



OpenAFS for Windows Status Report: July 2008

1. Introduction.....	2
1.1. Supported Versions of Microsoft Windows.....	2
1.2. Important Use Cases	3
1.3. Usability	4
1.4. Administration Tools and Servers.....	4
1.5. Support for Future Operating System Releases	5
1.6. The Future of OpenAFS for Windows.....	5
1.7. A History of OpenAFS for Windows.....	6
2. Improvements in OpenAFS for Windows since 1.2.10	8
2.1. Internal Resource Management.....	12
2.2. SMB/CIFS improvements.....	12
2.3. Cache Manager Improvements	13
2.4. Installation Packaging	14
2.4.1. NSIS 2.0.....	14
2.4.2. WiX 2.0.....	14
2.4.3. Digital Signatures.....	15
2.5. Improved Build System and Development Tool Compatibility.....	15
2.6. Improved Windows Integration	15
2.6.1. UNC handling	15
2.6.2. Byte Range Locking.....	15
2.6.3. Integrated Login.....	16
2.6.4. Windows Explorer Shell	16
2.6.5. Microsoft Loopback Adapter	16
2.6.6. Select LAN Adapter by Name	16
2.6.7. Power Management.....	17
2.6.8. VPN Compatibility	17
2.6.9. Profile Data	17
2.6.10. Terminal Server and Citrix Compatibility	17
2.6.11. Windows XP SP2, 2003 SP1, 2003 R2, Vista and 2008 Support	17
2.7. Kerberos 5 Support	18
2.8. DNS AFSDDB Support	18
2.9. Dynamic Root Volume (Freelance Mode).....	19
2.10. Hidden Dot Files	19
2.11. Logging Changes	19
2.12. Graphical User Interface Tools	19
2.12.1. AFS Authentication Tool (afscreds.exe).....	19
2.12.2. AFS Control Panel Tool (afs_config.exe).....	20
2.12.3. Network Identity Manager Provider	20
2.13. Command Line Tools.....	21
2.14. Debugging Tools.....	21
3. Mobile Client and Network Address Translation Support	22
4. Quality Assurance	24
4.1 The Stress Test	24
4.2 Windows Quality Online Service (Windows Error Reporting)	25
4.3 End User Testing and Bug Reports	25
5. Known Issues	26
6. Future Implementation Roadmap	27
6.1 AFS Client Service Improvements.....	27

6.1.1 Native File System Replacement for SMB Server Interface	27
6.1.2 RX Connection Pools	28
6.2 Explorer Shell Extension Improvements.....	29
6.2.1 Custom Column Handler	30
6.2.2 Custom Context Menu Handler	31
6.2.3 AFS Property Sheets	31
6.2.4 AFS Tool Band	34
6.2.5 AFS Tool Tips Handler.....	34
6.2.6 AFS MetaData Handler.....	35
6.2.7 Name Spaces	35
6.3 OpenAFS Control Panel Replacement.....	38
6.3.1 AFS Group Editor Panel	38
6.3.2 Network Identity Manager AFS Provider Panel	39
6.3.3 Microsoft Management Console Shortcut.....	39
6.4 OpenAFS Client Service Microsoft Management Console Plug-in.....	39
6.4.1 Vista User Account Control Privilege Separation.....	47
6.5 AFS Servers on Microsoft Windows	48
7. OpenAFS for Windows Needs Your Support.....	50
7.1. Financial Contributions	50
7.1.1. Secure Endpoints Inc.	50
7.1.2. The USENIX OpenAFS Fund.....	50
7.2. Direct contributions of code and/or documentation.....	51

1. Introduction

This document summarizes the improvements in OpenAFS for Windows (OAFW) since Secure Endpoints Inc. accepted responsibility for its on-going development in October 2003. It also outlines the remaining known issues, describes requested features, and outlines a roadmap for future development plans.

As of 28 July 2008, the current OpenAFS for Windows production release is 1.5.51. The OpenAFS for Windows client implements all of the traditional AFS client functionality permitting access to data stored within AFS volumes and supports federated authentication via the use of Kerberos v5. OpenAFS provides Microsoft Windows users the benefits of a globally distributed location independent caching file system capable of storing Unicode object names. Integration with the Microsoft Windows environment is obtained for client authentication during the login process as well as via extensions to the Windows Explorer Shell allowing graphical manipulation of AFS access control lists, volume mount points, symlinks, and object properties. The AFS Client Service and its associated tools are regularly stress tested and hundreds of thousands of copies are in use. End users come from diverse communities including financial, manufacturers, web services, and medical, academia and research institutions.

1.1. Supported Versions of Microsoft Windows

OpenAFS for Windows 1.5.51 is supported on 32-bit versions of Microsoft Windows 2000, XP, 2003, 64-bit Microsoft Windows XP, 2003, and all versions of Microsoft Windows Vista including Service Pack 1 and Windows Server 2008.



<http://support.microsoft.com/?kbid=933305>

The support for Microsoft Windows Vista was [announced](#) the day after Microsoft released Windows Vista to volume license customers. Support for Windows Server 2008 Server and Vista SP1 were available before the official release dates.

1.2. Important Use Cases

With robustness issues left in the past, recent work has focused on the performance and scalability of the OpenAFS for Windows client. There are several scenarios that are of special interest:

- **Compilation of OpenAFS within AFS.** The AFS cache should provide the ability to achieve build times close to that associated with using local disk. This is because store operations are performed in the background and once an object file is stored it should not have to be read back from the file server for linking. Optimizing the quantity of data written to the file server, pre-fetching required data chunks, and minimizing the number of directory lookups are critical to good performance.
- **Google Indexing of AFS.** The AFS cache manager should be robust enough to support the indexing of the global AFS name space from a single client. Google indexing of AFS demonstrates the strength of the global AFS name space for use in federated collaboration. This scenario requires the cache manager to support thousands of AFS cells and high turnover of volume objects, directory entry status objects, and data buffers. Optimizing cell, volume, status, and data buffer lookup times are critical to good performance.
- **Dynamic Content Distribution to Web Servers.** One task that AFS excels at is distribution of relatively static data to large numbers of systems for further distribution via the web. There are many organizations that are evaluating AFS for distribution of video content. DVD videos have a typical file size of 4GB to 8GB which makes them particularly painful to fetch from a file server in order to serve an incoming web request. The AFS cache manager's ability to fetch only the requested byte ranges and recycle data buffers least recently used first ensures that a sufficiently large cache will always contain the most recently desired content. This removes the need to perform complicated analysis of access logs to determine how to best replicate content to meet the needs of end users. To support video content distribution with a working set of 1000 movies, the cache manager must be capable of storing between four and six terabytes of data buffers as well as thousands of volume and status objects. Optimizing cell, volume, status and data buffer lookup times are critical to good performance as are the ability tune the RPC chunk size and data buffer size to large values to maximize throughput.

1.3. Usability

Overall the OpenAFS for Windows client does a very good job of providing end users the ability to access files stored within the global AFS namespace. OAFW integrates with Windows to provide a Single Sign-On experience. AFS is accessible from both the command prompt and the Explorer Shell. However, there are improvements that must be made to provide users with the day-to-day experience that they should expect.

- Native Microsoft Windows file systems implement:
 - a mandatory locking model that can be applied to byte ranges
 - multiple data streams per object
 - extended attributes
 - referrals to Microsoft's Distributed file system

This functionality is simply not supported by the current AFS3 protocol and must be added.

- The use of a CIFS-to-AFS3 gateway architecture results in additional overhead for each request, limits the cache manager throughput and can result in undesirable application behavior if the CIFS Windows client fails a request because it took longer than the CIFS client anticipated. A native Windows File System Redirector and Network Provider would provide a ten to twenty times improvement in throughput and will eliminate unwanted timeout issues. Funding for this work has been obtained and completion is anticipated by the end of 2008.
- AFS ships with an Explorer Shell extension that provides a crude graphical interface for the manipulation of AFS constructs such as mount points, symlinks, quotas, and access control lists. A much better interface has been designed that will make AFS objects a seamless part of the user's daily experience. Implementation resources are still being sought.

1.4. Administration Tools and Servers

In addition to an AFS client, the OpenAFS for Windows distribution includes administration tools and the AFS servers (file, volume, pts, bos). These components have received very little developer attention. The AFS Server Management tool has been updated to work with Kerberos 5 and since the 1.5.22 release has been quite usable. However, it suffers from horrible performance in cells with large numbers of file servers.

The AFS User Manager tool is not being worked on as it is a dedicated kserver tool. In order for the User Manager to become a generic tool it should support a variety of Kerberos administration protocols including MIT Kerberos kadmin, Heimdal kadmin, Solaris kadmin, and Microsoft Active Directory.

As for the AFS Server processes, they run but are not being tested from release to release and should be considered unstable. The servers lack support for power management events, dynamic network configuration changes, and Windows volume management. Many of the design decisions are based upon the limitations of NT 3.5.

The AFS Server installation wizards have been disabled. They assume the use of kaserver and have some serious thread safety issues that frequently result in unresponsive behavior. At the present time it is advised that users deploy AFS servers on MacOS X, Solaris, or Linux potentially within a VMWare session running on a Windows Server.

1.5. Support for Future Operating System Releases

One of the primary concerns of Information Technology managers when selecting a centralized storage system is product longevity and timely support for future operating system releases. Since 2003, Secure Endpoints Inc. and the OpenAFS community have twice been challenged by major revisions in the Microsoft Windows operating system: Windows XP Service Pack 2¹ and Windows Vista². New releases of OpenAFS for Windows supporting the new architectures were available within a day of the official Microsoft release dates. Although there are no guarantees, OpenAFS for Windows is compatible with the 64-bit frameworks that are the basis for the future of Microsoft Windows over the next decade³.

1.6. The Future of OpenAFS for Windows

Secure Endpoints Inc. has published a road map⁴ for the continued evolution of the OpenAFS for Windows client including a new credential management interface built as an extension to Network Identity Manager⁵. As OpenAFS is an open source effort, contributions from organizations that use OpenAFS are crucial to its continued improvement. Completion of the roadmap is dependent upon resource availability. The eventual goal is for AFS to be a first class file system for Microsoft Windows operating systems.

Beyond the OpenAFS for Windows development that will improve the AFS user experience, there are many changes to the AFS servers that must also be implemented⁶. These include the deployment of a new security class based on GSSAPI that will bring military grade authentication and data secrecy; an extension to the Protection Server database to allow the multiple authentication names to be associated with AFS Identifiers; server side support of byte range locking and the mandatory locking model; directory format changes to support Unicode object names and multiple data streams per file; and performance enhancements to the RX remote procedure call library. These changes once implemented will require client side support before they become useful.

With the continued support of the OpenAFS community, all of these projects will be successfully accomplished.

¹ <https://lists.openafs.org/pipermail/openafs-announce/2004/000081.html>

² <http://www.openafs.org/openafs-vista-announce.html>

³ Microsoft has announced that Windows 7 which is rumored to be shipped in late 2009 will be the last Windows operating system shipping in a 32-bit form.

⁴ <http://www.openafs.org/openafs-windows-roadmap.html>

⁵ Network Identity Manager is an extensible multiple identity credential manager distributed as part of MIT Kerberos for Windows.

⁶ <http://www.openafs.org/roadmap.html>

1.7. A History of OpenAFS for Windows

On 31 Oct 2000, IBM released an open source version of their AFS for Windows product as part of the general OpenAFS release⁷. While the OpenAFS community made substantial improvements to AFS on the UNIX platforms, the Windows product languished until November 2003. Organizations with broad AFS deployments struggled with the question of how to access AFS from Microsoft Windows. Those that attempted to support Windows clients as first class AFS citizens were repeatedly burned. In response, many organizations have searched for an alternative distributed file system to migrate to although there are few (if any) available which provide matching scalability and volume management capabilities that also permit the same level of data availability during maintenance windows.⁸

In November 2003, Secure Endpoints Inc. began a concerted effort to stabilize OpenAFS for Windows and add new functionality. The goal has been to improve stability; performance; interoperability; end user transparency; ease of deployment; and integration with Kerberos 5 environments via use of MIT's Kerberos for Windows product. With each subsequent release since March 2004 the OpenAFS for Windows product has improved. As problems were reported by end users, they were debugged and corrected in the subsequent release.

Testing of AFS clients and servers over the years has been ad-hoc in nature. Transarc Labs is rumored to have tested their new releases by running them in the production andrew.cmu.edu cell. If there were no problems reported by end users, everything must have been ok. The lack of a robust environment for stress testing and a reliance on end users to deploy new releases in order to test them caught up with the AFS community in 2004. Race conditions between clients and servers were discovered in production code which resulted in repeated downtime throughout the community as hyper-threaded and multi-processor systems were deployed and the number of clients increased. Mobile clients and those behind network address translators⁹ also resulted in serious performance degradation.

In December 2004, MIT's Information Services and Technology group began developing a test suite for use in stress testing the OpenAFS for Windows client. The 1.3.81 release was the first version of OpenAFS for Windows capable of passing the stress test. As the quality of the OpenAFS client has been improved the stress test has been improved. The

⁷ <http://www-128.ibm.com/developerworks/opensource/library/os-afs.html>

⁸ One of the primary benefits of AFS is the minimal impact that occurs for end users during periods of server maintenance, load rebalancing, and system failure. Organizations that have attempted to migrate to other networked storage solutions have discovered that the outages for end users tend to be more frequent and of longer duration. AFS volumes can be moved from server to server while in use and can be restored to servers other than the original server in case of catastrophic outage. In addition, outage of a single file server cannot disable access to other resources in the AFS name space.

⁹ The 1.4.1 OpenAFS file server was the first to provide support for multiple AFS clients using the same IP address. The 1.4.2 release is the first release to support transparent migration of mobile clients from IP address to IP address without delay. The 1.5.17 OpenAFS for Windows client release is the first to address the inability to continue the use of RX connections when a network address translator modifies the apparent source port number.

OpenAFS for Windows Status Report: July 2008

stress test is now a standard part of the OpenAFS for Windows development process. The OpenAFS client is not only reliable and easy to deploy but its performance is comparable to its UNIX counterparts.

On 1 November 2005, OpenAFS released version 1.4.0. OpenAFS.org celebrates its fifth anniversary.

On 16 Feb 2006, OpenAFS released version 1.5.0, the first development release to support 64-bit Microsoft Windows operating systems and CIFS support of byte range locking.

On 10 March 2006, OpenAFS released version 1.4.1

On 11 June 2006, OpenAFS released version 1.4.2-beta-1 and development version 1.5.2., the first releases to support Microsoft Windows Vista and Longhorn Server from the command prompt.

