http://msdn.microsoft.com/en-us/library/windows/desktop/aa363866%28v=vs.85%29.aspx] API and

follow all of the behaviors of Microsoft Windows' Symbolic Links [http://msdn.microsoft.com/en-us/library/windows/desktop/aa365680%28v=vs.85%29.aspx]. Any tool capable of creating symbolic links on NTFS can now do so within AFS.

Symbolic Links to Files are not supported by all Microsoft Windows applications because directory enumerations, GetFileAttributes and GetFileAttributesEx return the attributes and size of the Symbolic Link and not that of the target file. Applications that treat the size of the Symbolic Link as the size of the target file will misbehave. All Java releases 1.6.x and earlier and all .NET applications as of this writing use the Symbolic Link size as the size of the target file. Java 1.7 correctly processes Symbolic Links.

3.54. AFS Authentication Groups

When the OpenAFS Service is configured as an SMB Gateway, all AFS Tokens are associated with Windows user names. With the IFS redirector driver, tokens are associated with Authentication Groups. By default, an authentication group is allocated for each User SID and Logon Session Id combination. In addition, it is possible for processes to create additional Authentication Groups. Each thread in a process can select an Authentication Group within the process as the active Authentication Group.

One of the significant benefits of Authentication Groups within the Windows environment is that Windows services (svchost.exe, csrss.exe, etc.) which impersonate user processes will seamlessly gain access to the user's AFS credentials for the lifetime of the impersonation.

3.55. Known IFS redirector driver limitations

The following is a list of known issues when using the IFS redirector driver:

- Adobe Reader Protected Mode prevents saving PDF documents to AFS.

  In Acrobat Reader 9.3.2 Adobe added a new security feature "Protected Mode" which is enabled by default. Protected mode runs AcroRd32.exe in a sandbox and prevents undesirable network access. The release notes with all versions of Reader since 9.3.2 indicate that DFS and NFS network paths are inaccessible when Protected Mode is on.

- Command Prompt .LNK files do not behave properly when stored within AFS

  - Custom properties will be ignored.

  - It is not possible to make changes to the LNK properties.

  These issues are the result of the *Console Window Host* process (conhost.exe) running outside the logon session's authentication group. While conhost.exe impersonates the Windows user, it does not impersonate a logon session process. As a result, it has no tokens and cannot access the LNK file.

- The Windows File System Volume Query Quota Interface [http://msdn.microsoft.com/en-us/library/ff549293%28v=vs.85%29.aspx] is not implemented. As a result, AFS quota information is not available to application processes or end users via Windows dialogs.

- The Windows Volume Shadow Copy Service [https://secure.wikimedia.org/wikipedia/en/wiki/Shadow_Copy] is not implemented. As a result, AFS backup volumes are not accessible via the Explorer Shell.

- There is no support for storing DOS attributes such as Hidden, System, or Archive.

- There is no support for Alternate Data Streams as required by Windows User Account Control to store Zone Identity data.

- There is no support for Extended Attributes.
- There is no support for Access Based Enumeration [https://blogs.technet.com/b/hugofe/archive/2010/06/21/windows-2008-access-based-enumeration-abe.aspx].
- There is no support for Windows Management Instrumentation [https://secure.wikimedia.org/wikipedia/en/wiki/Windows_Management_Instrumentation]
- There is no support for Distributed Link Tracking and Object Identifiers [http://msdn.microsoft.com/en-us/library/aa363997%28v=vs.85%29.aspx]
- There is no support for storing Windows Access Control Lists [http://msdn.microsoft.com/en-us/magazine/cc982153.aspx]. Only the AFS ACLs are enforced.
- There is a bug in the Explorer Shell which can result in the Shell responding to a Ctrl-V (Paste) operation with an out of space error. The bug is that the Shell queries the root directory of the UNC Path or Drive Letter for free space instead of the path in which the Paste is being performed. To work around the bug, select a directory in another volume under the same root and then return to the target directory before initiating the Paste. Performing the Paste using the Context Menu instead of Ctrl-V will avoid triggering the bug.
- Windows file systems support a maximum of 31 reparse points (mount points or symbolic links) within a path.

3.56. Changes for Windows 8 and Server 2012

In Windows 8 and Server 2012 Microsoft has introduced a new file system, ReFS, and has begun the process of transitioning away from several legacy file system properties including 8.3 compatible short names for all file system objects. The OpenAFS file system has followed suit and is disabling automatic generation of 8.3 compatible names on Windows 8 and Server 2012.

3.57. The Explorer Shell, Drive Letter Mappings, and Read Only Volumes

File systems can expose a variety of information about the underlying volumes they serve to applications. All AFS volumes are described as supporting Case Preservation, Hard Links, Reparse Points and Unicode characters. For .readonly volumes the file system can indicate that the volume is a Read Only Volume. The benefit of doing so is that applications such as the Explorer Shell can alter their behavior to improve the user experience. For example, when the volume is reported as read-only the Explorer Shell can remove the Rename, Delete, and other file modifying operations from the user interface. Unfortunately, the Windows 7 Explorer Shell is broken with regards to Volume Information queries when issued on Network Mapped Drive Letters. Instead of performing a volume information query on the current directory, the Explorer Shell only queries the root directory of the mapped drive letter. As a result, if the drive letter is mapped to a .readonly volume, all paths accessed via the drive letter are considered to be read-only even when they are not. This behavior is fixed in Windows 8 and Server 2012.

Due to this bug, OpenAFS on Windows 7 and below does not report the FILE_READ_ONLY_VOLUME [http://msdn.microsoft.com/en-us/library/windows/desktop/aa964920%28v=vs.85%29.aspx] flag as part of the volume information. The Explorer Shell properly queries the volume information for UNC paths. If network mapped drive letters are not used, it is often convenient if the FILE_READ_ONLY_VOLUME flag is reported. This can be configured using the VolumeInfoReadOnlyFlag registry value.

Chapter 4. How to Troubleshoot Problems with OpenAFS for Windows

OpenAFS for Windows provides a wide range of tools to assist you in debugging problems. The techniques available to you are varied because of the wide range of issues that have been discovered over the years.

4.1. pioctl debugging (IoctlDebug [68] registry key)

pioctl (path-based ioctl) calls are used by various tools to communicate with the AFS Client Service. Some of the operations performed include:

- setting/querying tokens (tokens.exe, aklog.exe, afscreds.exe)
- setting/querying ACLs
- setting/querying cache parameters
- flushing files or volumes
- setting/querying server preferences
- querying path location
- checking the status of servers and volumes
- setting/querying the sysname list

pioctl calls are implemented by writing to a special UNC path that is processed by the AFS Client Service. If there is a failure to communicate with the AFS Client Service via SMB/CIFS, it will be impossible to perform any of the above operations.

To assist in debugging these problems, the registry value:

[HKLM\SOFTWARE\OpenAFS\Client]

REG_DWORD: IoctlDebug = 0x01

should be set. Then any of the commands that perform pioctl calls should be executed from the command prompt. With this key set the pioctl library will generate debugging output to stderr. The output will contain the Win32 API calls executed along with their most important parameters and their return code. The MSDN Library and the Microsoft KnowledgeBase can be used as a reference to help you determine the configuration probem with your system.

4.2. afsd_service initialization log (%WinDir% \TEMP\afsd_init.log)

Every time the AFS Client Service starts it appends data about its progress and configuration to a file. This file provides information crucial to determining why the service cannot start when there are problems.

When the process terminates due to a panic condition it will write to this file the source code file and line number of the error. In many cases the panic condition is due to a misconfiguration of the machine. In other cases it might be due to a programming error in the software. A quick review of the location in the source code will quickly reveal the reason for the termination.

The MaxLogSize [62] registry value determines the maximum size of the %WINDIR%\TEMP \afsd_init.log file. If the file is larger than this value when OpenAFS Client Service starts, the file will be reset to 0 bytes. If value is set to 0, the file will be allowed to grow indefinitely.

4.3. afsd_service debug logs (fs trace {-on, -off, -dump} ->%WinDir%\TEMP\afsd.log)

When attempting to debug the behavior of the SMB/CIFS Server and the Cache Manager it is often useful to examine a log of the operations being performed. While running the AFS Client Service keeps an in memory log of many of its actions. The default number of actions preserved at any one time is 5000. This can be adjusted with the TraceBufferSize registry value [55]:

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

REG_DWORD TraceBufferSize

A restart of the service is necessary when adjusting this value. Execute "fs trace -on -reset" to begin the logging and "fs trace -dump" to output the contents of the log to the file.

4.4. Using SysInternal's Debug Viewer, Process Monitor, Process Explorer and Process Dump Tools

An alternatve option to the use of "fs trace -dump" to capture internal OpenAFS Client Service events is to use a tool such as Sysinternal's Debug Viewer [http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx] to capture real-time debugging output. When the OpenAFS Client Service starts and Bit 2 of the TraceOption [60] value in the registry is set, all trace log events are output using the Windows Debug Monitor interface (OutputDebugString).

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

REG_DWORD TraceOption = 0x04

Use "fs trace –on" and "fs trace –off" to toggle the generation of log messages.

Sysinternal's Process Monitor [http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx] can be use to monitor the file operations requested by applications and their success or failure.

In Process Monitor, set a filter to include only events on file paths that refer to the AFS name space. Be sure to include both the UNC path as well as any drive letters mapped to AFS.

Turn on the *Clock Time* and *Show Milliseconds* options in both tools to make it easier to synchronize the application requests and the resulting OpenAFS Client Service operations. The captured data can be stored to files for inclusion in bug reports.

Sysinternal's Process Explorer [http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx] is a replacement for the Windows Task Manager and so much more. Process Explorer can be configured to

use the DbgHelp.dll from " Microsoft Debugging Tools for Windows [http://www.microsoft.com/whdc/devtools/debugging/default.mspx]" as well as the debug symbols shipped as an optional component of the OpenAFS for Windows installer. (Options->Configure Symbols) Once configured the "Threads" tab of the process properties dialog will permit the viewing of a fully documented stack for each displayed thread. Hint: If there is a deadlock in the cache manager, two or more of the threads will be stuck in a call to osi_TWait().