On 6 September 2006, OpenAFS released version 1.4.2-rc3 and development version 1.5.8. The 1.5.8 build implements the ability to fetch status data in bulk, can read and write files greater than 2GB, and updates the CIFS server interface to support the Microsoft Windows Vista Explorer Shell.

On 1 December 2006, OpenAFS released version 1.5.12, the first release to support all of the Microsoft Windows Vista versions and receive the “Works with Windows Vista” logo approval.

On 15 February 2007, OpenAFS released version 1.5.15. This release fixed a previously unrecognized error that resulted in data corruption of files written to the file server. If the network connectivity to the file server is lost while background writes operations have been queued, the writes will not succeed and that data will not be written to the file server. This leaves a hole in the file filled with NULs. In the same release a design flaw that prevented the re-use of cached data after a write operation to same file was corrected resulting in significant improvements in the cache hit ratio.

On 19 March 2007, OpenAFS released version 1.5.17, the first release that does not mark servers down as a result of Network Address Translators altering the port mapping for the RX connection. OpenAFS clients no longer believe the AFS file servers are bouncing between up and down states.

On 20 September 2007, OpenAFS released version 1.5.25, which includes major scalability and performance enhancements compared previous releases. These include local directory updates to avoid re-reading directory contents from the file server; write dirty bytes instead of the whole buffer; many internal object lists optimized for least recently used recycling and hash table based lookups; B+ tree directory search; and 64-bit Kerberos support.

On 10 January 2008, OpenAFS released version 1.5.30, which includes improvements for 64-bit Windows; networking stack improvements for Vista and Server 2008; and data pre-fetch improvements.

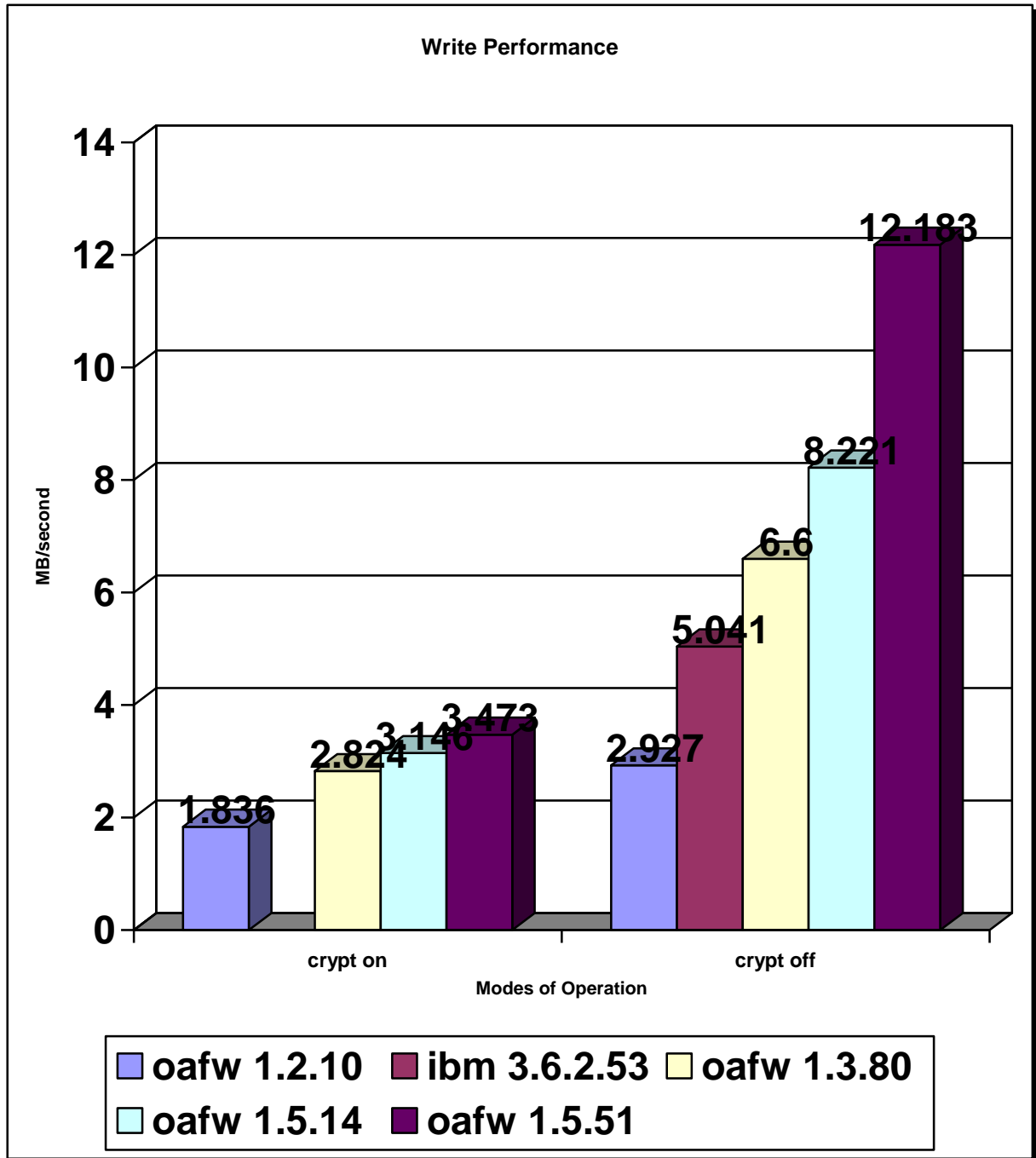
On 19 February 2008, OpenAFS released version 1.5.32 which more than doubled the throughput of the cache manager.

On 16 July 2008, OpenAFS released version 1.5.51 which added full support for the storage and access of Unicode object names within the AFS name space.

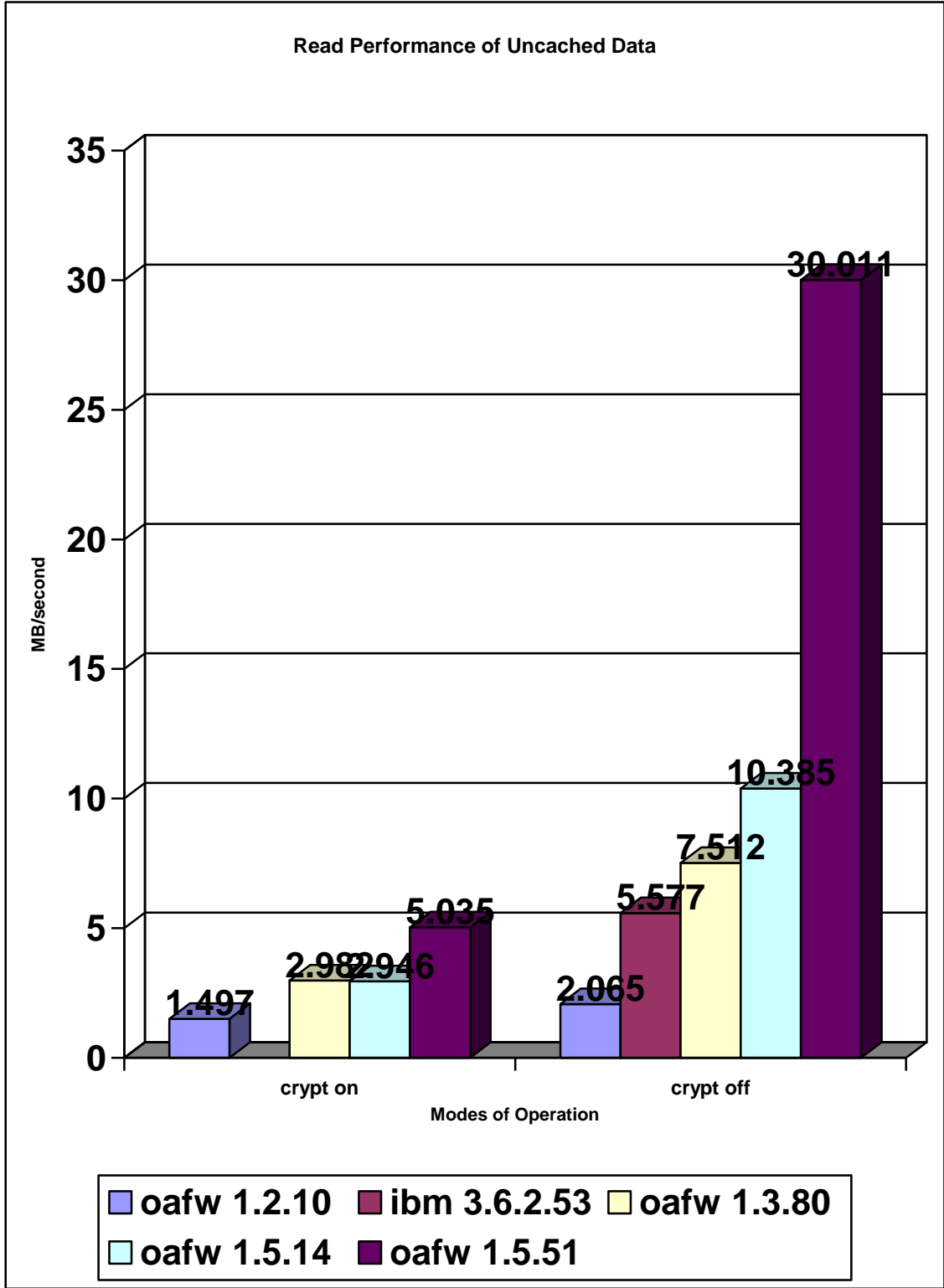
2. Improvements in OpenAFS for Windows since 1.2.10

Version 1.4.0 was a milestone release for OpenAFS for Windows and things have only gotten better since. There have been more than 800 improvements since the August 2003 1.2.10 release. The majority are changes to the client affecting stability, performance and Windows integration. The details are available in the [afs-changes-since-1.2.txt](http://www.openafs.org/dl/openafs/1.5.51/winxp/afs-changes-since-1.2.txt) file available at <http://www.openafs.org/dl/openafs/1.5.51/winxp/afs-changes-since-1.2.txt>. This section will focus on some of the highlights.

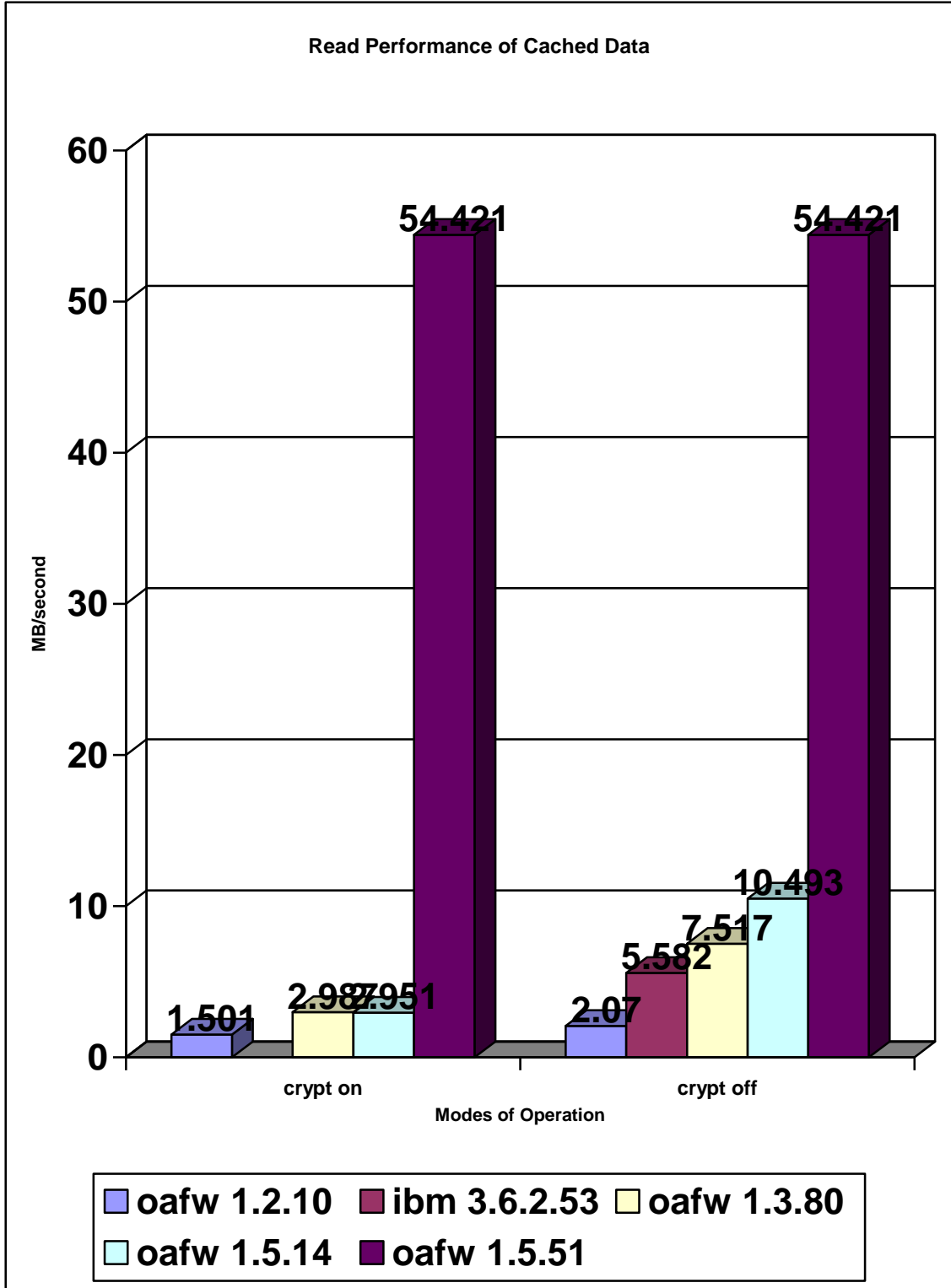
The resulting performance improvements can be summarized in the following charts. All of the tests were performed on a 32-bit XP system with a dual core Intel Xeon 3.4GHz CPU, 1GBit Ethernet, and 1GB of RAM and a 7200RPM SATA disk.



The write performance of OpenAFS for Windows has consistently improved.



Read performance of uncached data has consistently improved with each release. The cost of cryptographic operations within the RX protocol is a limiting factor.



Write operations on a file with cached data no longer invalidate the usability of the previously cached data. In older releases, after writing a file to AFS, read operations would be forced to read the data from the file server wasting precious network bandwidth.

2.1. Internal Resource Management

The OAFW 1.4.2 and earlier releases suffered from a large number of programming errors which adversely affect the stability and performance of the OpenAFS Client:

- use of un-initialized variables
- memory leaks due to reference count errors
- premature object destruction due to reference count errors
- kernel object leaks due to reference count and usage errors
- thread deadlocks due to improper lock management
- thread deadlocks due to a failure to wake up sleeping threads
- memory de-allocation errors
- queue management errors
- the number of allocated NetBIOS Control Blocks exceeded the number of objects Windows can monitor simultaneously resulting in NetBIOS operations which never complete
- race conditions between threads left data structures in invalid states
- the use of longjump() to restore the state of program registers is not safe to use in multi-threaded programs
- non-atomic combined pioctl/rpc operations produced race conditions between processes simultaneously setting or retrieving tokens from the AFS Client Service
- string table resources were improperly assigned index values resulting in memory corruption
- a failure to manage references to AFS RPC connections produced a “use once and discard” error which can overwhelm the file servers

2.2. SMB/CIFS improvements

- Extended SMB/CIFS messages were not supported resulting in file operations being rejected for a subset of files
- Authenticated connections between the Windows CIFS client and the OpenAFS SMB/CIFS Server are now negotiated using GSS SPNEGO. The actual authentication is performed using NTLM and the contents of the Windows logon cache.
- Unique virtual connection IDs are used for all connections
- Garbage collection of invalid or expired objects implemented
- Virtual Connection keep alive messages are periodically sent providing automatic detection of premature termination
- Virtual Connection termination forces cleanup of all open file handles and locks.
- Proper reporting of unsupported CIFS functions have been implemented
- Partial support for SMB/CIFS browsing has been added:
 - NETSHAREENUM
 - NETSHAREGETINFO
 - NETSERVERENUM2
 - NETSERVERGETINFO

- Force SMB/CIFS reconnects in the pioctl() library when Windows reports a downgrade attack error
- Properly follow soft symlinks
- Added support for hard symlinks
- Symlinks to [\\AFS\all\path](#) and [\\AFS\path](#) are now equivalent to /afs/path
- Directory Searches no longer produce invalid handle errors after 65536 FindFirst operations
- The use of foo.exe.local files or directories as a means of redirecting the location from which DLLs are loaded is now supported
- File times are now reported entirely in UTC. This prevents problems with backup software when switching back and forth from daylight savings time.
- Short file names (8.3 notation) were being generated using an algorithm that would produce invalid file names.
- Dynamic priority adjustments based upon the age of the outstanding CIFS request being processed.
- Multiple requests issued against the same object are completed in the order of receipt.
- Support for CIFS byte range lock requests
- Support for Unicode path names

2.3. Cache Manager Improvements

- Directory Name Lookup Cache is now case-sensitive
- Cell name comparisons are now case-insensitive
- New algorithm for computing filename pattern matches
- Memory utilization is now fixed
- Callback management improved
- Call timeout management improved
- Fixed Root Stat Cache entry initialization
- AFS RPC (RX) connections are no longer used once and discarded
- Cached data both stat and buffers are stored across AFS Client Service sessions
- UUIDs are now used to identify the AFS Client to the AFS servers. The UUID is kept across AFS Client Service sessions
- The lists of ACL entries no longer become corrupted
- IP addresses are no longer obtained at service startup and used for the life of the AFS Client Service. IP addresses are now obtained as needed allows the AFS client to report the correct set of IP addresses to the AFS servers upon request
- All AFS RPC callback interfaces are implemented including CM Debugging
- The default cache size has been increased to 96MB. The maximum cache size is 1.2 GB on 32-bit Windows and 512TB on 64-bit Windows.
- The default number of cache entries has been increased to 10,000.
- Volume and Bulk Callback revocations no longer deadlock (eventually causing the AFS Client Service to panic.)
- The logic used to determine if volumes are available, busy or offline has been fixed. Failover now works.

- The default @sys name list for 32-bit x86 systems is now “x86_win32 i386_w2k i386_nt40”. The default for Itanium is "ia64_win64" and for AMD X86-64 “amd64_win64”.
- Multi-homed servers are now supported.
- Threading optimizations reduce the number of locks that must be held for many operations. This increases the ability to pipeline operations and in turn improve performance especially of write operations.
- A number of race conditions and object reference errors in the RX library have been fixed. These improve the stability of the program.
- The RX library has been optimized to reduce the number of global locks. This improves the ability of AFS to take full advantage of multiprocessor and hyper-threaded systems.
- Byte range locks are managed by the client. In 1.5.2, allocated locks are backed by full file locks obtained from the AFS file server. AFS file locks which cannot be renewed block access to the file until the file is closed.
- AFS File Server capabilities are now queried.
- Universal AFS Error codes are now supported.
- Dirty buffer management optimized to reduce CPU utilization.
- Writing of dirty buffers is resistant to transient network or file server outages.
- When a file is locally modified, the previously cached data buffers associated with that file are no longer automatically invalidated.
- When a directory is locally modified, the cached directory buffers are updated to avoid re-reading the directory from the file server.
- Directories are locally converted into B+ trees to permit faster search times.
- Read only volume management is improved reducing the number of FetchStatus calls to the file server.
- Unicode object names are now supported. Unicode Normalization Form C is used during object lookup to ensure that names generated on MacOS X and other operating systems can be accessed on Microsoft Windows XP, 2003, Vista, and 2008 systems.

2.4. Installation Packaging

Two open source installation options are supported: NSIS and WiX.

2.4.1. NSIS 2.0

- Rob Murawski implemented a new executable installer using the open source Nullsoft Scriptable Installer Framework 2.0
- Supports new installs, uninstalls and upgrades from previous releases
- Designed for interactive installations by individual users

2.4.2. WiX 2.0

- Asanka Herath implemented an MSI installer for OpenAFS utilizing the open source WiX installation builder

- Designed for automated installation via Windows Group Policy but may be used interactively as well
- Supports new installs and uninstalls
- The OpenAFS.org distributed MSI can be customized by the use of MSI Transforms for use by all organizations without requiring the ability to build OpenAFS for Windows from source

2.4.3. Digital Signatures

- All binary files and installers distributed by OpenAFS.org are digitally signed by “Secure Endpoints Inc.” using a code signing certificate issued by Verisign.
- All digitally signed files are timestamped by the Verisign Timestamping Server
- Digital signatures may be used to ensure that installed files have not been replaced or modified

2.5. Improved Build System and Development Tool Compatibility

- The latest Microsoft Development Tools are now supported
 - Visual Studio .NET 2003
 - Visual Studio .NET 2005 (aka Visual Studio 8)
 - Visual Studio 2008 (aka Visual Studio 9)
- Windows Server 2003 SP1 SDK required
- Windows Device Driver Kit 6.0 required
- Only Windows 2000 and above. Windows 9x no longer supported

2.6. Improved Windows Integration

2.6.1. UNC handling

- UNC paths of the form [\\afs\cellname](#) are now supported when the Microsoft Loopback adapter is installed
- The “NetbiosName” registry value can be used to specify alternatives to “afs”
- No longer need to use [\\afs\all\cellname](#)
- UNC paths of the form [\\afs\cellname#volume](#) are now supported permitting direct access to any volume. % can be used instead of # to force the use of the read-write volume. The volume name can be replaced with the volume id.

2.6.2 Byte Range Locking

- All versions of AFS for Windows prior to 1.4.1 and 1.5.0 would grant a lock to the requestor whether or not such a lock could be obtained.
- OAFW 1.4.1 and later locally manage lock allocations but do not back the allocations with AFS file server locks.
- OAFW 1.5.0 and later locally manage lock allocations but only grant locks that are backed by AFS file server locks unless the user access is no better than “rl” or the volume is read-only.

2.6.3. Integrated Login

- Kerberos 5 is used to obtain tokens when MIT Kerberos for Windows is installed.
- All uppercase user names authentication attempts are retried using all lowercase upon failure
- Closed security hole which leaked plaintext passwords on the wire
- WinLogon Event Notification handler added to destroy AFS tokens at logout
- Domain specific configuration is now supported. This permits a separate configuration to be used for logging in via a local machine account vs a Windows domain account vs an MIT Kerberos principal.
- The “TheseCells” registry key enables the retrieval of AFS tokens for multiple cells.
- Timeout processing has been fixed. A request to ‘retry’ by the end user will now wait for a full timeout period.
- If the service is in the START_PENDING state, login will not timeout until the state changes.
- Kerberos 5 tickets obtained during the login process are now preserved and passed into the user’s logon session for storage in the user CCAPI credential cache. (Use MIT Kerberos for Windows 3.1 or above.)

2.6.4. Windows Explorer Shell

- AFS UNC paths are now supported in the Explorer
- Browsing of the “AFS” Server is now supported (limited to 13 character names)
- The AFS Context Sensitive Popup Menu works on all files and directories located within the AFS name space
- When the AFS Client Service is disabled, the AFS Shell Extension is dynamically disabled to prevent performance delays

2.6.5. Microsoft Loopback Adapter

- Installed by both OpenAFS.org installers
- Provides a locally visible adapter to bind the SMB/CIFS Service Name
- Provides an adapter for the AFS Client Service to bind to when network connectivity is not available
- Prevents the AFS Client Service from halting due to dynamic reconfiguration of plug and play network devices

2.6.6. Select LAN Adapter by Name

- The display name of the LAN Adapters can be used as a means of specifying which LAN adapter should be used by the AFS Client Service.
- Simply name the desired LAN Adapter “AFS”
- This functionality may be disabled using the “NoFindLanaByName” registry value

2.6.7. Power Management

- Receipt of Standby, Suspend or Shutdown notifications by the AFS Client Service force all dirty buffers in the cache to be written back to the server (when possible).
- Receipt of Standby or Suspend notifications by the AFS Client Service result in the network adapter binding being released. A Resume notification will cause the AFS Client Service to bind to a new network adapter.

2.6.8. VPN Compatibility

- OpenAFS for Windows was found to be incompatible with the Cisco IPsec VPN client older than version 5.0.
- In order for AFS RPC requests to pass through the Cisco IPsec VPN version 4.x, the maximum size of Rx packets must be kept no larger than 1292 bytes.
- Use the “RxMaxMTU” registry value to 1260 to provide compatibility.

2.6.9. Profile Data

- HKLM\Software\OpenAFS\Client key used to set system default values
- HKCU\Software\OpenAFS\Client key used to store user configuration data
- Used for:
 - Token Expiration Reminders
 - Use of MIT Kerberos for Windows for Kerberos 5
 - Use of the Kerberos 5 to Kerberos 4 translation service
 - Show Tray Icon (afscreds.exe auto start)
 - afscreds.exe shortcut parameters
 - Freelance data (mount points and symlinks)
 - Submount entries
 - Drive mappings
 - Default Authentication Cell
 - Windows' SMB Client Side Caching configuration

2.6.10. Terminal Server and Citrix Compatibility

- All configuration files have been removed from the %WINDIR% directory
- All user configuration data is now stored in the per-user registry allowing for multiple users and user instances
- SMB/CIFS sessions are authenticated to the logon session thereby removing the need for the random SMB Names utilized by “High Security” mode to enforce token ownership separation
- Corrected detection of the current logon user name
- Power Management support is multi-user safe

2.6.11. Windows XP SP2, 2003 SP1, 2003 R2, Vista and 2008 Support

- Sets a magic registry value to permit the use of SMB/CIFS Service Names which do not match the local hostname

- Sets a magic registry value to allow GSS SPNEGO authentication over a loopback connection
- Communicates with the Windows Integrated Firewall to dynamically open the ports used for callback messages.
- DLLs no longer initialize the AFS RPC library in DllMain entry-point which caused the Integrated Logon support code to block the successful startup of the Microsoft Windows operating system.
- OpenAFS can now be used in multi-domain Windows forests when users log in with a Kerberos 5 principal from a non-Windows realm. (Roaming profiles cannot be used in such a configuration due to bugs in Windows XP.)
- Support for 64-bit Windows is available in all 1.5.x releases.
- Support for Windows Vista and 2008 Server is available in 1.5.12 and later releases.

2.7. Kerberos 5 Support

- Integrates with MIT Kerberos for Windows 2.6.5 and greater. KFW 3.0 and later ships with the Network Identity Manager. An AFS provider for NetIDMgr is distributed as part of OpenAFS for Windows release 1.5.12 and above. The recommended version of KFW to use is release 3.2.2. Use of the AFS Authentication Tool (afscreds.exe) is discouraged.
- The NetIdMgr AFS provider enables automated token acquisition for multiple cells from a single network identity. The acquisition mechanism (Kerberos v5 native, Kerberos v5 to v4, or Kerberos v4) can be specified individually for each cell.
- Integrated Logon:
 - Obtains tokens using Kerberos 5 (Kerberos 5 to Kerberos 4 conversion is not used by default but can be enabled globally with a registry value.)
 - Kerberos v5 use can be disabled for the entire machine
 - For each Windows logon domain a separate Kerberos v5 realm can be specified for Kerberos v5 authentication.
- The AFS Authentication Tool:
 - Imports credentials from both the MSLSA and CCAPI credential caches
 - The AFS System Tray tool (afscreds.exe) automatically renews tokens and tickets as they approach expiration
 - Tokens can be obtained for multiple cells from a single Kerberos 5 TGT (limited User Interface functionality)
 - Kerberos v5 use can be disabled either per machine or per user (no UI)
- Maximum token size increased to 12,000 bytes to permit the use of large Kerberos v5 tickets issued by Windows 2003 Active Directory

2.8. DNS AFSDb Support

- Configuration information for cells not specified in the CellServDB file may be discovered via DNS
- Use of DNS AFSDb records is enabled by default
- DNS support extended to all operations which referenced the CellServDB file

- Windows DNS Query API now used instead of home grown implementation
- Controlled by “UseDNS” registry value
- AFS Server records obtained via DNS AFSDDB are valid for the DNS AFSDDB time-to-live period.

2.9. Dynamic Root Volume (Freelance Mode)

- On by default
- Fixed initialization code
- Added support for read-write mount points
- Added support for symlinks
- Stores locally defined mount points and symlinks in the Registry
- Configurable via “FreelanceClient” registry value
- Timestamp from most recent update to mount point data used for all mount point stat entries
- Algorithm used for detecting the fake root.afs volume replaced to avoid conflicts with existing deployed cells

2.10. Hidden Dot Files

- Following Unix tradition, files/directories whose names begin with a period are given the Hidden attribute when the “HideDotFiles” registry value is set
- This functionality is enabled by default

2.11. Logging Changes

- afsd_init.log and afsd.log moved to the %TEMP% directory (usually %WINDIR%\TEMP for the SYSTEM account)
- Stack Trace data logged to afsd_init.log during assertion failure or unhandled exceptions
- The maximum size of the afsd_init.log file is now restricted to either 100Kb or the value specified by the “MaxLogSize” registry value
- “fs trace” logging defaults to off for release builds and on for debug builds.
- “fs trace” logging can be configured to write to the Debug Output stream via a registry setting.
- “fs trace” logging also controls logging of AFS RPC debug output to the Debug Output stream.
- All log files use CR-LF end of line so that they can be viewed in notepad.exe.