4.5. Creating Microsoft MiniDumps (fs minidump -> %WinDir%\TEMP\afsd.dmp)

If the AFS Client Service become unresponsive to any form of communication there may be a serious error that can only be debugged by someone with access to the source code and a debugger. The "fs minidump" command can be used to force the generation of a MiniDump file containing the state of all of the threads in the AFS Client Service process. The most accurate MiniDump files will be produced after installing " Microsoft Debugging Tools for Windows [http://www.microsoft.com/whdc/devtools/debugging/default.mspx]".

The MiniDumpType [68] registry value can be used to adjust the scope of the process information included within the dump file. By default the MiniDump only contains the stacks of all threads and the values of all global variables. A much more useful MiniDump is one that contains the contents of the heap. Be warned, a MiniDump with heap will be as large as the cache file. In addition, it will include all of the data stored within the cache. If there are privacy concerns, do not produce a MiniDump with heap.

4.6. Single Sign-on (Integrated Logon) debugging

If you are having trouble with the Integrated Logon operations it is often useful to be able to obtain a log of what it is attempting to do. Setting the Debug registry value:

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

REG_DWORD Debug = 0x01

will instruct the Integrated Logon Network Provider and Event Handlers to log information to the Windows Event Log: Application under the name "AFS Logon".

4.7. RX (AFS RPC) debugging (rxdebug)

The rxdebug.exe tool can be used to query a variety of information about the AFS services installed on a given machine. The port for the AFS Cache Manager is 7001.

```
Usage: rxdebug -servers <server machine> [-port <IP port>] [-nodally]
       [-allconnections] [-rxstats] [-onlyserver] [-onlyclient]
       [-onlyport <show only <port>>]
       [-onlyhost <show only <host>>]
       [-onlyauth <show only <auth level>>] [-version]
       [-noconns] [-peers] [-help]
Where: -nodally         don't show dallying conns
       -allconnections  don't filter out uninteresting connections
```

```
-rxstats          show Rx statistics
-onlyserver       only show server conns
-onlyclient       only show client conns
-version          show AFS version id
-noconns          show no connections
-peers            show peers
```

4.8. Cache Manager RPC debugging (cmdebug)

The cmdebug.exe tool can be used to query the state of the AFS Cache Manager over the network.

```
Usage: cmdebug -servers <server machine> [-port <IP port>] [-long] [-refcounts
       [-callbacks] [-ctime] [-addrs] [-cache] [-cellservdb] [-help]
Where: -long         print all info
       -refcounts    print only cache entries with positive reference counts
       -callbacks    print only cache entries with callbacks
       -ctime        print human readable expiration time
       -addrs        print only host interfaces
       -cache        print only cache configuration
       -cellservdb   print only cellservdb info
```

4.9. Persistent Cache consistency check

The persistent cache is stored in a Hidden System file at %WinDir%\TEMP\AFSCache. If there is a problem with the persistent cache that prevent the AFS Client Service from being able to start a validation check on the file can be performed.

afsd_service.exe --validate-cache <cache-path>

4.10. Token Acquisition Debugging

If you are having trouble obtaining tokens with the Network Identity Manager AFS credential provider, it is recommended that you verify your ability to obtain tokens using the command-line tools *klog.exe* (if you are using kaserver) or *kinit.exe* and *aklog.exe* (if you are using Kerberos v5.) The aklog.exe *–d* option will be quite helpful in diagnosing Kerberos v5 related problems.

Chapter 5. Reporting Bugs

Bug reports should be sent to openafs-bugs@openafs.org [mailto:openafs-bugs@openafs.org?subject=Bug%20Report]. Please include as much information as possible about the issue. If you are reporting a crash, please install the debugging symbols by re-running the installer. If a dump file is available for the problem, %WINDIR%\TEMP\afsd.dmp, include it along with the AFS Client Trace file %WINDIR%\TEMP\afsd.log. The AFS Client startup log is %WINDIR%\TEMP\afsd_init.log. Send the last continuous block of log information from this file.

Configuring DrWatson to generate dump files for crashes:

1. Run drwtsn32.exe to configure or to identify where the log and the crash dump files are created:
2. click Start > Run...
3. type drwtsn32 <enter>.
4. Select either a Crash Dump Type: Mini or Full.
5. Clear Dump Symbol Table
6. Clear Append to Existing Log file.
7. Check Dump All Thread Contexts.
8. Check Create Crash Dump File
9. Next run the monitoring module of Dr. Watson:
10. click Start > Run...
11. type drwatson <enter>.
12. Once a crash happens, Dr. Watson generates a dump file and a report in the log file, including the address of the crash and the stack dump.

Once you have the Dr. Watson's logfile and minidump, zip them and attach them to your e-mail.

When reporting a error, please be sure to include the version of OpenAFS.

Chapter 6. How to Contribute to the Development of OpenAFS for Windows

Contributions to the development of OpenAFS for Windows are continuously needed. Contributions may take many forms including cash donations, support contracts, donated developer time, and even donated tech writer time.

6.1. The USENIX OpenAFS Fund

USENIX [http://www.usenix.org/], a 501c3 non-profit corporation, has formed the USENIX OpenAFS Fund in order to accept tax deductible donations on behalf of the OpenAFS Elders. The donated funds will be allocated by the OpenAFS Elders to fund OpenAFS development, documentation, project management, and maintaining openafs.org.

```
USENIX OpenAFS Fund
USENIX Association
2560 Ninth St., Suite 215
Berkeley, CA 94710
```

Donations can be made by sending a check, drawn on a U.S. bank, made out to the USENIX OpenAFS Fund or by making a donation online [https://db.usenix.org/cgi-bin/openafs/openafs.cgi].

6.2. Secure Endpoints Inc.

Secure Endpoints Inc. [http://www.secure-endpoints.com/] provides development and support services for OpenAFS for Windows and MIT Kerberos for Windows [http://web.mit.edu/kerberos/]. Donations provided to Secure Endpoints Inc. for the development of OpenAFS are used to cover the OpenAFS gatekeeper responsibilities; providing support to the OpenAFS community via the OpenAFS mailing lists; and furthering development of desired features that are either too small to be financed by development contracts.

Secure Endpoints Inc. accepts software development agreements from organizations who wish to fund a well-defined set of bug fixes or new features.

Secure Endpoints Inc. provides contract based support for the OpenAFS for Windows and the MIT Kerberos for Windows [http://web.mit.edu/kerberos/] products.

6.3. Your File System Inc.

Your File System Inc. [http://www.your-file-system.com/] provides development and support services for OpenAFS for Windows. Donations provided to Your File System Inc. for the development of OpenAFS are used to cover the OpenAFS gatekeeper responsibilities; providing support to the OpenAFS community via the OpenAFS mailing lists; and furthering development of desired features that are either too small to be financed by development contracts.

Your File System Inc. accepts software development agreements from organizations who wish to fund a well-defined set of bug fixes or new features.

Your File System Inc. provides contract based support for OpenAFS on all platforms.

6.4. Direct contributions of code and/or documentation

Organizations that use OpenAFS in house and have development staffs are encouraged to contribute code and documentation modifications to OpenAFS.org via >*http://gerrit.openafs.org/*.

6.5. OpenAFS for Windows Mailing Lists

If you wish to participate in OpenAFS for Windows development, please join the openafs-win32-devel@openafs.org [mailto:openafs-win32-devel@openafs.org?subject=OpenAFS%20for%20Windows%20Development%20Contribution] mailing list.

https://lists.openafs.org/mailman/listinfo/openafs-win32-devel

User questions should be sent to the openafs-info@openafs.org [mailto:openafs-info@openafs.org?subject=OpenAFS%20for%20Windows%20User%20Question] mailing list.

https://lists.openafs.org/mailman/listinfo/openafs-info

You must join the mailing lists if you wish to post to the list without incurring a moderation delay.

Chapter 7. MSI Deployment Guide

7.1. Introduction

A MSI installer option is available for those who wish to use Windows Installer for installing OpenAFS and for organizations that wish to deploy OpenAFS through Group Policy. The first version of OpenAFS for Windows available as an MSI was 1.3.65.

This document provides a guide for authoring transforms used to customize the MSI package for a particular organization. Although many settings can be deployed via transforms, in an Active Directory environment it is advisable to deploy registry settings and configuration files through group policy and/or startup scripts so that machines where OpenAFS for Windows is already installed will pick up these customizations.

7.1.1 Requirements

The information in this document applies to MSI packages distributed with OpenAFS for Windows releases from 1.3.65 and onwards or MSI packages built from corresponding source releases. Not all releases support all the configuration options documented here.

Authoring a "Windows Installer" transform requires additional software for editing the MSI database tables and generating the transform from the modified MSI package. ORCA.EXE and MSITRAN.EXE which are included in the Windows Platform SDK ("Windows Installer" SDK) can be used for this purpose.

For reference, the schema for the MSI package is based on SCHEMA.MSI distributed with the Platform SDK.

For general information about "Windows Installer", refer to:

http://msdn.microsoft.com/library/en-us/msi/setup/windows_installer_start_page.asp

For general information about authoring MSI transforms, refer to:

http://msdn.microsoft.com/library/en-us/msi/setup/transforms.asp

The remainder of this document assumes some familiarity with authoring transforms. While the MSDN documentation for Windows Installer is a bit dense, the guide on MSI transforms found at the second link above is recommended reading. MSDN also includes a step-by-step example for creating a transform at:

http://msdn.microsoft.com/library/en-us/msi/setup/a_customization_transform_example.asp

7.1.2 Authoring a Transform

Transforms describe a set of modifications to be performed on an existing MSI for the purpose of customizing it. This is ordinarily done by making a copy of the MSI to be customized, modifying the copy and then using the old and the new MSI to generate a transform. For example:

1. copy openafs.msi openafs-modified.msi
2. (edit the openafs-modified.msi to include the necessary changes)
3. msitran -g openafs.msi openafs-modified.msi openafs-transform.mst
4. (generates openafs-transform.mst, which is the transform)

Transforms have an extension of .mst. 'msitran' is a tool distributed as part of the "Windows Installer" SDK (part of the Windows Platform SDK).