2.12. Graphical User Interface Tools

2.12.1. AFS Authentication Tool (afscreds.exe)

- Use of afscreds.exe is discouraged. Network Identity Manager and the AFS Provider implement a superior end user experience.
- No longer requires administrator account to operate
- Uses Kerberos v5 instead of the kauth library’s Kerberos v4 implementation to obtain tokens when MIT Kerberos for Windows is available

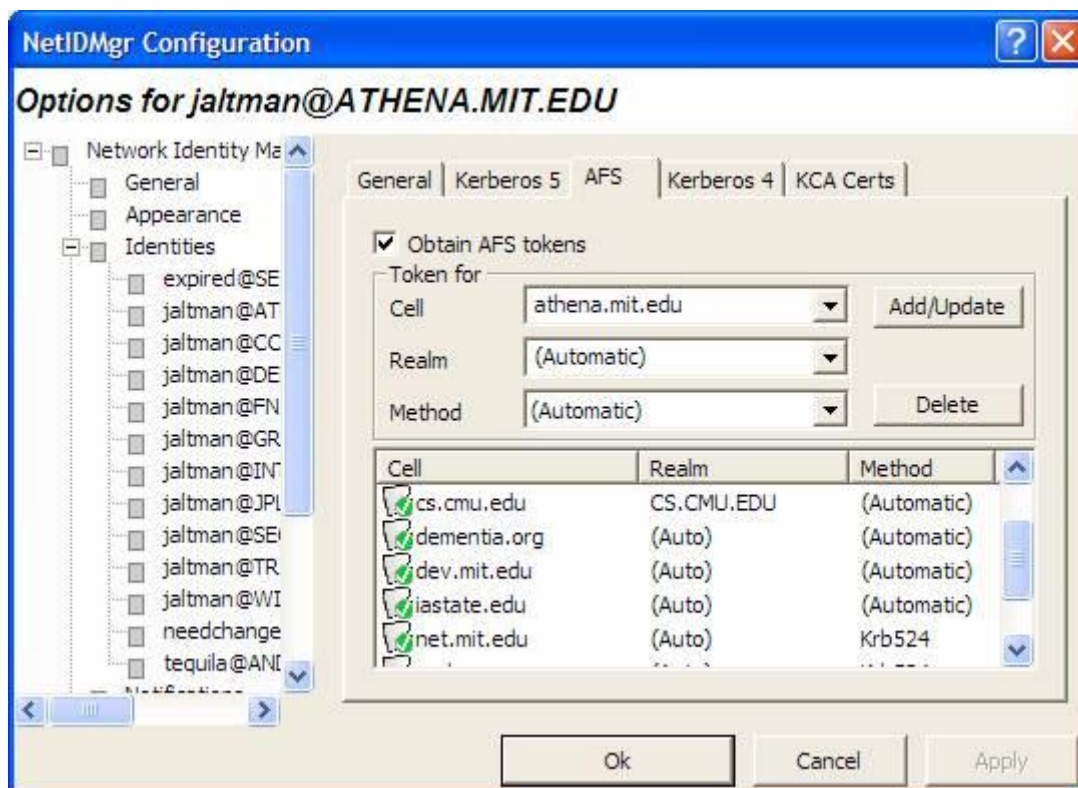
- Automatically renews Kerberos v5 tickets and derived tokens prior to expiration
- Supports multiple Kerberos v5 principals
- A single Kerberos v5 principal may be used to obtain tokens for multiple cells without repeated password entry (including environments which require Kerberos 5 cross realm authentication)
- Monitors network connectivity by watching for IP address changes and probing the KDC. If network access is restored and the AFS Client Service has not been started, the service will be started. If network access is restored and the user does not possess any valid tokens, the user will be prompted to obtain tokens.
- Drive mappings are restored upon startup
- PTS registration of new users to foreign cells has been added

2.12.2. AFS Control Panel Tool (afs_config.exe)

- Fields which modify the AFS Client Configuration now require logon account membership in the Windows “AFS Client Admin” group.

2.12.3. Network Identity Manager Provider

The Network Identity Manager replaces the former Kerberos for Windows ticket manager, Leash”, which when combined with the AFS Provider, distributed as part of OpenAFS, is intended to 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 AFS Provider can easily manage AFS tokens for multiple cells from one or more Kerberos v5 identities.



2.13. Command Line Tools

- aklog.exe has been added
- afsshare.exe modified to support the new registry entries
- several coding errors which would result in crashes fixed in fs.exe and vos.exe
- **fs cspolicy** command added
- UNC paths supported in fs commands
- cmdebug.exe has been added
- afstdacl.exe has been added
- “vos.exe listvol –format” is now supported
- **fs uuid** command (1.5.8) allows the AFS client UUID to be changed on the fly

2.14. Debugging Tools

- **cmdebug**
- **fs minidump** command added to generate Debugging Mini Dump files without requiring a debugger.¹⁰
- **fs trace**
- OutputDebugString
- Debug Symbols

¹⁰ On Windows 2000, the Microsoft Debugging Tools for Windows must be installed on the machine in order for this command to work. On Windows XP and above, the required debugging APIs ship as part of the operating system.

3. Mobile Client and Network Address Translation Support

AFS was originally designed for a world in which clients were assigned unique IP addresses which did not change over time. The addition of the *WhoAreYou* and *TellMeAboutYourself* Cache Manager RPCs combined with the assignment of Universally Unique Identifiers (UUIDs) to each AFS client in AFS3 were meant to separate the identity of the client from its address. The AFS file server implementations used UUIDs in very limited ways and as a result AFS file servers and clients continued to experience performance issues associated with cache manager callbacks that other file systems such as CIFS and NFS did not.¹¹

In order to improve the response time of repetitive read operations, AFS unlike other network file systems employs a client side cache. When a client reads directory information (performs a *FetchStatus* RPC), a callback is registered with the file server. When directory contents change, the file server contacts the registered clients to notify them that the affected contents of the cache must be invalidated. Performance problems occur when the *Callback* RPCs cannot be successfully completed.

The performance problems can be categorized as follows:

- AFS volumes could not be released while outstanding callbacks could not be delivered to registered clients. This was fixed in OpenAFS 1.4.0 by a change to the file server. Instead of blocking the release of a volume until all callbacks could be delivered, the clients with undelivered callbacks are flagged. The next time the client is heard from, the callback is delivered before any other file server requests can successfully complete.
- NATs permit multiple clients to contact the file server from the same IP address. The file server only tracked clients by IP address and ignored the port number. Although the file server was able to distinguish two clients by their UUID, the file server was unable to maintain state information for more than one client. This prevented the AFS clients from being able to properly maintain the contents of their caches. This was fixed in OpenAFS 1.4.1 by tracking clients by both their IP address and port number.
- AFS clients which migrate to a new IP address or port¹² do not receive callbacks from the file server and do not notice directory changes until either the directory status expires or the client attempts to modify the contents of the directory. This prevents the file server from being able to contact the client. As of OpenAFS 1.4.1, the Windows client pings the file servers once every ten minutes in order ensure that directory changes will be detected.

¹¹ CIFS and NFS do not use a callback mechanism to enable the file server to notify the client of directory and file modifications.

¹² NATs allow multiple machines to share one public IP address. They do so by mapping external port numbers to internal IP addresses and ports. These mappings are transient and frequently have a lifetime of less than five minutes. If no outbound communications take place using the port mapping, the mapping is removed. Subsequent outbound communications result in a new port mapping being assigned.

OpenAFS for Windows Status Report: July 2008

- AFS client which have migrated to a new IP address or port and contact the file server after a callback failure has occurred have experienced 56 second delays while the AFS file server attempts to contact the client on its old address. This was fixed in OpenAFS 1.4.2 by replacing the callback connection as a side effect of migration detection.
- AFS clients sitting behind a NAT that has UDP port mapping idle timeout shorter than ten minutes would experience AFS servers bouncing between the up and down states. This was a side effect of the RX protections against connection hijacking. When the source port changes, the RX connection must be replaced. As of 1.5.17, client's that do not receive a response from an AFS server retry the request with a new RX connection before marking the server down.

As of OpenAFS 1.5.17, client mobility and network address translation should no longer be a concern for AFS users.

4. Quality Assurance

The quality of OpenAFS for Windows releases is ensured through stress testing, examination of crash reports collected via Microsoft's *Windows Quality Online Service* and End User testing and bug reporting.

4.1 The Stress Test

The OpenAFS for Windows stress test developed by MIT's Information Services and Technology (IS&T) group is modeled on the Samba Team's SMB torture test. The test engine allows for scripted operations to be performed against an SMB server by a client. A set of test files is provided along with scripted operations which are designed to simulate the behavior of a large number of popular Windows applications including Microsoft Office, the Paradox database, Corel Draw, and many others.

The test engine allows for multiple client processes to be started. Each process in turn can be configured to execute separate instances of the script in independent threads. The processes can be started with a specified delay between each one to ensure that the widest range of simultaneous operations are being performed against the SMB server at a time

This test engine is useful for testing the OpenAFS for Windows client because the AFS Client Service is essentially an SMB to AFS gateway. Sitting between an SMB server and the AFS servers is an AFS cache manager. SMB operations are mapped to cache manager operations which in turn result in AFS remote procedure calls being issued against the AFS servers.

IS&T designed the test engine to integrate with MIT's AFS locker and Moira administration tools. In addition, they implemented reporting functionality that allows AFS trace log output to be triggered when errors are detected during test runs.

IS&T executes the stress test on a variety of platforms including Windows 2000 workstation, Windows XP workstation, Windows 2003 server, Windows 2000 Citrix Terminal Server. The Windows 2003 Server and 2000 Citrix Terminal Server machines are dual-processor systems. One of the Windows XP workstations is a dual-processor hyper-threaded system providing the equivalent of a 4-way system.

A typical run on the 2003 Server would utilize ten processes each with ten threads of operation against a volume which is released (taken offline) every 15 minutes and cloned once an hour. The ability to execute 100 simultaneous threads each performing simultaneous operations is an important milestone because the Windows architecture limits the number of simultaneous SMB operations to 63. The fact that 50% more operations could be successfully queued and processed without data loss is a major achievement.

On the Windows 2000 Citrix Terminal Server the typical test scenario would include starting three test processes per user session with each process reading/writing to a

different volume. Like the previous tests, the servers were configured to release the volumes every fifteen minutes.

All clients prior to 1.3.81 would at some point during the testing reach a deadlock condition or trigger a reference count assertion. It took four months of testing to shakeout the entire set of known deadlock and assertion conditions. The end result is a fast and robust client. As time goes on additional testing will be performed to ensure that new errors are not introduced. This statement is not meant to indicate that OpenAFS for Windows is bug free. No software is. The claim is simply that OpenAFS for Windows is significantly more robust than any prior AFS client and it can be trusted to work without issues under all known circumstances.

4.2 Windows Quality Online Service (Windows Error Reporting)

As of Microsoft Windows XP, Microsoft has begun to automatically collect mini-dumps of processes and device drivers that crash. These dumps are then provided to the application author to assist in improving the quality of the application. Secure Endpoints Inc. is registered with Microsoft and receives all of the crash reports. Error reports for resolved problems or from non-current AFS releases result in the user being directed to a page at the Secure Endpoints Inc. web site specifying where the user can obtain the necessary corrective action.

Since August 2006, Windows Error Reporting has become the primary method by which bugs are reported against the OpenAFS for Windows client. Windows Error Reports provide details that end users cannot collect. All bugs identified by a report are fixed in the subsequent release.

4.3 End User Testing and Bug Reports

As with any software product, OpenAFS for Windows does occasionally introduce new errors. End user testing and reporting of discovered bugs is a critical requirement for bug fixing. Most errors once reported are fixed within 48 hours of reproducing them.

The majority of end user reports originate with organizations that have purchased support contracts. Organizations without support contracts are reluctant to take the time to file trouble reports.

5. Known Issues

The 1.5.51 release of OpenAFS for Windows is a stable and functional AFS client which provides 32-bit and 64-bit Windows 2000, XP, 2003, Vista and 2008 systems access to the AFS global file system space. However, there remain a number of deficiencies:

- The User Interface is sorely lacking
 - Poor separation of functionality between tools. On Windows Vista and Server 2008, the appropriate separation of administrative and non-administrative functions is required in order to support the User Account Control¹³ model.
 - No GUI configuration for integrated logon options
 - No GUI configuration for all of the new functionality since 1.2
- The Microsoft Loopback Adapter is required for consistent operation.¹⁴
- SMB/CIFS implementation deficiencies:
 - Path names restricted to 256 characters
 - Share names restricted to 13 characters
 - No support for Microsoft Dfs Referrals
 - Remote Administration Protocol is incomplete
 - Per message integrity protection and authentication (digital signatures) is not supported
- No file permission integration with Windows Security model
- Performance and robustness can be improved by replacing the SMB/CIFS gateway server with a native File System driver running in kernel mode.¹⁵
- No disconnected mode functionality similar to the Windows SMB/CIFS Client Side Caching (aka Offline Folders)
- The strength of data confidentiality and integrity protection provided for use by AFS RPC calls leaves much to be desired. (fcrypt is a weakened version of DES.)
- DOS Attributes such as Hidden and System cannot be associated with files stored in AFS
- Extended Attributes associated with files stored in AFS are lost
- Multiple data streams are not supported. Streams are increasingly used by Microsoft and third parties to store meta-data associated with the file. This meta-data is used to enhance search capabilities and enforce security boundaries.
- Drive mapping within the AFS Authentication tool requires “AFS Client Admin” group membership¹⁶

¹³ <http://www.microsoft.com/technet/windowsvista/library/0d75f774-8514-4c9e-ac08-4c21f5c6c2d9.mspx>

¹⁴ The MLA is installed by default and is necessary for portable \\AFS UNC names, this is rarely an issue. Some organizations deploy remote administration tools that are dependent on workstation reporting of the IP address and which are incapable of filtering out loopback devices. At these organizations, use of the MLA is often discouraged even though the proper fix would be to correct the behavior of the administration tools.

¹⁵ CIFS implements a hard timeout of 45 seconds. If AFS operations take longer than 45 seconds, the CIFS client will terminate the virtual circuit to the AFS Client Service which forces the destruction of file handles and locks.

¹⁶ Drive mapping is best performed via the Explorer Shell.

6. Future Implementation Roadmap

At the 2004 AFS Best Practices Workshop, held at Stanford Linear Accelerator Center in March 2004, it was the consensus of the attendees that the future growth of AFS was dependent upon the availability of a Windows client which is secure, robust, fast, and transparent to the end users. With 95% of the world's desktops running Microsoft Windows, if the Windows clients are not robust then supporting end users will be a nightmare; if the Windows clients are slow and inefficient then the load on the AFS servers will be too high; if the Windows clients do not integrate transparently with the operating system then the users will become frustrated and will use something else to do their job; if the client is not secure then organizations with data confidentiality and integrity requirements cannot deploy AFS.

The following are some features that are on the wish list for future OpenAFS for Windows releases and their estimated cost to implement. This list only includes items that can be implemented without changes to the AFS Servers (File, Volume, or Protection). The [OpenAFS.org road map](#) and describes proposed modifications that require server modifications.

6.1 AFS Client Service Improvements

The AFS Client Service has come a long way in the last three years. For a summary of the progress please read the latest [OpenAFS for Windows Status Report](#). Still, there are many things that can be done to improve the user experience and performance of the product.

6.1.1 Native File System Replacement for SMB Server Interface

The existing OpenAFS Client relies on an SMB server implementation (similar to Samba) to export the AFS name space to Windows Applications. This has a number of negative side effects that would be avoided if the OpenAFS for Windows client were to be implemented via a combination of Network Redirector and File System Filter drivers. The current OpenAFS client on Windows is not a true Windows file system. Instead it operates as a SMB translator service. The Windows OpenAFS client creates a SMB fileserver on the client machine, and Windows accesses this SMB server as a normal Windows shared volume. For each I/O operation made to this virtual SMB server, the OpenAFS client translates the SMB request into a comparable operation on the OpenAFS fileserver. This impacts the Windows OpenAFS client in a number of negative ways:

The semantics of the CIFS file system are different than the semantics provided by OpenAFS. Because Windows sees the OpenAFS file system as an CIFS share, it has no way of acquiring the true capabilities or semantics of OpenAFS. This causes some applications to perform poorly when they expect the semantics of CIFS, which OpenAFS does not necessarily provide.

The use of the translator service requires data to be received by the OpenAFS client via RX, translated into SMB packets, and then sent over the virtual loopback interface to the actual SMB server on the same machine. This results in a number of extra data copies, which greatly reduces OpenAFS performance. Making OpenAFS a native Windows file system will reduce the number of data copies and protocol translations, which will increase performance.

The CIFS/SMB protocol does not provide any mechanism for the server to inform a client that an operation is actively being processed even if it is taking a long time to complete. The CIFS clients in Windows 2000 and above implement a dynamic timeout algorithm that estimates how long a request should take based upon the prior response time of the server and the amount of data being transferred. As the OpenAFS SMB server and cache manager are local to the machine, it is frequently the case that the response time is on the order of hundreds of microseconds. When a request to read or write large amounts of data from/to a file server occurs or if the needed volume is temporarily busy, the CIFS client will frequently timeout the request and tear down the SMB virtual circuit. This has a negative impact on applications as it results in all file handles being invalidated and all locks being dropped which must then be re-established.

In order for the UNC server name "AFS" to be visible on all clients, the Microsoft Loopback Adapter (MLA) must be installed in order to provide a private network adapter to which the "AFS" Netbios name can be bound. The installation of the MLA negatively affects several popular software licensing and anti-spyware products which use the network adapter MAC address as a unique key.

The solution is to replace the SMB server with a native Windows File System Redirector that can be supported on Windows XP SP2, Windows 2003 SP1, Windows XP 64, Windows 2003 R2, Windows Vista, and 2008 Server.

Estimated implementation time: 18 to 20 months.

Funding for this project has been obtained. Implementation is expected to be completed by the end of 2008.

6.1.2 RX Connection Pools

The AFS client communicates with the AFS File Server using the RX Remote Procedure Call library. In order to prevent one user on a multi-user machine from starving all of the other users, RX only permits four outstanding calls per RX connection. There is typically one RX connection per user context. This restriction can result in a significant bottleneck when the AFS client service is only being used to serve a single user (or service).

Theoretically it is possible to support multiple RX connections per user context and thereby get around this restriction. While this functionality would never be enabled by default, under specific circumstances it could come in quite handy.

Estimated implementation time: 60 to 80 hours

6.2 Explorer Shell Extension Improvements

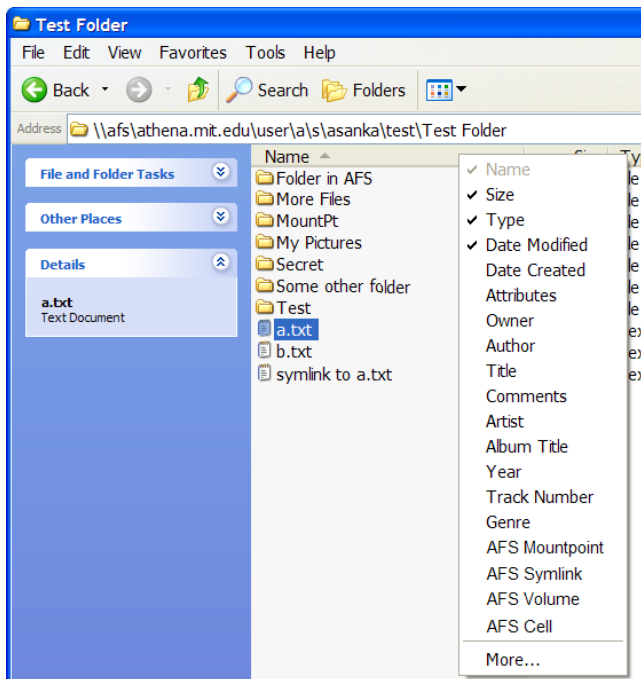
The existing AFS Shell Extension adds an "AFS" menu when the current folder or selected objects exist within the AFS name space. This menu is a hodge-podge of items. A few of the items appropriately belong on a context menu but are in the wrong place and most of the items should be Property sheets or should display the information to the end user in a different manner. The following is a proposal for a replacement.

6.2.1 Custom Column Handler

OpenAFS should provide a custom column handler to introduce several new detail columns that would display additional information about objects within AFS. For example, columns can be provided to optionally display and perhaps edit:

- symlink details
- mount point details
- AFS FID
- AFS owner
- AFS group
- AFS cell
- AFS volume

Name	Size	Type	Date Modified	AFS Owner	AFS Group
Folder in AFS		File Folder	12/20/2006 1:26 AM	asanka	mit
More Files		File Folder	12/19/2006 8:21 PM	asanka	mit
MountPt		File Folder	12/19/2006 8:21 PM	asanka	mit
My Pictures		File Folder	12/19/2006 8:21 PM	asanka	mit
Secret		File Folder	12/19/2006 8:20 PM	asanka	mit
Some other folder		File Folder	12/19/2006 8:20 PM	asanka	mit
Test		File Folder	12/19/2006 8:19 PM	asanka	mit
a.txt	1 KB	Text Document	1/2/2007 3:38 PM	asanka	mit
b.txt	0 KB	Text Document	1/2/2007 7:41 PM	asanka	mit
symlink to a.txt	1 KB	Text Document	1/2/2007 3:38 PM	asanka	mit



Estimated implementation time: 18 to 24 hours

6.2.2 Custom Context Menu Handler

Instead of the existing handler that simply adds an "AFS" menu containing everything, the new handler will selectively modify the existing context menu's behaviors when the current folder is located within AFS.

- The "New" submenu will be extended with "Symlink" and "Mount Point" menu items when the Current Folder is active.
- The "Delete" menu item will be removed whenever a symlink or mount point is selected.
- A "Remove Symlink" menu item will be added whenever all the selected items are symlinks.
- A "Remove Mount Point" menu item will be added whenever all the selected items are mount points.
- A "Flush File/Dir" menu item will be added.
- A "Flush Volume" menu item will be added.

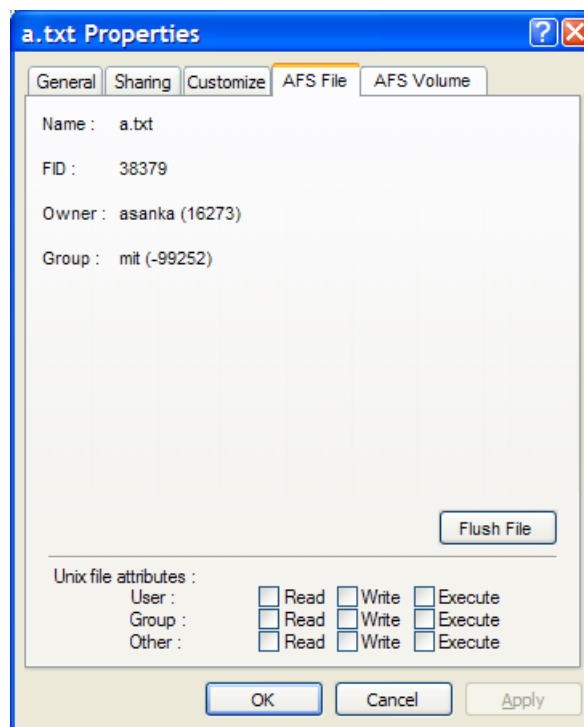
Estimated implementation time: 12 to 16 hours

6.2.3 AFS Property Sheets

A variety of property sheets will be added. Some will be context sensitive and others will be general tools.

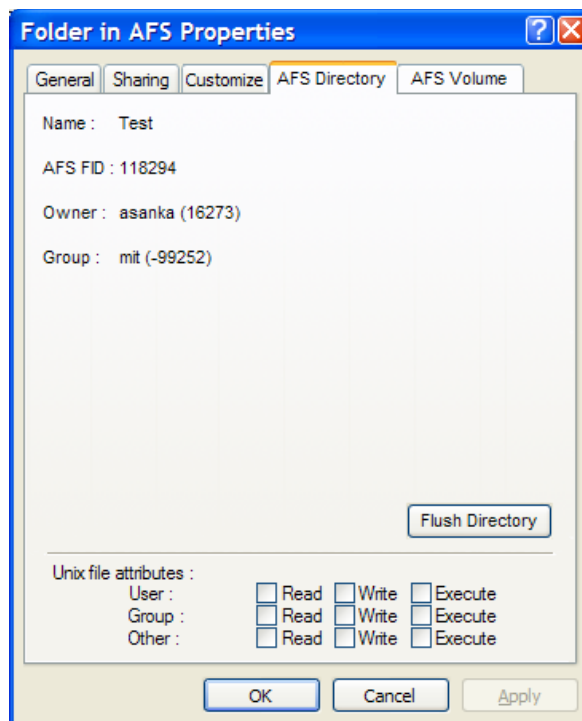
An "AFS File" sheet

- File name
- AFS FID
- Owner name and AFS ID
- Group ID
- UNIX file attributes
- Flush File from cache button



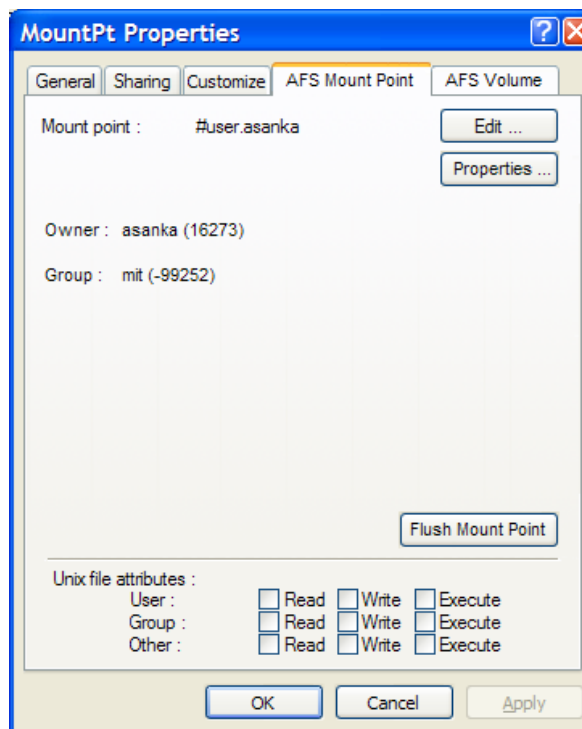
An "AFS Directory" sheet

- Directory name
- AFS FID
- Owner name and AFS ID
- Group ID
- UNIX file attributes
- Flush Directory button



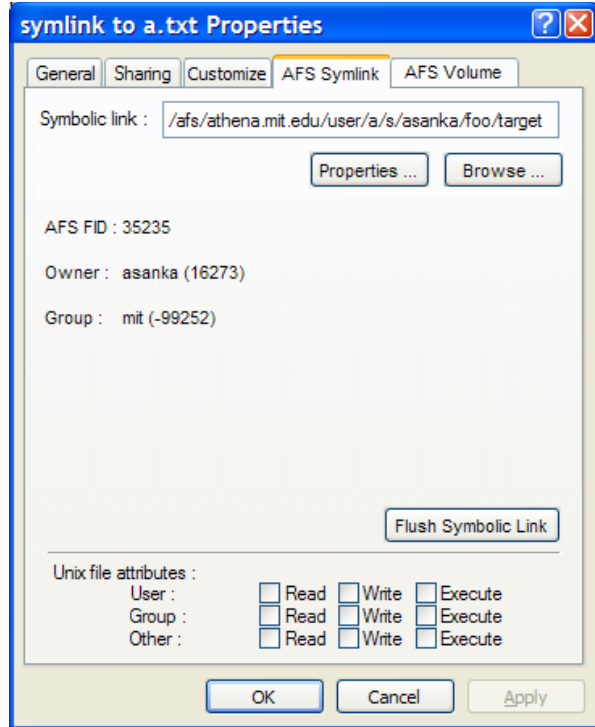
An "AFS Mount Point" sheet

- Mount Point name
- Mount Point destination with Edit button
- Mount Point destination Properties button which displays a properties dialog for root directory of the destination volume.
- Volume Name
- Owner Name and AFS ID
- Group ID
- UNIX file attributes
- Flush Mount Point button