You can test a transform by:

1. copy openafs.msi openafs-test.msi
2. msitran -a openafs-transform.mst openafs-test.msi

and then checking the resulting openafs-test.msi to see if all changes you have made above to openafs-modified.msi is present in openafs-test.msi. 'msitran' will complain if some modification in the transform can not be successfully applied.

As mentioned above, you can use a tool like ORCA.EXE to edit the MSI databases directly when editing openafs-modified.msi. More details are given below.

7.2. Configuration Options

The logic necessary to implement many of the settings described in Appendix A are present in the MSI. Most of these can be controlled by setting the corresponding properties to the desired value. Some settings may require modifying existing registry entries (though not recommended) or adding new resources (like files or registry keys). Instructions for performing these tasks are below.

7.2.1 Configurable Properties

Most configurable properties correspond to registry keys or values. Due to the logic invoked based on the existence of these registry keys or values, they are only set if the associated property is defined to have a non null value. If the associated property is not defined in the MSI, the registry key or value will not be touched. By default, the MSI does not contain these properties and hence will not set the registry keys. You will need to add properties as needed to the MSI.

When one of the configurable properties is set, the installer will use the property value to set the corresponding setting in the HKEY_LOCAL_MACHINE registry hive. The HKEY_CURRENT_USER hive is not touched by the installer.

For each property, the associated registry setting is referenced by the same text used in Appendix A.

Strings are quoted using single quotes (e.g. 'a string'). An empty string is denoted as ''. Note that you can't author null values into the 'Property' table.

Numeric values should be authored as decimal strings.

7.2.1.1 Setting Properties

In order to set a property,

1. Open the MSI in ORCA.EXE
2. Select the 'Property' table from the list of tables on the left.
3. Find the property in the list of properties on the right, double click the value and type the new value.
4. If the property does not exist in the property list, right click the list and select 'Add Row', type the property name and the desired value.

7.2.1.2 OpenAFS for Windows Properties

(Service parameters):

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]
(Network provider): [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]
(OpenAFS Client): [HKLM\SOFTWARE\OpenAFS\Client]

7.2.1.2.1 Registry Properties

These properties are used to set the values of registry entries associated with OpenAFS for Windows.

AFSCACHEPATH Registry key : (Service Parameters) Registry value : CachePath [53] Valid values : string .
AFSCACHESIZE Registry key : (Service Parameters) Registry value : CacheSize [51] Valid values : numeric
AFSCELLNAME Registry key : (Service Parameters) Registry value : Cell [57] Valid values : string
FREELANCEMODE Registry key : (Service Parameters) Registry value : FreelanceClient [56] Valid values : '1' or '0'
HIDEDOTFILES Registry key : (Service Parameters) Registry value : HideDotFiles [57] Valid values : '1' or '0'
LOGONOPTIONS Registry key : (Network Provider) Registry value : LogonOptions [78] Valid values : '0', '1' or '3'

See Appendix Asection 2.1 (Domain Specific Configuration keys for Network Provider) for more details.
MOUNTROOT Registry key : (Service Parameters) Registry value : Mountroot [53] Valid values : string
NETBIOSNAME Registry key : (Service Parameters) Registry value : NetbiosName [54] Valid values : string (at most 15 characters)
NOFINDLANABYNAME Registry key : (Service Parameters) Registry value : NoFindLanaByName [61] Valid values : '1' or '0'
RXMAXMTU Registry key : (Service Parameters) Registry value : RxMaxMTU [58] Valid values : numeric
SECURITYLEVEL Registry key : (Service Parameters) Registry value : SecurityLevel [56] Valid values : '1' or '0'
SMBAUTHTYPE Registry key : (Service Parameters) Registry value : SMBAuthType [61] Valid values : '0','1' or '2'
STOREANSIFILENAMES Registry key : (OpenAFS Client) Registry value : StoreAnsiFilenames [69] Valid values : '0' or '1' This option is no longer supported as of 1.5.50 now that all file names are stored to AFS file servers using the UTF-8 encoding of Unicode.
USEDNS

Registry key : (Service Parameters)

Registry value : UseDNS [56]

Valid values : '1' or '0'

7.2.1.2.2 AFSCreds.exe Properties

These properties are combined to add a command line option to the shortcut that will be created in the Start:Programs:OpenAFS and Start:Programs:Startup folders (see CREDSSTARTUP). The method of specifying the option was chosen for easy integration with the Windows Installer user interface. Although other methods can be used to specify options to AFSCREDS.EXE, it is advised that they be avoided as transforms including such options may not apply to future releases of OpenAFS.

CREDSSTARTUP Valid values : '1' or '0' Controls whether AFSCreds.exe starts up automatically when the user logs on. When CREDSSTARTUP is '1' a shortcut is added to the 'Startup' folder in the 'Program menu' which starts AFSCREDS.EXE with the options that are determined by the other CREDS* properties.
CREDSAUTOINIT Valid values : '-a' or '' Enables automatic initialization.
CREDSIPCHDET Valid values : '-n' or '' Enables IP address change detection.
CREDSQUIET Valid values : '-q' or '' Enables quiet mode.
CREDSRENEWDRMAP Valid values : '-m' or '' Enables renewing drive map at startup.
CREDSSHOW Valid values : '-s' or '' Enables displaying the credential manager window when AFSCREDS starts up.

7.2.2 Existing Registry Entries

You can change existing registry values subject to the restrictions mentioned in the Windows Platform SDK. Pay special attention to component key paths and try to only change the 'Value' column in the 'Registry' table. If you want to add additional registry keys please refer to section 3 (Additional resources).

7.2.3 Replacing Configuration Files

The OpenAFS configuration files (CellServDB) can be replaced by your own configuration files. These files are contained in separate MSI components so that you can disable them individually.

The recommended method for replacing these files is to first disable the components containing the configuration files that you want to replace, and then add new components for the replacement files. This is outlined below (assuming you are using ORCA.EXE to author the transform).

Note that transforms are not a good way to add a new file as an embedded stream. The method outlined here places the file in the same directory as the MSI for deployment.

The walkthrough below is to add a custom 'CellServDB' file.

1. Disable the component that contains the configuration file that you want to replace.

   a. Locate and select the 'Component' table in the 'Tables' list.

   b. In the Component table, locate the component you need to change (Ctrl-F invokes the 'Find' dialog). The component names are listed below in section 7.2.3.1. For this example, the component name is 'elf_CellServDB'.

   c. Go to the 'Condition' column of the component.

   d. Enter a condition that evaluates to false. I.e. 'DONOTINSTALL'. (Note that an undefined property always evaluates to false).

      Note that you can also use this step to disable other configuration files without providing replacements.

2. Add a new component containing the new configuration file.

   a. Select the 'Component' table in the 'Tables' list.

   b. Select 'Tables'->'Add Row' (Ctrl-R).

   c. Enter the following :

Component	cmf_my_CellServDB
ComponentID	{7019836F-BB2C-4AF6-9463-0D6EC9035CF1}
Directory_	dirClient
Attributes	144
Condition	
KeyPath	fil_my_CellServDB

   Note that the ComponentId is an uppercase GUID. You can generate one using GUIDGEN.EXE or UUIDGEN.EXE, both of which are included in the Platform SDK.

   The Attributes value of 144 is a sum of msidbComponentAttributesPermanent (16) and msidbComponentAttributesNeverOverwrite (128). This ensures that local modifications are not overwritten or lost during an installation or uninstallation. These are the same settings used on the default configuration files.

   'fil_my_CellServDB' is a key into the 'File' table which we will fill later.

3. Add a new feature to hold the new component.

   a. Select the 'Feature' table.

   b. Add a new row (Ctrl-R or 'Tables'->'Add Row') with the following values:

Feature	fea_my_CellServDB
Feature_Parent	feaClient
Title	
Description	
Display	0
Level	30
Directory_	
Attributes	8

   It is important to create the new feature under the 'feaClient' feature, which will ensure that the configuration file will be installed when the client binaries are installed.

   Setting 'Display' to 0 will hide this feature from the feature selection dialog during an interactive installation. A value of 30 for 'Level' allows this feature to be installed by default (on a 'Typical' installation).

   The 'Attributes' value is msidbFeatureAttributesDisallowAdvertise (8), which is set on all features in the OpenAFS MSI. The OpenAFS MSI is not designed for an advertised installation.

4. Join the component and the feature.

   a. Select the 'FeatureComponents' table.

   b. Add a new row with the following values:

Feature	fea_my_CellServDB
Component	cmf_my_CellServDB

5. Add an entry to the 'File' table.

   a. Select the 'File' table.

   b. Add a new row with the following values:

File	fil_my_CellServDB
Component_	cmf_my_CellServDB
FileName	CellServDB
FileSize	(enter file size here)
Attributes	8192
Sequence	1000

   (leave other fields blank)

The 'Attributes' value is msidbFileAttributesNonCompressed (8192). This is because we will be placing this file in the same directory as the MSI instead of embedding the file in it. Transforms do not support updating compressed sources or adding new cabinet streams.

Finally, the 'Sequence' value of 1000 will be used later to distinguish the file as being in a separate source location than the other files in the MSI.

6. Set a media source for the file.

   a. Select the 'Media' table.

   b. Add a row with the following values :

DiskId	2
LastSequence	1000

(leave other fields blank)

The sequence number of 1000 designates this as the media source for the newly added file.

7.2.3.1 Components for Configuration Files

CellServDB: 'cpf_CellServDB' (ID {D5BA4C15-DBEC-4292-91FC-B54C30F24F2A})

7.2.4 Adding Domain Specific Registry Keys

Following is an example for adding domain specific registry keys.

Refer to Appendix A section 2.1 for more information.

Columns that are unspecified should be left empty.

We create a new feature and component to hold the new registry keys.

'Feature' table:
(new row) Feature : 'feaDomainKeys' Feature Parent : 'feaClient' Display : 0 Level : 30 Attributes : 10
'Component' table:
(new row) Component : 'rcm_DomainKeys' ComponentId : '{4E3FCBF4-8BE7-40B2-A108-C47CF743C627}' Directory : 'TARGETDIR' Attributes : 4 KeyPath : 'reg_domkey0'
'FeatureComponents' table:
(new row) Feature : 'feaDomainKeys' Component : 'rcm_DomainKeys'
'Registry' table:
(new row) Registry : 'reg_domkey0' Root : 2 Key : 'SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain' Component : 'rcm_DomainKeys'
(new row) Registry : 'reg_domkey1' Root : 2 Key : 'SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain' Name : '*' Component : 'rcm_DomainKeys'
(new row) Registry : 'reg_domkey2' Root : 2 Key : 'SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\ATHENA.MIT.EDU' Name : '*' Component : 'rcm_DomainKeys'

(new row) Registry : 'reg_domkey3' Root : 2 Key : 'SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain\ATHENA.MIT.EDU' Name : 'LogonOptions' Value : 1 Component : 'rcm_DomainKeys'
(new row) Registry : 'reg_domkey4' Root : 2 Key : SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST' Name : '*' Component : 'rcm_DomainKeys'
(new row) Registry : 'reg_domkey5' Root : 2 Key : 'SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST' Name : 'LogonOptions' Value : 0 Component : 'rcm_DomainKeys'
(new row) Registry : 'reg_domkey6' Root : 2 Key : 'SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST' Name : 'FailLoginsSilently' Value : 1 Component : 'rcm_DomainKeys'

The example adds domain specific keys for 'ATHENA.MIT.EDU' (enable integrated logon) and 'LOCALHOST' (disable integrated logon and fail logins silently).

7.2.5 Adding Site Specific Freelance Registry Keys

Following is an example for adding site specific Freelance registry keys to pre-populate the Mountpoints and Symlinks in the fake root.afs volume.

Columns that are unspecified should be left empty.

We create a new feature and component to hold the new registry keys.

'Feature' table:
(new row) Feature : 'feaFreelanceKeys' Feature Parent : 'feaClient' Display : 0 Level : 30 Attributes : 10
'Component' table:
(new row) Component : 'rcm_FreelanceKeys' ComponentId : '{4E3B3CBF4-9AE7-40C3-7B09-C48CF842C583}' Directory : 'TARGETDIR' Attributes : 4 KeyPath : 'reg_freekey0'
'FeatureComponents' table:
(new row) Feature : 'feaFreelanceKeys' Component : 'rcm_FreelanceKeys'
'Registry' table:
(new row) Registry : 'reg_freekey0' Root : 2 Key : 'SOFTWARE\OpenAFS\Client\Freelance' Component : 'rcm_FreelanceKeys'
(new row) Registry : 'reg_freekey1' Root : 2 Key : 'SOFTWARE\OpenAFS\Client\Freelance' Name : '0' Value : 'athena.mit.edu#athena.mit.edu:root.cell.' Component : 'rcm_FreelanceKeys'
(new row) Registry : 'reg_freekey2' Root : 2 Key : 'SOFTWARE\OpenAFS\Client\Freelance' Name : '1' Value : '.athena.mit.edu%athena.mit.edu:root.cell.' Component : 'rcm_FreelanceKeys'
(new row) Registry : 'reg_freekey3' Root : 2 Key : 'SOFTWARE\OpenAFS\Client\Freelance\Symlinks' Component : 'rcm_FreelanceKeys'
(new row) Registry : 'reg_freekey4' Root : 2 Key : 'SOFTWARE\OpenAFS\Client\Freelance\Symlinks' Name : '0' Value : 'athena:athena.mit.edu.' Component : 'rcm_FreelanceKeys'
(new row) Registry : 'reg_freekey5' Root : 2 Key : 'SOFTWARE\OpenAFS\Client\Freelance\Symlinks' Name : '1' Value : '.athena:.athena.mit.edu.' Component : 'rcm_FreelanceKeys'

The example adds a read-only mountpoint to the athena.mit.edu cell's root.afs volume as well as a read-write mountpoint. Aliases are also provided using symlinks.

7.3 Additional Resources

If you want to add registry keys or files you need to create new components and features for those. Refer to the Windows Platform SDK for details.

It is beyond the scope of this document to provide a comprehensive overview of how to add new resources through a transform. Please refer to the "Windows Installer" documentation for details. The relevant section is at :

http://msdn.microsoft.com/library/en-us/msi/setup/using_transforms_to_add_resources.asp

A sample walkthrough of adding a new configuration file is in section 2.3.

Add new features under the 'feaClient' or 'feaServer' as appropriate and set the 'Level' column for those features to equal the 'Level' for their parent features for consistency. Note that none of the features in the OpenAFS for Windows MSI package are designed to be installed to run from 'source' or 'advertised'. It is recommended that you set 'msidbFeatureAttributesFavorLocal' (0), 'msidbFeatureAttributesFollowParent' (2) and 'msidbFeatureAttributesDisallowAdvertise' (8) attributes for new features.

If you are creating new components, retain the same component GUID when creating new transforms against new releases of the OpenAFS MSI package.

After making the adjustments to the MSI database using ORCA.EXE you can generate a transform with MSITRAN.EXE as follows :

(Modified MSI package is 'openafs-en_US_new.msi' and the original MSI package is 'openafs-en_US.msi'. Generates transform 'openafs-transform.mst')

> msitran.exe -g openafs-en_US.msi openafs-en_US_new.msi openafs-transform.mst

See the Platform SDK documentation for information on command line options for MSITRAN.EXE.

7.4. Upgrades

The MSI package is designed to uninstall previous versions of OpenAFS for Windows during installation. Note that it doesn't directly upgrade an existing installation. This is intentional and ensures that development releases which do not have strictly increasing version numbers are properly upgraded.

Versions of OpenAFS that are upgraded by the MSI package are:

1. OpenAFS MSI package

   Upgrade code {6823EEDD-84FC-4204-ABB3-A80D25779833}

   Up to current release

2. MIT's Transarc AFS MSI package

   Upgrade code {5332B94F-DE38-4927-9EAB-51F4A64193A7}

   Up to version 3.6.2

3. OpenAFS NSIS package

   All versions

Note that versions of the OpenAFS NSIS package prior to 1.3.65 had a bug where it couldn't be uninstalled properly in unattended mode. Therefore the MSI package will not try to uninstall an OpenAFS NSIS package if running unattended. This means that group policy based deployments will fail on machines that have the OpenAFS NSIS package installed.

If you have used a different MSI package to install OpenAFS and wish to upgrade it you can author rows into the 'Upgrade' table as described in the Platform SDK.

When performing an upgrade with msiexec.exe execute the MSI with the repair options "vomus".

Chapter Appendix A. Registry Values

A.1. Service parameters

The service parameters primarily affect the behavior of the AFS client service (afsd_service.exe).

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Value: LanAdapter

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: -1

Variable: LANadapter

LAN adapter number to use. This is the lana number of the LAN adapter that the SMB server should bind to. If unspecified or set to -1, a LAN adapter with named 'AFS' or a loopback adapter will be selected. If neither are present, then all available adapters will be bound to. When binding to a non-loopback adapter, the NetBIOS name hostname%-AFS' will be used (where %hostname% is the NetBIOS name of the host truncated to 11 characters). Otherwise, the NetBIOS name will be 'AFS'.

[This parameter is ignored unless SMB mode is active.]

Value: CacheSize

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 98304 (CM_CONFIGDEFAULT_CACHESIZE)

Variable: cm_initParams.cacheSize

Size of the AFS data cache specified in 1k blocks.

Value: ChunkSize

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 20 (CM_CONFIGDEFAULT_CHUNKSIZE)

Variable: cm_logChunkSize (cm_chunkSize = 1 << cm_logChunkSize)

Maximum size of chunk for reading and writing. Actual chunk size is 2^cm_logChunkSize. The default chunk size is therefore 1 MB.

Value: BlockSize

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 4096 (CM_CONFIGDEFAULT_BLOCKSIZE)

Size of buffer allocation. Must be an even multiple of 1024 and (2^cm_logChuckSize mod BlockSize) must equal zero.

Value: Daemons

Value: Daemons

Type: DWORD

Default: 16 (CM_CONFIGDEFAULT_DAEMONS)

Variable: numBkgD

Number of background daemon threads used to fetch data from and store data to the file servers.

Value: ServerThreads

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 25 (CM_CONFIGDEFAULT_SVTHREADS)

Variable: numSvThreads

Number of SMB server or AFS Redirector worker threads used to process application file system requests..

Value: Stats

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 10000 (CM_CONFIGDEFAULT_STATS)

Variable: cm_initParams.nStatCaches

The number of file system status objects. One status object is required for each directory entry (file, directory, mount point, symlink).

Value: Volumes

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 3333 (CM_CONFIGDEFAULT_STATS/3)

Variable: cm_initParams.nVolumes

The number of volume group objects. One volume group object is required for each set of volume, volume.readonly and volume.backup.

Value: Cells

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 1024 (CM_CONFIGDEFAULT_CELLS)

Variable: cm_initParams.nCells

The number of cell objects. One cell object is required for each cell accessed by the AFS cache manager.

Value: LogoffPreserveTokens

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default : 0

If enabled (set to 1), the Logoff Event handler will not attempt to delete the user's tokens if the user's profile is stored outside of AFS. This option only applies when the AFS SMB gateway interface is in use.

Value: RootVolume

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_SZ

Default: "root.afs"

Variable: cm_rootVolumeName

Root volume name.

Value: MountRoot

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_SZ

Default: "/afs"

Variable: cm_mountRoot

Name of root mount point. In symlinks, if a path starts with cm_mountRoot, it is assumed that the path is absolute (as opposed to relative) and is adjusted accordingly. Eg: if a path is specified as /afs/athena.mit.edu/foo/bar/baz and cm_mountRoot is "/afs", then the path is interpreted as \\afs\all \athena.mit.edu\foo\bar\baz. If a path does not start with with cm_mountRoot, the path is assumed to be relative and suffixed to the reference directory (i.e. directory where the symlink exists)

Value: CachePath

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: Use REG_SZ if the path contains no expansion variables or REG_EXPAND_SZ if it does.

Default: "%TEMP%\AFSCache" (REG_EXPAND_SZ)

Variable: cm_CachePath

Location of on-disk cache file. The default is the SYSTEM account's TEMP directory. The attributes assigned to the file are HIDDEN and SYSTEM.