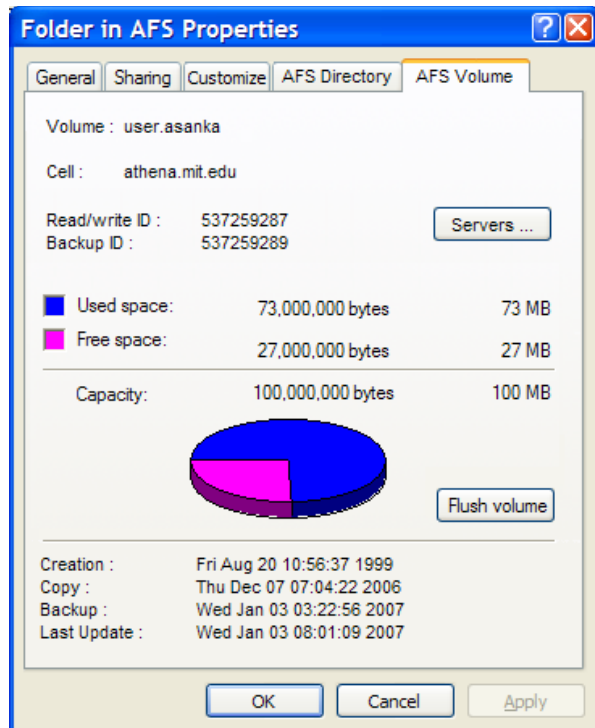
An "AFS Symlink" sheet

- Symlink name
- Symlink destination with Edit button
- Symlink destination Properties button which displays a properties dialog for the item referred to by the link
- AFS FID
- Owner Name and AFS ID
- Group ID
- UNIX file attributes
- Flush Symlink button



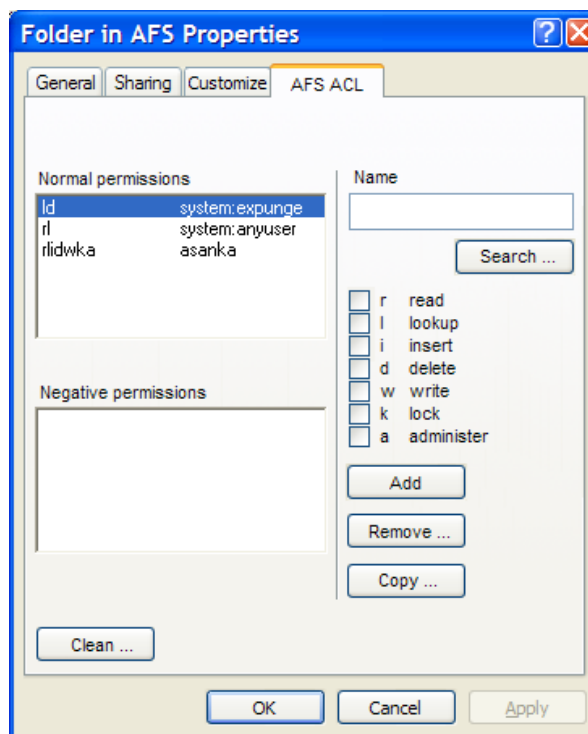
An "AFS Volume" sheet

- Volume name
- Cell Name
- Owner name and AFS ID
- Quota
 - Space allocated
 - Space used
 - Space free
 - Percentage used (perhaps a graph)
- Partition
 - Space allocated
 - Space used
 - Space available
- Flush Volume button



An "AFS ACL" sheet

- Displays current positive and negative ACEs
- Add new positive or negative ACEs
 - Select permissions as checkboxes
 - Search functionality assist with selection of usernames or groups
- Remove existing positive or negative ACEs
- Clone ACLs to subdirectories within same volume
- AFS Group Editor button displays the AFS Groups Panel from the Control Panel



For any given item the "AFS ACL", "AFS Volume", and "AFS Directory" sheets will be added to the Properties dialog. The "AFS File", "AFS Symlink", and "AFS Mount Point" sheets will be added when an item of that type is selected.

Estimated implementation time: 60 to 80 hours

6.2.4 AFS Tool Band

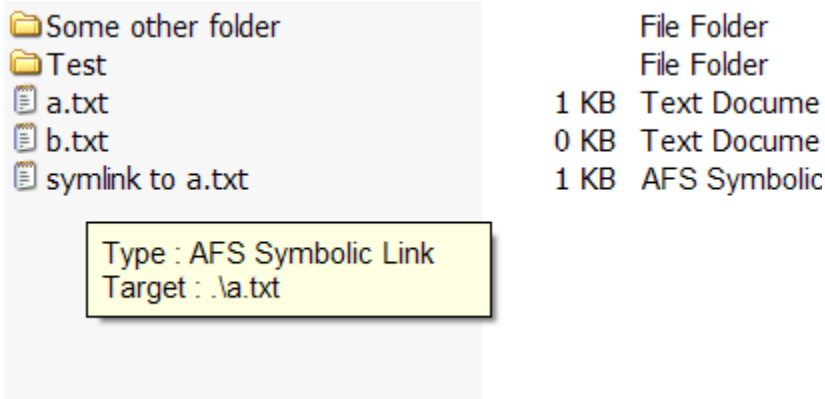
An AFS Tool Band will provide graphical shortcuts to:

- The AFS Groups editor
- The AFS Volume Property sheet
- The AFS ACL Property sheet
- The AFS Directory Property sheet
- The AFS File Property sheet (if the selected item is a file)
- The AFS Symlink Property sheet (if the selected item is a symlink)
- The AFS Mount Point Property sheet (if the selected item is a mount point)

Estimated implementation time: 12 to 16 hours

6.2.5 AFS Tool Tips Handler

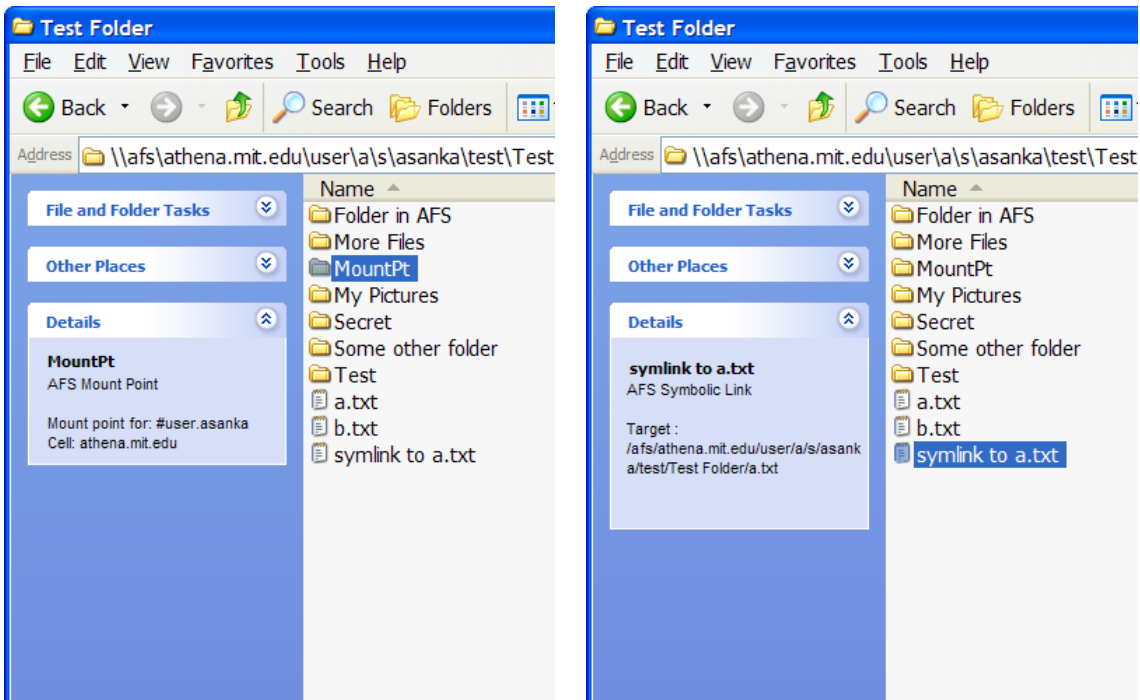
A tool tips handler provides support for additional information about an object when the user hovers the mouse over the object. Tool tips could be used to display the destinations for symlinks and mount points.



Estimated implementation time: 12 to 16 hours

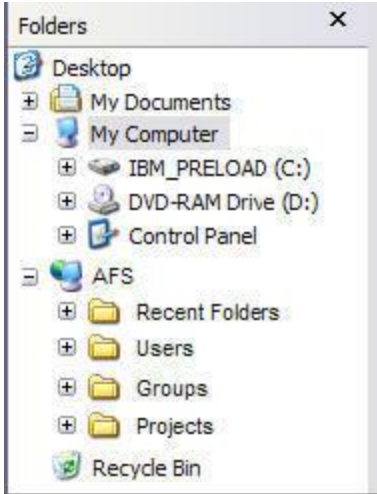
6.2.6 AFS MetaData Handler

A metadata handler can be used to provide enhanced information about selected objects with the "Details" box within the Folders view.



Estimated implementation time: 18 to 24 hours

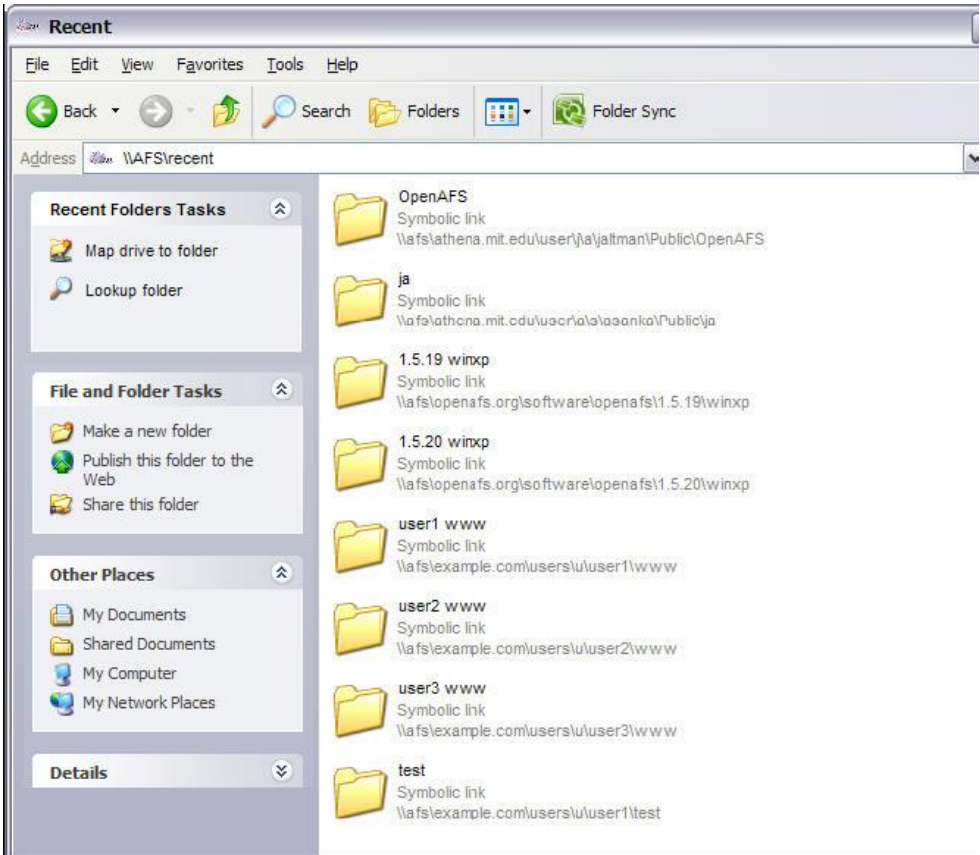
6.2.7 Name Spaces



The Explorer Shell namespace can be extended by providing a plug-in to Windows Explorer called a Shell Namespace Extension. These extensions create a virtual namespace within Windows Explorer and have full control over how the namespace is rendered to the user. A new namespace can be attached to the current Shell namespace as a child of an existing namespace, including file system folders allowing the user to navigate to the new namespace.

6.2.7.1 Recently Viewed Volumes

The numbers of volumes in a typical cell for a large organization are frequently counted in the tens of thousands. The **Recent** name space will show users the most recently accessed volumes from which file data was either read or written. Users will be able to pin volumes within the name space and can easily map drive letters to any listed volume. Volumes are represented as shortcuts. Opening an entry in this name space will redirect the user to a permanent path that can be accessed by applications.

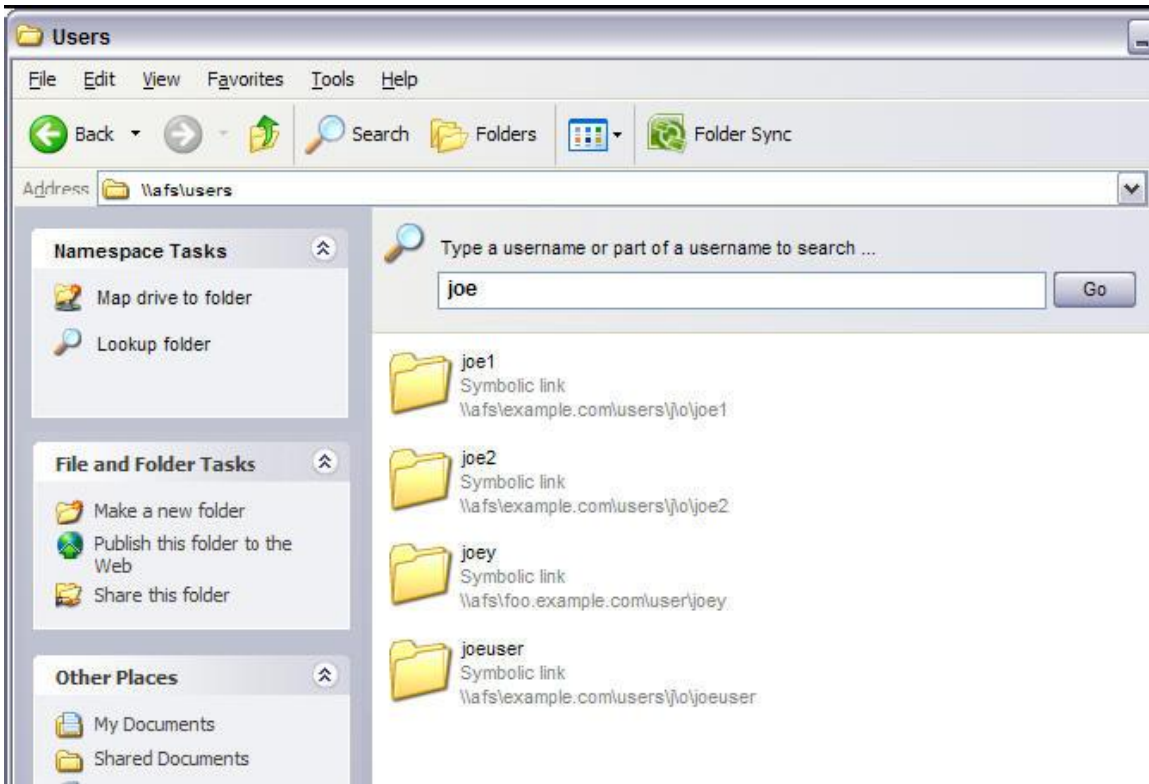


6.2.7.2 My User Volume

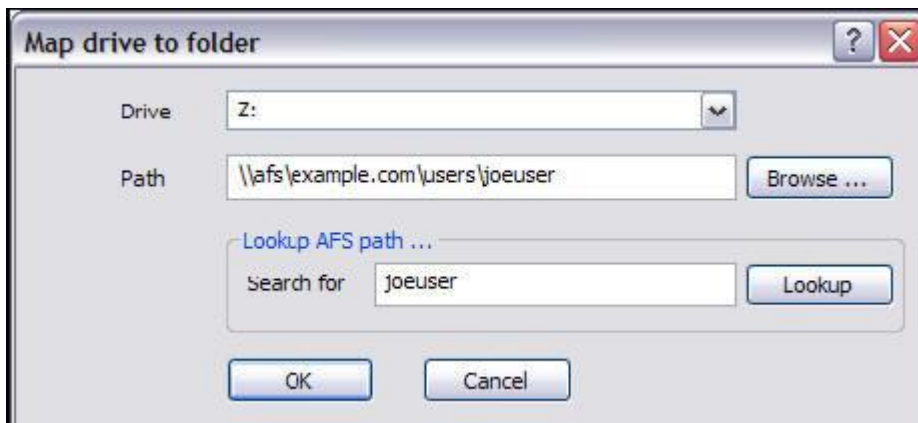
A name space extension can be developed for individual cells which will dynamically select the user's personal volume based upon the Kerberos principal name stored in the AFS token used to access the cell.