Value: NonPersistentCaching

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD [0..1]

Default: 0

Variable: buf_CacheType

When this registry value is set to a non-zero value, the CachePath value is ignored and the cache data is stored in the windows paging file. This disables the use of persistent caching and the ability to maintain a single UUID for the AFS client service across restarts.

Value: ValidateCache

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD [0..2]

Default: 1

Variable: buf_CacheType

This value determines if and when persistent cache validation is performed.

0 - Validation is disabled
1 - Validation is performed at startup
2 - Validation is performed at shutdown

Value: TrapOnPanic

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 0

Variable: traceOnPanic

Issues a breakpoint in the event of a panic. (breakpoint: _asm int 3).

Value: NetbiosName

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_EXPAND_SZ

Default: "AFS"

Variable: cm_NetbiosName

Specifies the NetBIOS name (or SMB Server Name) to be used when binding to a Loopback adapter. To provide the old behavior specify a value of "%COMPUTERNAME%-AFS". When the AFS Redirector interface is in use, this value specifies the UNC server name registered with the Windows Multiple UNC Provider.

Value: IsGateway

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 0

Variable: isGateway

Select whether or not this AFS client should act as a gateway. If set and the NetBIOS name hostname-AFS is bound to a physical NIC, other machines in the subnet can access AFS via SMB connections to hostname-AFS.

When IsGateway is non-zero, the LAN adapter detection code will avoid binding to a loopback adapter. This will ensure that the NetBIOS name will be of the form hostname-AFS instead of the value set by the "NetbiosName" registry value.

This setting only applies when the AFS SMB interface is in use.

Value: ReportSessionStartups

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 0

Variable: reportSessionStartups

If enabled, all SMB sessions created are recorded in the Application event log. This also enables other events such as drive mappings or various error types to be logged.

Value: TraceBufferSize

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 10000 (CM_CONFIGDEFAULT_TRACEBUFSIZE)

Variable: traceBufSize

Number of entries to store in trace log.

Value: SysName

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_SZ

Default: "x86_win32 i386_w2k i386_nt40" (X86)

"amd64_win64 x86_win32 i386_w2k" (AMD64)

Variable: cm_sysName

Provides an initial value for "fs sysname". The string can contain one or more replacement values for @sys in order of preference separated by whitespace.

Value: SecurityLevel

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 0

Variable: cryptall

Enables encryption on RX calls.

Value: UseDNS

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 1

Variable: cm_dnsEnabled

Enables resolving volservers using AFSDB DNS and SRV DNS queries.

Value: FreelanceClient

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 0

Variable: cm_freelanceEnabled

Enables freelance client.

Value: FreelanceImportCellServDB

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 0

Variable: cm_freelanceImportCellServDB

Creates Freelance Mount Points for each cell listed in the CellServDB during startup.

Value: FreelanceDiscovery

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 1

Variable: cm_freelanceDiscovery

Enables automatic discovery of cell mount points within the Freelance root.

Value: HideDotFiles

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1,0}

Default: 1

Variable: smb_hideDotFiles

Enables marking dotfiles with the hidden attribute. Dot files are files whose name starts with a period (excluding "." and "..").

Value: MaxMpxRequests

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 50

Variable: smb_maxMpxRequests

Maximum number of multiplexed SMB requests that can be made.

Value: MaxVCPerServer

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 100

Variable: smb_maxVCPerServer

Maximum number of SMB virtual circuits.

Value: Cell

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_SZ

Default: <none>

Variable: rootCellName

Name of root cell (the cell from which the RootVolume, root.afs, should be mounted in \\afs\all).

Value: RxEnablePeerStats

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

Variable: rx_enable_peer_stats

When set to 1, the Rx library collects peer statistics.

Value: RxEnableProcessStats

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

Variable: rx_extra_process_stats

When set to 1, the Rx library collects process statistics.

Value: RxExtraPackets

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 120

Variable: rx_extraPackets

When set, this number of extra Rx packets are allocated at startup.

Value: RxMaxMTU

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 0

Variable: rx_mtu

If set to anything other than 0, that value is used as the maximum send and receive MTU supported by the RX interface.

In order to enable OpenAFS to operate across releases of the Cisco IPSec VPN client prior than 5.0, this value must be set to 1264 or smaller.

Value: RxNoJumbo

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0,1}

Default: 0

Variable: rx_nojumbo

If enabled, does not send or indicate that we are able to send or receive RX jumbograms.

Value: ConnDeadTimeout

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 0 (seconds)

Variable: ConnDeadtimeout

When the value is 0, this setting is configured to be one-half the HardDeadTimeout value.

Value: HardDeadTimeout

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 0 (seconds)

Variable: HardDeadtimeout

The Hard Dead Time is enforced to be at least double the ConnDeadTimeout. The provides an opportunity for at least one retry.

The value 0 seconds means that the real timeout should be set to be equal to the minimum SMB timeout which can be configured in the registry at:

[HKLM\SYSTEM\CurrentControlSet\Services\lanmanworkstation\parameters]

SessTimeout

If the minimum SMB timeout is not specified the value is 45 seconds. See http://support.microsoft.com:80/support/kb/articles/Q102/0/67.asp [http://support.microsoft.com/support/kb/articles/Q102/0/67.asp]

Value: IdleDeadTimeout

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 1200 (seconds)

Variable: IdleDeadtimeout

The Idle Dead Time determines how long the cache manager will wait for an RPC on a non-replicated volume to complete when the service is responding only with keep alive messages. When there is no replica available there is no other file server to try. An idle dead timeout in this case is fatal. This option is intended to protect a client against a file server that never responds. This value must be larger that the file server hard dead timeout of 120 seconds.

Value: ReplicaIdleDeadTimeout

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 180 (seconds)

Variable: ReplicaIdleDeadtimeout

The Replica Idle Dead Time determines how long the cache manager will wait for an RPC on a replicated volume to complete when the service is responding only with keep alive messages. When a volume is replicated the cache manager can choose to retry the request against a file server hosting one of the replicas. This value must be larger that the file server hard dead timeout of 120 seconds.

Value: NATPingInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 0 (seconds)

Variable: NatPingInterval

The NAT Ping Interval determines how frequently a low-level Rx ping packet is sent to every file server using an anonymous connection. The value 0 seconds disables the functionality. A non-zero value activates the NAT ping behavior. This option can be enabled on clients that access AFS file servers from behind a NAT or port mapping firewall that maintains a short timeout on UDP port mappings. In this case the AFS cache manager may not receive AFS callbacks from the file server. This registry value should be set to a number of seconds shorter than the port mapping timeout period. When there is no other information to go on, a value of 20 seconds can be used.

Value: TraceOption

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0-15}

Default: 0

Enables logging of debug output to the Windows Event Log.

Bit 0 enables logging of "Logon Events" processed by the Network Provider and Winlogon Event Notification Handler.

Bit 1 enables logging of events captured by the AFS Client Service.

Bit 2 enables real-time viewing of "fs trace" logging with DbgView or similar tools.

Bit 3 enables "fs trace" logging on startup.

Value: AllSubmount

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

Variable: allSubmount (smb.c)

By setting this value to 0, the "\\NetbiosName\all" mount point will not be created. This allows the read-write versions of root.afs to be hidden.

Value: NoFindLanaByName

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

Disables the attempt to identity the network adapter to use by looking for an adapter with a display name of "AFS".

Value: MaxCPUs

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {1..32} or {1..64} depending on the architecture

Default: 2

If this value is specified, afsd_service.exe will restrict itself to executing on the specified number of CPUs if there are a greater number installed in the machine. Performance profiling shows that overall system performance degrades when the afsd_service.exe is permitted to execute on more than two cores.

Value: SmbAuthType

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0..2}

Default: 2

If this value is specified, it defines the type of SMB authentication which must be present in order for the Windows SMB client to connect to the AFS Client Service's SMB server. The values are:

0 = No authentication required

1 = NTLM authentication required
2 = Extended (GSS SPNEGO) authentication required

The default is Extended authentication This value only applies when the SMB server interface is in use.

Value: MaxLogSize

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0 .. MAXDWORD}

Default: 100K

This entry determines the maximum size of the %WINDIR%\TEMP\afsd_init.log file. If the file is larger than this value when afsd_service.exe starts the file will be reset to 0 bytes. If this value is 0, it means the file should be allowed to grow indefinitely.

Value: FlushOnHibernate

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0,1}

Default: 1

If set, flushes all volumes before the machine goes on hibernate or stand-by.

Value: DaemonCheckDownInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 180

This value controls how frequently the AFS cache manager probes servers that are marked as "down".

Value: DaemonCheckUpInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 600

This value controls how frequently the AFS cache manager probes servers that are marked as "up".

Value: DaemonCheckVolInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 3600

This value controls how frequently the AFS cache manager forces a reset on the existing volume database information.

Value: DaemonCheckCBInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 60

This value controls how frequently the AFS cache manager checks for callback invalidation.

Value: DaemonCheckLockInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 60

This value controls how frequently the AFS cache manager checks for invalid file locks.

Value: DaemonCheckTokenInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 180

This value controls how frequently the AFS cache manager checks for expired tokens.

Value: DaemonCheckOfflineVolInterval

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD (seconds)

Default: 600

This value controls how frequently the AFS cache manager checks offline volumes to see if they have come back online. At the same time volumes which were determined to be busy have their state reset to online.

Value: CallBackPort

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 7001

This value specifies which port number should be used for receiving callbacks from the file server. The standard AFS Callback port is 7001. Alternative values can be useful if the client is behind a NAT and a permanent port mapping for the client is being configured.

Value: EnableServerLocks

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1, 2}

Default: 1

Determines whether or not the AFS file server is contacted for

0: never obtain server locks

1: obtain server locks unless the file server says not to

2: always obtain server locks

Value: DeleteReadOnly

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

Determines whether or not the AFS Cache Manager will permit files marked with the "Read Only" DOS attribute to be deleted or not. For compatibility with Explorer, the default is ‘no'.

0: do not permit "Read Only" files to be deleted.

1: delete files that have the "Read Only" attribute set without complaint.

Value: BPlusTrees

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

Determines whether or not the AFS Cache Manager uses locally constructed B+ Trees to speed up the performance of directory searches.