6.2.7.3 Custom Name Spaces for Organizations

Each organization develops their own layout for volumes in their cell which are frequently based around users, projects, groups, departments, classes, etc. When there are thousands of volumes within each category it can often be challenging for end users to find the particular data they are searching for. The AFS Name Space extension permits custom name spaces to be defined which can be used to assist end users in finding the volumes that most interest them.



Once a volume is selected, drive letters can be conveniently mapped.



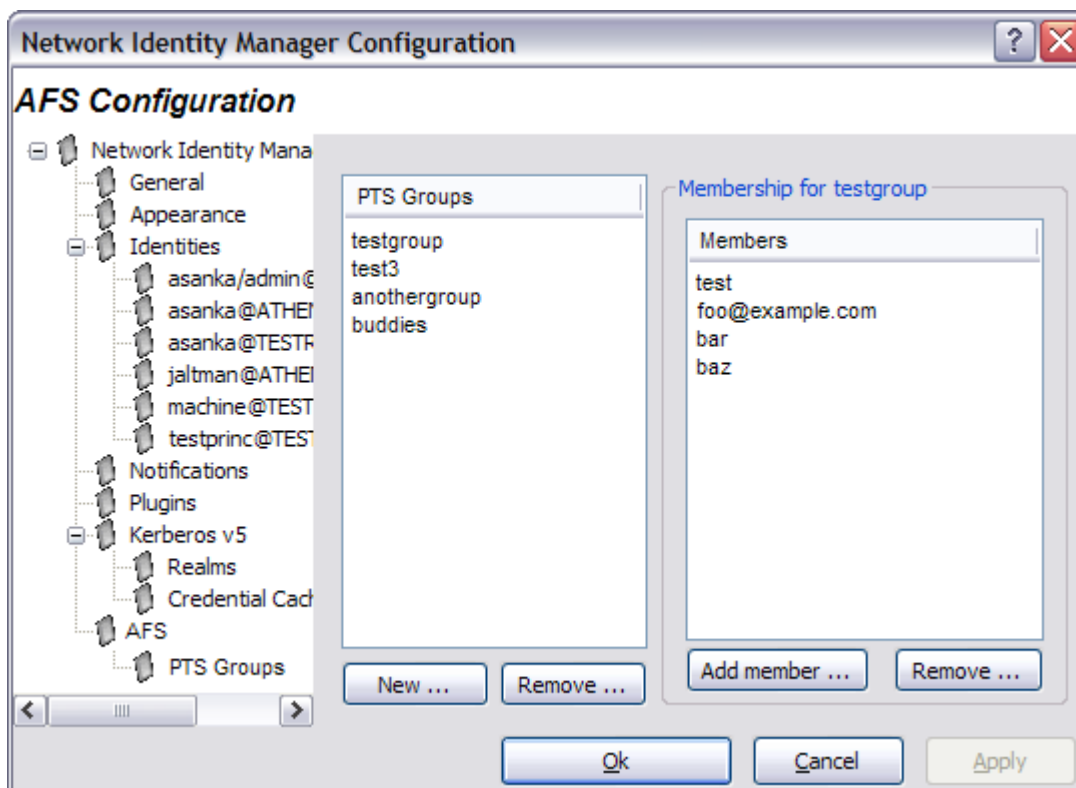
6.3 OpenAFS Control Panel Replacement

The existing control panel provides a mixture of functionality related to the logged in user as well as the system-wide configuration of the AFS Client Service. Configuration of the AFS Client Service should require administrative access which should not be available to all users. To improve the user experience, the control panel should only contain functionality that is applicable to the user. All of the administrator privileged functionality should be moved to the Microsoft Management Console. Microsoft Windows Vista will enforce this functional split by preventing processes from starting with administrative privilege even when the user is an "Administrator" unless the user explicitly grants the permission. Separate processes that run only with Administrative privilege must be created when changes to the machine or service configuration is required.

Therefore, the new OpenAFS Control Panel will only provide the following functionality:

6.3.1 AFS Group Editor Panel

The AFS Group Editor will provide all of the functionality of the **pts** command line tool. Shortcuts to this panel will be accessible from the AFS Explorer Shell Extensions.



Estimated implementation time: 36 to 48 hours

6.3.2 Network Identity Manager AFS Provider Panel

This panel will duplicate some of the configuration capabilities of Network Identity Manager and permit the user to select which Network Identities will be used to obtain tokens for which cells.

Estimated implementation time: To be determined

6.3.3 Microsoft Management Console Shortcut

A shortcut to start the OpenAFS Client Service MMC Plug-in will be provided if the user has administrative privileges.

Estimated implementation time: 2 to 3 hours

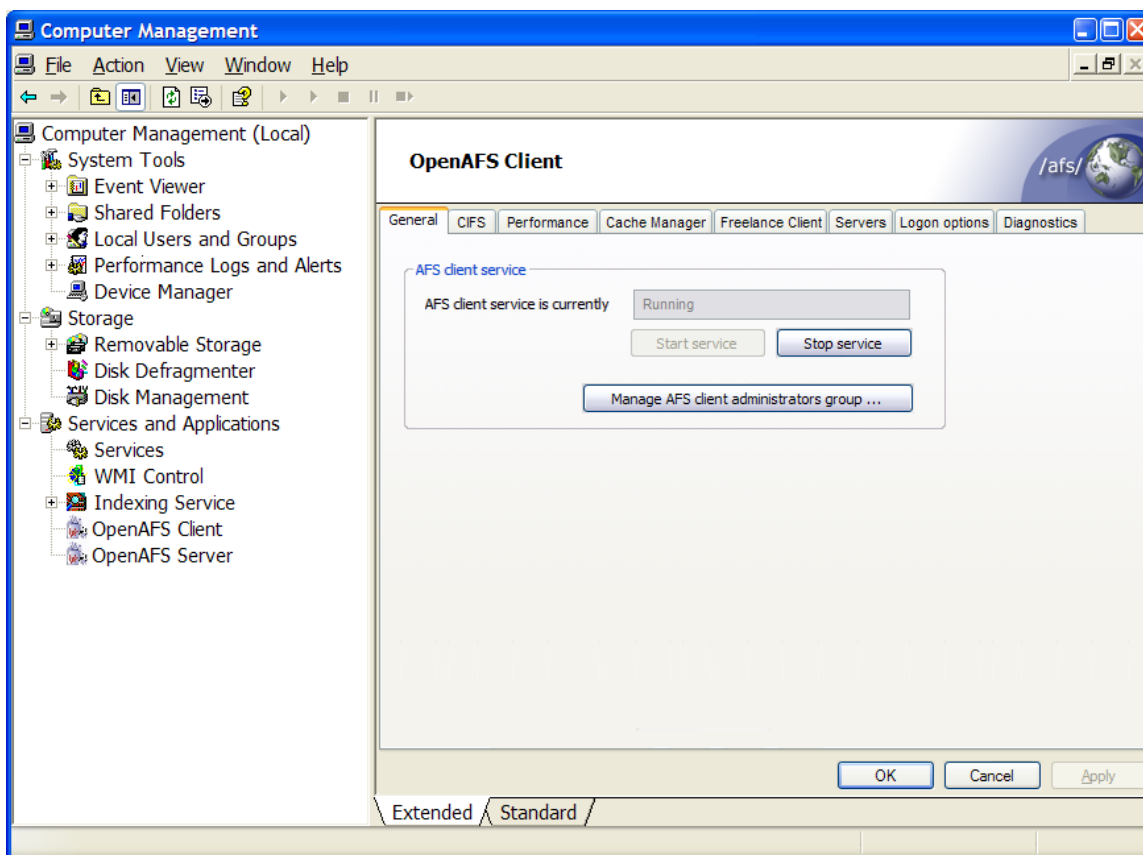
6.4 OpenAFS Client Service Microsoft Management Console Plug-in

The Microsoft Management Console (MMC) has become the standard for configuring policy for Windows Services. The AFS Client Service is highly configurable and yet only a small fraction of the options can be adjusted via the existing AFS Control Panel.

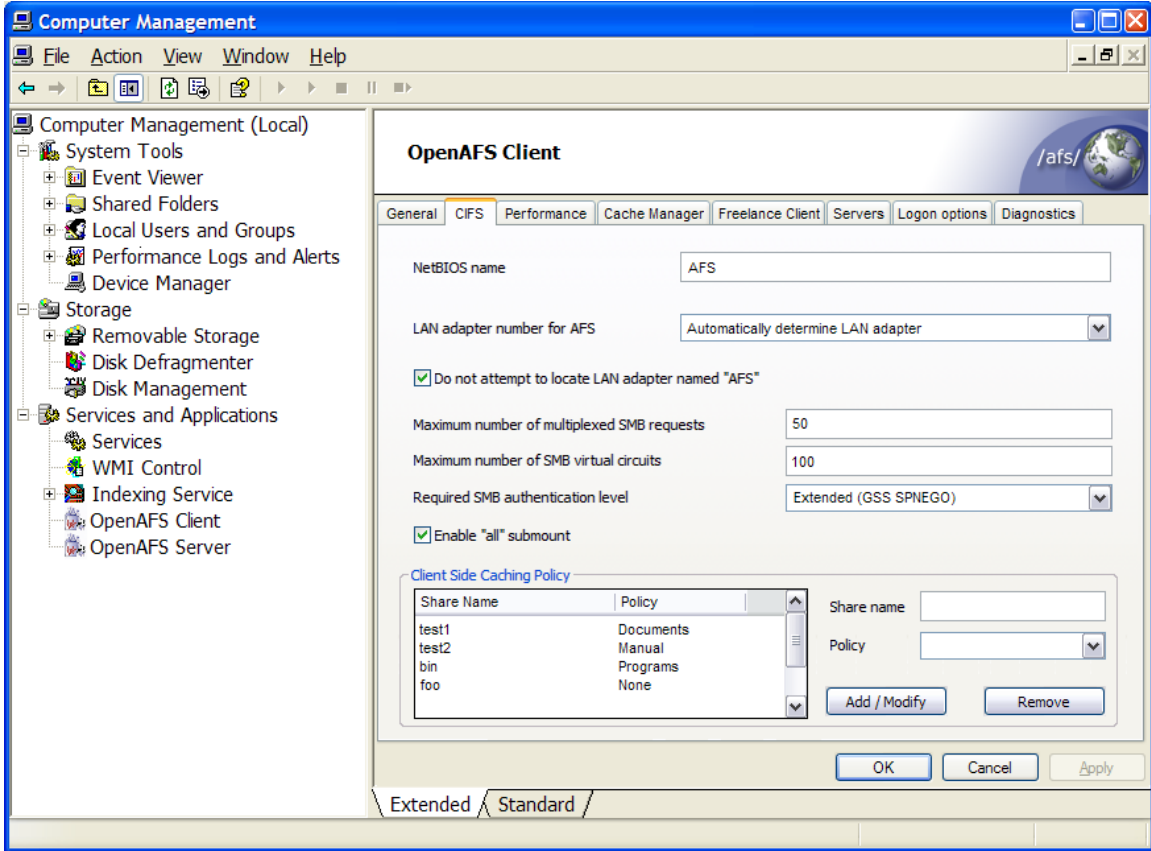
The majority must be set by manually adding or modifying registry values. The list of registry values used by OpenAFS for Windows is documented in [Appendix A of the Release Notes](#).