0: do not use B+ Trees for directory lookups

1: use B+ Trees for directory lookups

Value: PrefetchExecutableExtensions

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: MULTI_SZ

Default: none specified

The AFS Cache Manager will pre-fetch the entire contents of any file whose name matches ends with one of the specified extensions. This option is intended for use primarily with executables and dynamic link libraries that should be fully cached prior to a machine losing its connection with the file server.

Value: OfflineReadOnlyIsValid

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

Determines whether or not cached data from .readonly volumes is considered valid even if a callback cannot be registered with a file server. This option is meant to be used by organizations for whom .readonly volume content very rarely changes (if ever.)

0: do not treat offline .readonly content as valid

1: treat offline .readonly content as valid

Value: GiveUpAllCallBacks

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

Determines whether or not the AFS Cache Manager will give up all callbacks prior to the service being suspended or shutdown. Doing so will have significant performance benefits for the file servers. However, file servers older than 1.4.6 can become unstable if the GiveUpAllCallBacks RPC is executed.

0: do not perform GiveUpAllCallBacks RPCs

1: perform GiveUpAllCallBacks RPCs

Value: ReadOnlyVolumeVersioning

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

Determines whether or not the AFS Cache Manager will will make use of the volume version information reported by the file server as part of the AFSVolSync data structure. Use of volume version information can significantly reduce the number of FetchStatus RPCs issued on objects stored in read-only volumes. This functionality is disabled by default because all OpenAFS file servers older than OpenAFS 1.4.10 failed to include valid volume version information as part of the BulkStatus and InlineBulkStatus RPCs.

0: do not make use of volume version information

1: make use of volume version information

Value: FollowBackupPath

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

Determines whether or not the AFS Cache Manager will give preference to .backup volumes when following mount points that originate in a .backup volume.

0: do not prefer .backup volumes when the mount point originates in a .backup volume.

1: prefer .backup volumes when the mount point originates in a .backup volume.

Value: RxUdpBufSize

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {bytes}

Default: 262144

Specifies the UDP socket receive and send buffer sizes..

Value: VerifyData

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

1: after every RXAFS_StoreData RPC immediately perform an RXAFS_FetchData RPC and verify that the data was correctly stored on the file server. If the data does not match, retry the store operation until it does.

The "fs getverify" and "fs setverify {on, off}" commands can be used to query and set this value at runtime.

Value: ShortNames

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0 on Windows 7, Windows 8, Server 2008 R2 and Server 2012, 1 otherwise

Determines whether or not the AFS Cache Manager will generate 8.3 compatible shortnames for all objects stored in AFS. Short names are disabled by default on Windows 8 and Server 2012. All prior operating systems enable short names by default.

0: do not generate 8.3 compatible short names.

1: generate 8.3 compatible short names.

Value: DirectIO

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

The AFS redirector as of 1.7.22 supports two I/O processing mechanisms. The new implementation is referred to as Direct I/O. Direct I/O provides two benefits over the prior implementation. First, it is faster. Second, it provides support for CreateFile(FILE_FLAG_NO_BUFFERING). When a file is opened with the FILE_FLAG_NO_BUFFERING flag set, the AFSCache is bypassed and all I/O operations on the file are performed directly to and from the file server.

0: use the older I/O processing mechanism.

1: use the new Direct I/O processing mechanism.

Value: VolumeInfoReadOnlyFlag

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 1

The Win32 GetVolumeInformation and GetVolumeInformationByHandle APIs permit applications to query volume attributes such as Case Preserving, Case Insensitive lookups, support for hard links, support for reparse points, support for Unicode and whether or not the volume is read only. The FILE_READ_ONLY_VOLUME flag when set permits applications such as the Explorer Shell to disable the "Delete" and "Rename" options and prevent copying files into the volume without issuing the request to the file system. Unfortunately, the Windows 7 explorer shell has a bug when a drive letter is mapped to a UNC path. If the mapped path refers to a read only volume, then all volumes accessible via the drive letter are also treated as read only. This bug is fixed in Windows 8 and Server 2012. To improve application compatibility the setting of the FILE_READ_ONLY_VOLUME flag is disabled by default on Windows 7 and below and enabled on Windows 8 and above.

0: prevent setting the FILE_READ_ONLY_VOLUME flag. (default on Win7 and below).

1: permit setting the FILE_READ_ONLY_VOLUME flag. (default on Win8 and above)

Value: ReparsePointPolicy

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD {0, 1}

Default: 0

Windows file systems use reparse points [http://msdn.microsoft.com/en-us/library/windows/desktop/aa365503%28v=vs.85%29.aspx] to represent special file system objects such as NTFS Junctions, Symbolic Links and AFS Mount Points. Windows applications must be designed to work with Symbolic Links [http://msdn.microsoft.com/en-us/library/windows/desktop/aa365680%28v=vs.85%29.aspx] because several standard file system functions [http://msdn.microsoft.com/en-us/library/windows/desktop/aa365682%28v=vs.85%29.aspx] behave differently when the provided path specifies a reparse point. Although there is not a significant impact for Symbolic Links to Directories and Mount Points to volume root directories, Symbolic Links to Files can result in applications misinterpreting the file size and attributes.

The ReparsePointPolicy value permits alternate behaviors for reparse point objects on a global basis. In this version, there is only one policy option which permits Symbolic Links to Files to be represented with the target file's size and attributes in the output of FindFirstFile, GetFileAttributes, and GetFileAttributesEx operations.

0: All Reparse Points are treated as reparse points.

1: Reparse Points to Files treated as the target File.

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters\GlobalAutoMapper]

Value: <DriveLetter>

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters\GlobalAutoMapper]

Type: REG_SZ

Specifies the submount name to be mapped by afsd_service.exe at startup to the provided drive letter.

This option is deprecated.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Value: CellServDBDir

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_SZ

Default: <not defined>

Specifies the directory containing the CellServDB file. When this value is not specified, the ProgramData directory is searched and if the CellServDB file is not found, the AFS Client install directory is used.

Value: VerifyServiceSignature

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_DWORD

Default: 0x1

This value can be used to disable the runtime verification of the digital signatures applied to afsd_service.exe and the OpenAFS DLLs it loads. This test is performed to verify that the DLLs which are loaded by afsd_service.exe are from the same distribution as afsd_service.exe. This is to prevent random errors caused when DLLs from one distribution of AFS are loaded by another one. This is not a security test. The reason for disabling this test is to free up additional memory which can be used for a large cache size.

Value: IoctlDebug

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_DWORD

Default: 0x0

This value can be used to debug the cause of pioctl() failures. Set a non-zero value and the pioctl() library will output status information to stdout. Executing command line tools such as tokens.exe, fs.exe, etc can then be used to determine why the pioctl() call is failing.

Value: MiniDumpType

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_DWORD

Default: 0x0 (MiniDumpNormal)

This value is used to specify the type of minidump generated by afsd_service.exe either when the process crashes or when a user initiated is dump file is generated with the "fs.exe minidump" command.

Valid values are dependent on the version of DbgHelp.dll installed on the machine. The best version to use is not the version that comes with the operating system but the version that is included in the most recent release of " Microsoft Debugging Tools for Windows [http://www.microsoft.com/whdc/devtools/debugging/default.mspx]". See the Microsoft Developer Library for further information.

MiniDumpNormal = 0x00000000
MiniDumpWithDataSegs = 0x00000001
MiniDumpWithFullMemory = 0x00000002
MiniDumpWithHandleData = 0x00000004
MiniDumpFilterMemory = 0x00000008
MiniDumpScanMemory = 0x00000010
MiniDumpWithUnloadedModules = 0x00000020
MiniDumpWithIndirectlyReferencedMemory = 0x00000040
MiniDumpFilterModulePaths = 0x00000080
MiniDumpWithProcessThreadData = 0x00000100
MiniDumpWithPrivateReadWriteMemory = 0x00000200
MiniDumpWithoutOptionalData = 0x00000400
MiniDumpWithFullMemoryInfo = 0x00000800
MiniDumpWithThreadInfo = 0x00001000
MiniDumpWithCodeSegs = 0x00002000

Value: EnableSMBAsyncStore

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_DWORD

Default: 0x1

This value can be used to disable the use of SMB Asynchronous Store operations.

Value: SMBAsyncStoreSize

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_DWORD

Default: 32

This value determines the size of SMB Asynchronous Store operations.This value can be used to increase the write performance on higher speed networks by increasing the value. The value must be a multiple of the cache buffer block size and cannot be larger than the cache manager chunk size. The specified value will be adjusted to enforce its compliance with these restrictions.

Value: StoreAnsiFilenames

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Type: REG_DWORD

Default: 0x0

This value can be used to force the AFS Client Service to store filenames using the Windows system's ANSI character set instead of the OEM Code Page character set which has traditionally been used by SMB file systems.

Note: The use of ANSI characters will render access to files with 8-bit OEM file names inaccessible from Windows. This option is of use primarily when you wish to allow file names produced on Windows to be accessible from Latin-1 UNIX systems and vice versa.

This value is ignored now that all file names are processed as Unicode and stored on the file server as UTF-8.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CSCPolicy]

Value: <smb share name>

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CSCPolicy]

Type: REG_SZ

Default: <none>

This key is used to map SMB/CIFS shares to Client Side Caching (off-line access) policies. For each share one of the following policies may be used: "manual", "programs", "documents", "disable".

These values used to be stored in afsdsbmt.ini

Regkey: [HKLM\SOFTWARE\OpenAFS\Client \CellServDB]

The *CellServDB* key is an alternative to the *CellServDB* file that can be used either to supplement or override its contents. This registry entry is meant to provide organizations that centrally manage their client configurations using *Active Directory Group Policy* a means of updating records for individual cells or servers without pushing out a new file.

At the present time the *CellServDB* key contains no values; only subkeys. Each subkey is the name of a *Cell*. For example, *grand.central.org*.

Support for registry *CellServDB* configuration was added in 1.5.60.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB \<cellname>]

The actual name of the <*cellname*> key is the full name of the cell whose configuration is being specified. The <*cellname*> key contains both values and subkeys. Each subkey represents a single host name or IP address. When a host is to be known by more than one name or IP address, a separate subkey should be created for each. Unlike the <cellname> key name, the <server> key names do not have to be actual host names.

Value: Description

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>]

Type: REG_SZ

Default: <none>

This value is used to store a description of the Cell appropriate for display in end user facing tools.

Value: ForceDNS

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>]

Type: REG_DWORD

Range: 0 or 1

Default: 0 if <server> subkeys exist; 1 otherwise

When set to 1 all server configuration provided in the registry or the *CellServDB file* is ignored and DNS AFSDB lookups are used instead.

Value: LinkedCell

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>]

Type: REG_SZ

Default: <none>

This value names an alternative cell to which this cell should be linked. When two cells are linked by the OpenAFS client, volume lookups that fail in the specified cell will be searched for in the linked cell and when tokens are requested for one of the cells they will be obtained for both. This functionality can be used for example to develop a test cell that is equivalent to a production cell with the exception that it substitutes test versions of volumes for the production versions. Another use is to assist in the transition from one cell name to another.

See also: Linked Cells.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<cellname>\<server>]

The actual name of the *<server>* key may be a fully qualified domain name of the server whose configuration is being specified. If a domain name is specified as the key name, all values become optional.

Value: HostName

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>\<Server>]

Type: REG_SZ

Default: <none>

This value is used to specify a fully qualified domain name appropriate that matches either a DNS A or DNS CNAME record. If provided, this value supercedes the name of the <server> key. It is recommended that the value of this field be terminated with a period in order to avoid the use of domain substitution during the gethostbyname() evaluation.

Value: IPv4Address

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>\<Server>]

Type: REG_SZ

Default: <none>

If the DNS host name lookup fails this value will be used as the IPv4 address for the server.

Value: Rank

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>\<Server>]

Type: REG_DWORD

Range: 0..65535

Default: 0

This value specifies the default server preference. A value of 0 indicates that no preference has been specified. When non-zero values are specified lower values indicate a stronger preference than higher values.

Value: Comment

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\CellServDB\<Cell Name>\<Server>]

Type: REG_SZ

Default: <none>

A text string that can be displayed to end users to describe the server.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Freelance]

Value: <numeric value>

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Freelance]

Type: REG_SZ

Default: <none>

This key is used to store dot terminated mount point strings for use in constructing the fake root.afs volume when Freelance (dynamic roots) mode is activated.

"athena.mit.edu#athena.mit.edu:root.cell."

".athena.mit.edu%athena.mit.edu:root.cell."

These values used to be stored in afs_freelance.ini

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Freelance \Symlinks]

Value: <numeric value>

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Freelance\Symlinks]

Type: REG_SZ

Default: <none>

This key is used to store a dot terminated symlink strings for use in constructing the fake root.afs volume when Freelance (dynamic roots) mode is activated.

"linkname:destination-path."

"athena:athena.mit.edu."

"home:athena.mit.edu\user\j\a\jaltman."

"filename:path\file."

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Realms]

The Realms key is used to provide initialization data to be used when new identities are added to the Network Identity Manager. The AFS Provider will search for a subkey that matches the realm of the identity. If such a key exists, its values will be used to populate the AFS configuration for the identity.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Realms \<Realm Name>]

In addition to the optional values, this key contains one subkey for each cell that is to be added to the AFS Provider configuration.

Value: AFSEnabled

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Realms\<Realm Name>]

Type: REG_DWORD

Default: 0x01

This key is used to specify whether the new identity should be configured to obtain AFS credentials. In general, it is only specified when disabling the acquisition of AFS credentials is desired. The default is to obtain AFS credentials.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Realms \<Realm Name>\<Cell Name>]

Value: MethodName

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Realms\<Realm Name>\<Cell Name>]

Type: REG_SZ

Default: <none>

This key is used to specify the token acquisition method to be used. When unspecified, the AFS provider will automatically try Kerberos v5 and then Kerberos v5 (if available). As of this writing valid method names include "Auto", "Kerberos5", "Kerberos524", "Kerberos4".

Note: Kerberos524 and Kerberos4 cannot be used with 64-bit Kerberos for Windows.

Value: Realm

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Realms\<Realm Name>\<Cell Name>]

Type: REG_SZ

Default: <none>

This key is used to specify the realm to be used when acquiring AFS tokens. If not specified, the realm will be determined by performing a domain to realm mapping on the domain of a random volume location database server for the cell.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client \Submounts]

Value: <Submount Name>

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Submounts]

Type: REG_EXPAND_SZ

Default: <none>

This key is used to store mappings of UNIX style AFS paths to submount names which can be referenced as UNC paths. For example the submount string "/athena.mit.edu/user/j/a/jaltman" can be associated with the submount name "jaltman.home". This can then be referenced as the UNC path \\AFS\jaltman.home.

These values used to be stored in afsdsbmt.ini

NOTE: Submounts should no longer be used with OpenAFS. Use the Windows Explorer to create drive mappings to AFS UNC paths instead of using the AFS Submount mechanism.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Server Preferences\VLDB]

Value: <hostname or ip address>

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Server Preferences\VLDB]

Type: REG_DWORD

Default: <none>

This key is used to specify a default set of VLDB server preferences. For each entry the value name will be either the IP address of a server or a fully qualified domain name. The value will be the ranking. The ranking will be adjusted by a random value between 0 and 15 prior to the preference being set.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Server Preferences\File]

Value: <hostname or ip address>

Regkey: [HKLM\SOFTWARE\OpenAFS\Client\Server Preferences\File]

Type: REG_DWORD

Default: <none>

This key is used to specify a default set of File server preferences. For each entry the value name will be either the IP address of a server or a fully qualified domain name. The value will be the ranking. The ranking will be adjusted by a random value between 0 and 15 prior to the preference being set.

A.2. Integrated Logon Network Provider Parameters

Affects the network provider (afslogon.dll).

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Value: FailLoginsSilently

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: DWORD

Default: 0

Do not display message boxes if the login fails.

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Value: NoWarnings

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: DWORD

Default: 0

Disables visible warnings during logon.

Value: Debug

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: DWORD

Default: 0

Set to 1 to turn on "AFS Logon" event logging to the Windows Event Log.

Value: AuthentProviderPath

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: REG_SZ

NSIS: %WINDIR%\SYSTEM32\afslogon.dll

Specifies the install location of the authentication provider dll.

Value: Class

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: DWORD

NSIS: 0x02

Specifies the class of network provider

Value: DependOnGroup

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: REG_MULTI_SZ

NSIS: PNP_TDI

Specifies the service groups upon which the AFS Client Service depends. Windows should not attempt to start the AFS Client Service until all of the services within these groups have successfully started.

Value: DependOnService

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: REG_MULTI_SZ

NSIS: Tcpip NETBIOS RpcSs

Specifies a list of services upon which the AFS Client Service depends. Windows should not attempt to start the AFS Client Service until all of the specified services have successfully started.

Value: Name

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: REG_SZ

NSIS: "OpenAFSDaemon"

Specifies the display name of the AFS Client Service

Value: ProviderPath

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider]

Type: REG_SZ

NSIS: %WINDIR%\SYSTEM32\afslogon.dll

Specifies the DLL to use for the network provider

A.2.1 Domain specific configuration keys for the Network Provider

The network provider can be configured to have different behavior depending on the domain that the user logs into. These settings are only relevant when using integrated login. A domain refers to an Active Directory (AD) domain, a trusted Kerberos (non-AD) realm or the local machine (i.e. local account logins). The domain name that is used for selecting the domain would be the domain that is passed into the NPLogonNotify function of the network provider.

Domain specific registry keys are:

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider]

(NP key)

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain]

(Domains key)

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain\<Domain Name>]

(Specific domain key. One per domain.)

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST]

(Localhost key)

Domain Specific Example:

HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider

|

+- Domain

+-AD1.EXAMPLE.COM

+-AD1

+-AD2.EXAMPLE.NET

+-AD2

+-LOCALHOST

+-Administrator

+-Other User

Each of the domain specific keys can have the set of values described in 2.1.1. The effective values are chosen as described in 2.1.2.

A.2.1.1 Domain Specific Configuration Values

Regkeys: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider] [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain] [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\"domain name"] [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\"domain name"]["user name"] [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST] [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST\"user name"]

Value: LogonOptions

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain\LOCALHOST\<user name>]

Type: DWORD

Default: 0x01

NSIS/WiX: depends on user configuration

0x00 - Integrated Logon is not used
0x01 - Integrated Logon is used

Value: FailLoginsSilently

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: DWORD (1|0)

Default: 0

NSIS/WiX: (not set)

If true, does not display any visible warnings in the event of an error during the integrated login process.

Value: LogonScript

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: REG_SZ or REG_EXPAND_SZ

Default: (null)

NSIS/WiX: (only value under NP key) <install path>\afscreds.exe -:%s -x -a -m -n -q

A logon script that will be scheduled to be run after the profile load is complete. If using the REG_EXPAND_SZ type, you can use any system environment variable as "%varname%" which would be expanded at the time the network provider is run. Optionally using a "%s" in the value would result in it being expanded into the AFS SMB username for the session.

Value: LoginRetryInterval

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: DWORD

Default: 30

NSIS/WiX: (not set)

If the OpenAFS client service has not started yet, the network provider will wait for a maximum of "LoginRetryInterval" seconds while retrying every "LoginSleepInterval" seconds to check if the service is up.

Value: LoginSleepInterval

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: DWORD

Default: 5

NSIS/WiX: (not set)

See description of LoginRetryInterval.

Value: Realm

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: REG_SZ

NSIS: <not set>

When Kerberos v5 is being used, Realm specifies the Kerberos v5 realm that should be appended to the first component of the Domain logon username to construct the Kerberos v5 principal for which AFS tokens should be obtained.

Value: TheseCells

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: REG_MULTI_SZ

NSIS: <not set>

When Kerberos v5 is being used, TheseCells provides a list of additional cells for which tokens should be obtained with the default Kerberos v5 principal.

Value: Username

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \<domain name>\<user name>]

[HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider\Domain \LOCALHOST\<user name>]

Type: REG_SZ

NSIS: <not set>

Username specifies an alternate username to be combined with the Realm when constructing the Kerberos v5 principal for which AFS tokens should be obtained.

A.2.1.2 Selection of effective values for domain specific configuration

During login to domain X, where X is the domain passed into NPLogonNotify as lpAuthentInfo->LogonDomainName or the string 'LOCALHOST' if lpAuthentInfo->LogonDomainName equals the name of the computer, the following keys will be looked up.

1. NP key. ("HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\NetworkProvider")

2. Domains key. (NP key\"Domain")

3. Specific domain key. (Domains key\X)

If the specific domain key does not exist, then the domains key will be ignored. All the configuration information in this case will come from the NP key.

If the specific domain key exists, then for each of the values metioned in (2), they will be looked up in the specific domain key, domains key and the NP key successively until the value is found. The first instance of the value found this way will be the effective for the login session. If no such instance can be found, the default will be used. To re-iterate, a value in a more specific key supercedes a value in a less specific key. The exceptions to this rule are stated below.

A.2.1.3 Exceptions to A.2.1.2

To retain backwards compatibility, the following exceptions are made to A.2.1.2.

2.1.3.1 'FailLoginsSilently'

Historically, the 'FailLoginsSilently' value was in HKLM\SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\Parameters key and not in the NP key. Therefore, for backwards compatibility, the value in the Parameters key will supercede all instances of this value in other keys. In the absence of this value in the Parameters key, normal scope rules apply.

2.1.3.2 'LogonScript'

If a 'LogonScript' is not specified in the specific domain key nor in the domains key, the value in the NP key will only be checked if the effective 'LogonOptions' specify a high security integrated login. If a logon script is specified in the specific domain key or the domains key, it will be used regardless of the high security setting. Please be aware of this when setting this value.

A.3. AFS Credentials System Tray Tool parameters

Affects the behavior of afscreds.exe

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services \TransarcAFSDaemon\Parameters]

Value: Gateway

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_SZ

Default: ""

Function: GetGatewayName()

If the AFS client is utilizing a gateway to obtain AFS access, the name of the gateway is specified by this value.

Value: Cell

Regkey: [HKLM\SYSTEM\CurrentControlSet\Services\TransarcAFSDaemon\Parameters]

Type: REG_SZ

Default: <none>

Variable: IsServiceConfigured()

The value Cell is used to determine if the AFS Client Service has been properly configured or not.

Regkey: [HKLM\SOFTWARE\OpenAFS\Client] [HKCU \SOFTWARE\OpenAFS\Client]

Value: ShowTrayIcon

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Type: DWORD {0, 1}

Default: 1

Function: InitApp(), Main_OnCheckTerminate()

This value is used to determine whether or not a shortcut should be maintained in the user's Start Menu->Programs->Startup folder.

This value used to be stored at [HKLM\Software\TransarcCorporation\AFS Client\AfsCreds].

The current user value is checked first; if it does not exist the local machine value is checked.

Value: EnableKFW

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Type: DWORD {0, 1}

Default: 1

Function: KFW_is_available()

When MIT Kerberos for Windows can be loaded, Kerberos v5 will be used to obtain AFS credentials. By setting this value to 0, the internal Kerberos v4 implementation will be used instead. The current user value is checked first; if it does not exist the local machine value is checked.

Value: AcceptDottedPrincipalNames

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Type: DWORD {0, 1}

Default: 1

Function: KFW_accept_dotted_usernames()

Kerberos v5 principal names are traditionally mapped to Kerberos v4 names by the AFS servers before they can be looked up in the Protection database. The mapping algorithm used permits collisions to occur. Both of the Kerberos v5 names, "user.admin@REALM" and "user/admin@REALM" are interpreted as the same user identity within the cell. To enable both names to be sent to the server by AFSCreds or Integrated Logon, set this value to 1.

Value: Use524

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Type: DWORD {0, 1}

Default: 0

Function: KFW_use_krb524()

When MIT Kerberos for Windows can be loaded, Kerberos v5 will be used to obtain AFS credentials. By setting this value to 1, the Kerberos v5 tickets will be converted to Kerberos v4 tokens via a call to the krb524 daemon. The current user value is checked first; if it does not exist the local machine value is checked.

Value: AfscredsShortcutParams

Regkey: [HKLM\SOFTWARE\OpenAFS\Client]

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Type: REG_SZ

Default: "-A -M -N -Q"

Function: Shortcut_FixStartup

This value specifies the command line options which should be set as part of the shortcut to afscreds.exe. afscreds.exe rewrites the shortcut each time it exits so as to ensure that the shortcut points to the latest version of the program. This value is used to determine which values should be used for command line parameters. The current user value is checked first; if it does not exist the local machine value is checked.

The following subset of the command line options is appropriate for use in this registry setting:

-A = autoinit
-M = renew drive maps
-N = ip address change detection
-Q = quiet mode. do not display start service dialog if afsd_service is not already running
-S = show tokens dialog on startup
-Z = unmap drives

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Value: Authentication Cell

Regkey: [HKCU\SOFTWARE\OpenAFS\Client]

Type: REG_SZ

Default: <none>

Function: Afscreds.exe GetDefaultCell()

This value allows the user to configure a different cell name to be used as the default cell when acquiring tokens in afscreds.exe.

Regkey: [HKCU\SOFTWARE\OpenAFS\Client\Reminders]

Value: <afs cell name>

Regkey: [HKCU\SOFTWARE\OpenAFS\Client\Reminders]

Type: DWORD {0, 1}

Default: <none>

Function: LoadRemind(), SaveRemind()

These values are used to save and restore the state of the reminder flag for each cell for which the user has obtained tokens.

This value used to be stored at [HKLM\Software\TransarcCorporation\AFS Client\AfsCreds].

Regkey: [HKCU\SOFTWARE\OpenAFS\Client\Active Maps]

Value: <upper case drive letter>

Regkey: [HKCU\SOFTWARE\OpenAFS\Client\ActiveMaps]

Type: DWORD {0, 1}

Default: <none>

These values are used to store the persistence state of the AFS drive mappings as listed in the [...\Client \Mappings] key.

These values used to be stored in the afsdsbmt.ini file

Regkey: [HKCU\SOFTWARE\OpenAFS\Client\Mappings]

Value: <upper case drive letter>

Regkey: [HKCU\SOFTWARE\OpenAFS\Client\Mappings]

Type: REG_SZ

Default: <none>

These values are used to store the AFS path in UNIX notation to which the drive letter is to be mapped.

These values used to be stored in the afsdsbmt.ini file.

A.4 OpenAFS Client Service Environment Variables

Value: AFS_RPC_ENCRYPT

Values:

"OFF" disables the use of RPC encryption
any other value allows RPC encryption to be used

Default: RPC encryption is on

Value: AFS_RPC_PROTSEQ

Values:

"ncalrpc" - local RPC
"ncacn_np" - named pipes
"ncacn_ip_tcp" - tcp/ip

Default: local RPC

A.5 AFS Redirector Parameters

The AFS Redirector is implemented with three components: %windir%\system32\drivers\AFSRedir.sys, %windir%\system32\drivers\AFSRedirLib.sys and %windir%\system32\AFSRDFSProvider.dll. These components provide the interface between the Windows Installable File System interface and the WNet application interface and the AFS file system. The

[HKLM\SYSTEM\CurrentControlSet\Services \AFSRedirector\Parameters]

Value: DebugFlags

RegKey: [HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\Parameters]

Type: REG_DWORD

Default: 0

Bit 0 (0x1): Trigger Debug Break on AFSRedir.sys start. Used for kernel debugging.

Bit 1 (0x2): Output trace logging to Kernel Debugger. Used for kernel debugging.

Bit 2 (0x4): Enable Force Crash Ioctl. Checked builds only. Used for force a BSOD.

Bit 3 (0x8): Enable Bug Check on all exceptions. Normally exceptions are caught by handlers. Used during testing.

Bit 4 (0x10): Reserved.

Bit 5 (0x20): Do not start the AFS Redirector if Windows did not perform a clean shutdown.

Value: TraceBufferSize

RegKey: [HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\Parameters]

Type: REG_DWORD

Default: 0 {0 .. 10000} (KBs)

Specifies the size of the circular trace log buffer allocated within kernel memory. 0 disables trace logging.

Value: TraceLevel

RegKey: [HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\Parameters]

Type: REG_DWORD

Default: 0 {0..4}

0: No logging; 1: Error; 2: Warning; 3: Verbose; 4: Maximum Verbosity

Value: TraceSubsystem

RegKey: [HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\Parameters]

Type: REG_DWORD

Default: 0

Bit 0 (0x1): I/O Subsystem

Bit 1 (0x2): File Control Blocks and Name Processing

Bit 2 (0x4): Lock Processing (requires Verbose or higher level)

Bit 3 (0x8): Extent Processing

Bit 4 (0x10): Worker Thread Processing

Bit 5 (0x20): Reference counting of directory entries

Bit 6 (0x40): Reference counting of objects

Bit 7 (0x80): Reference counting of volumes

Bit 8 (0x100): Reference counting of file control blocks

Bit 9 (0x200): Garbage Collection

Bit 10 (0x400): Pipe and share processing

Bit 11 (0x800): Directory notification interface

Bit 12 (0x1000): Network Provider support processing

Bit 13 (0x2000): Directory node count processing

Bit 14 (0x4000): PIOCTL processing

Bit 15 (0x8000): Authentication Group creation and assignment

Bit 16 (0x10000): Library load and unload, task queuing

Bit 17 (0x20000): Process creation and destruction

Bit 18 (0x40000): Extent Active counting

Bit 19 (0x80000): Redirector initialization

Bit 20 (0x100000): Name Array processing

[HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\NetworkProvider]

Value: Debug

RegKey: [HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\NetworkProvider]

Type: REG_DWORD

Default: 0

Set to 1 to log all AFSRDFSProvider Network Provider requests to C:\TEMP\AFSRDFSProvider.log. The C:\TEMP directory cannot be changed and must exist.

Value: Name

RegKey: [HKLM\SYSTEM\CurrentControlSet\Services\AFSRedirector\NetworkProvider]

Type: REG_SZ

Default: "OpenAFS Network"

This value defines the name displayed in the Explorer Shell and to which network drive mappings are made.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y