The MMC will also provide access to:

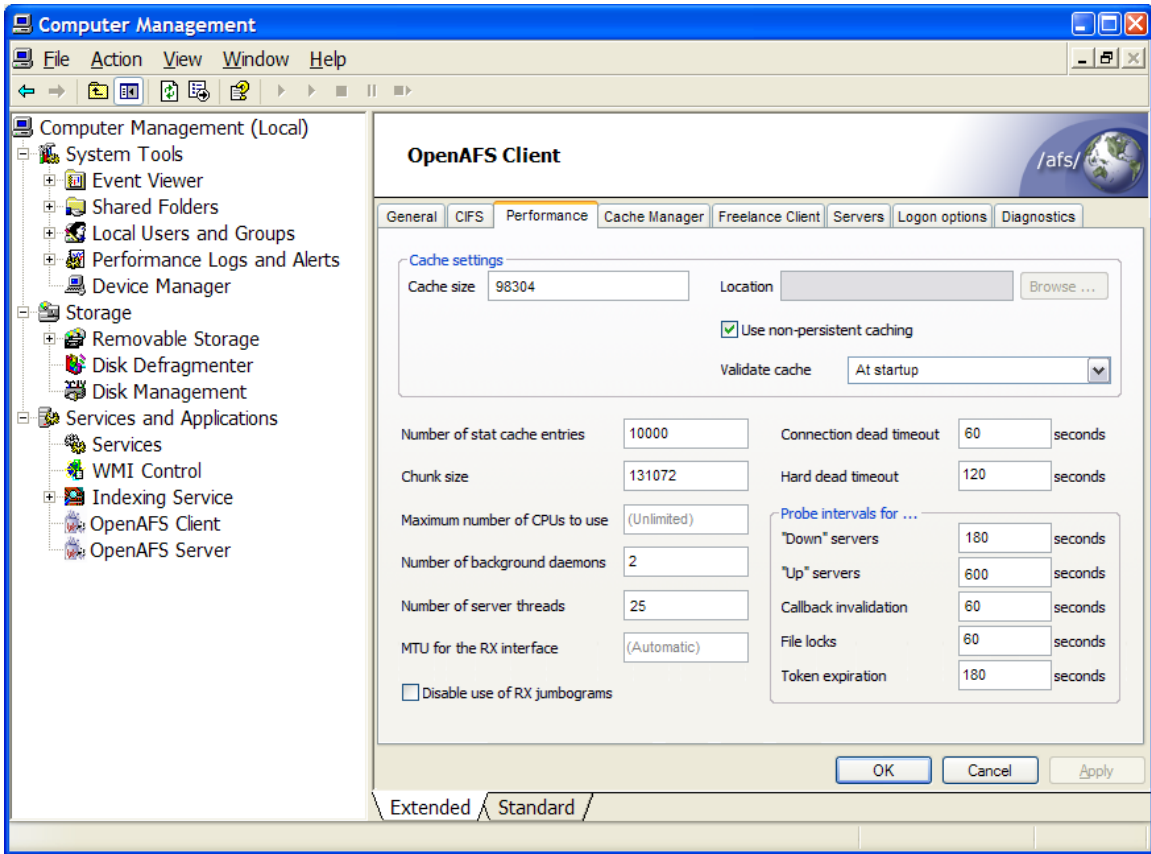
- File Server Preferences (Display, Add, Modify, Make Default)
- Volume Server Preferences (Display, Add, Modify, Make Default)
- CellServDB Editor
- AFS Client Service (Start, Stop, Restart)



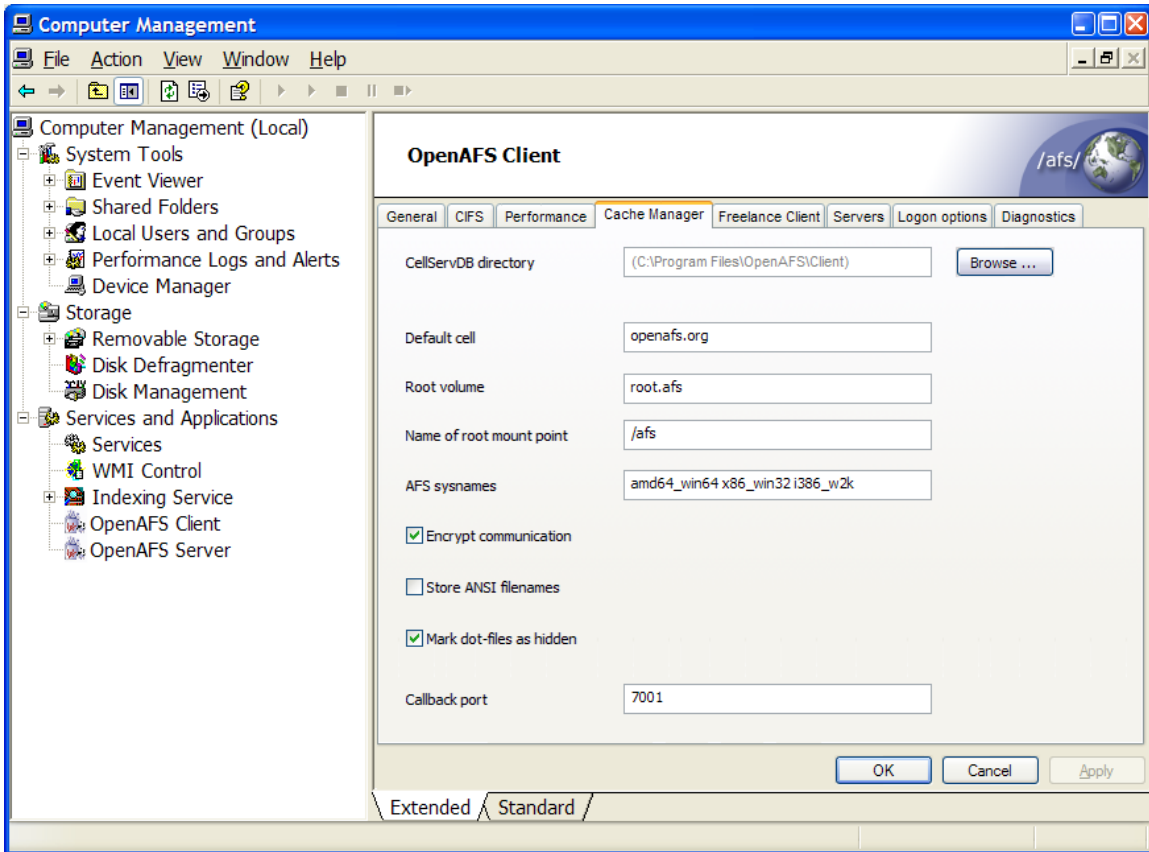
OpenAFS for Windows Status Report: July 2008



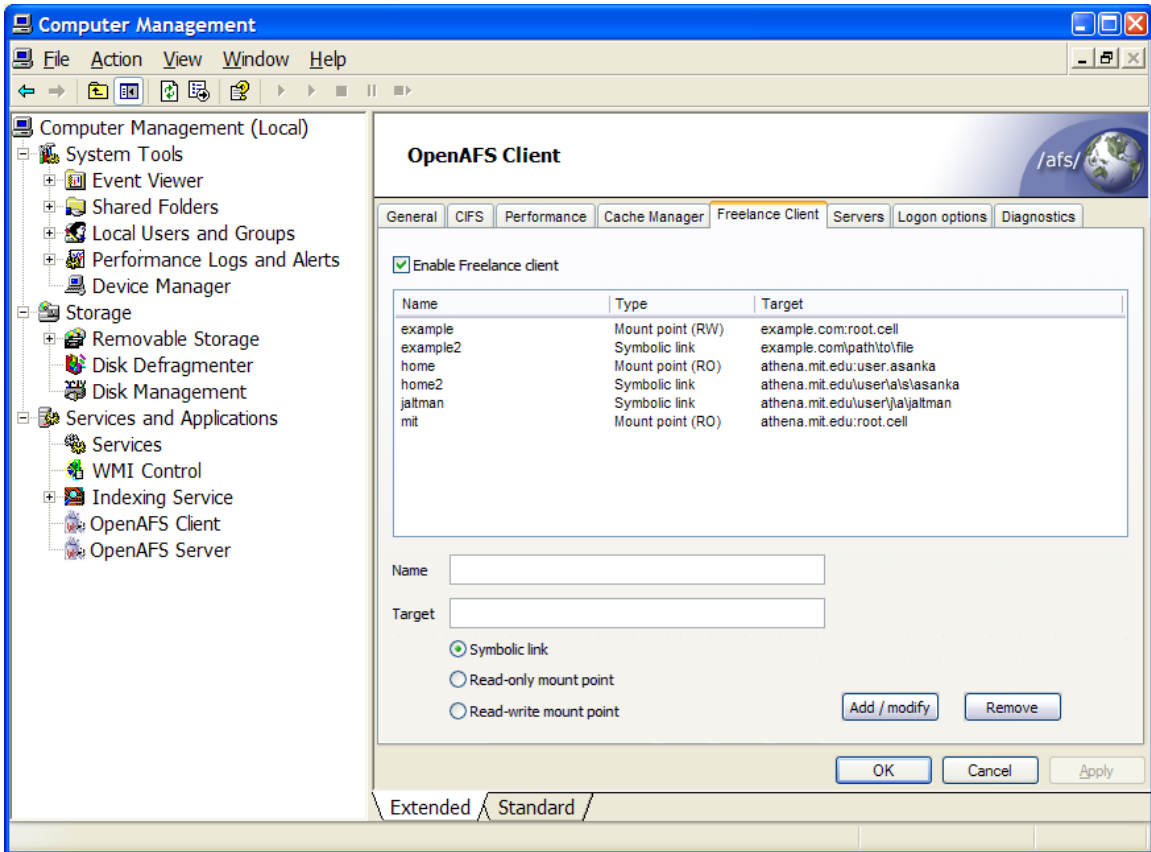
OpenAFS for Windows Status Report: July 2008



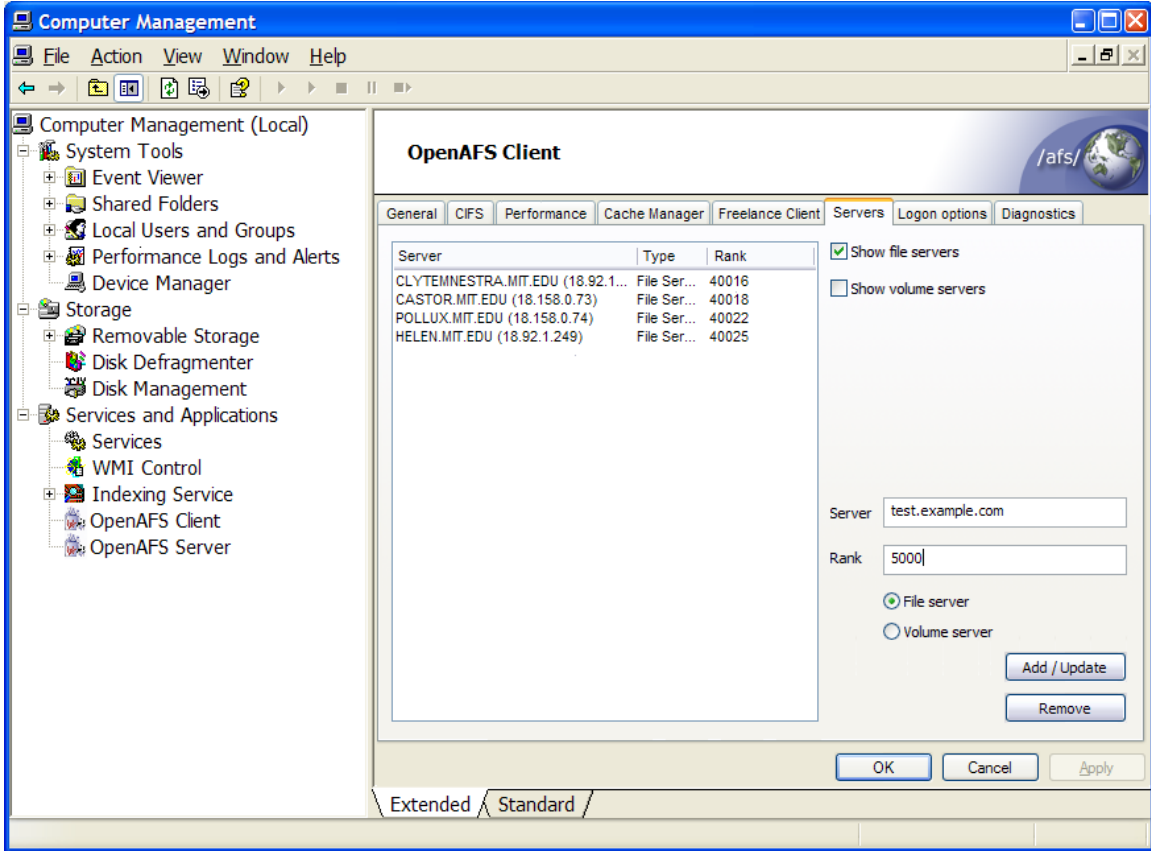
OpenAFS for Windows Status Report: July 2008



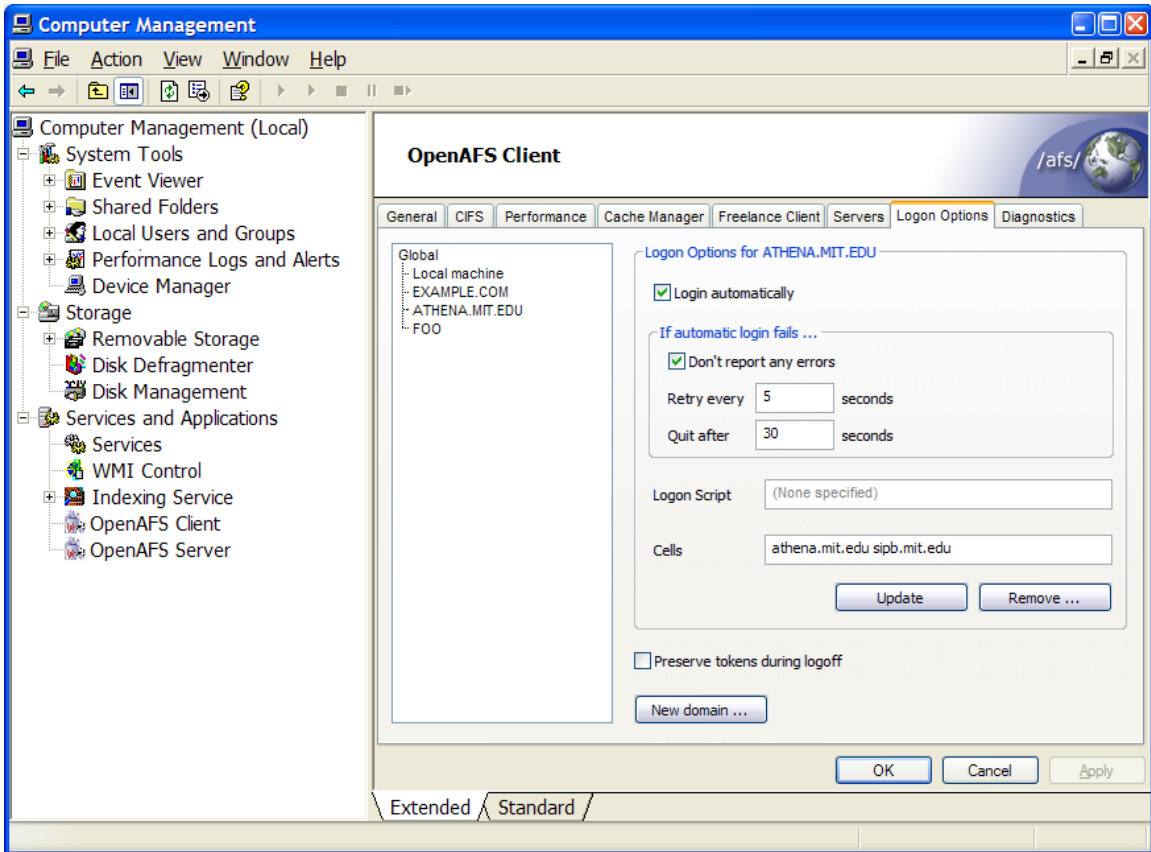
OpenAFS for Windows Status Report: July 2008

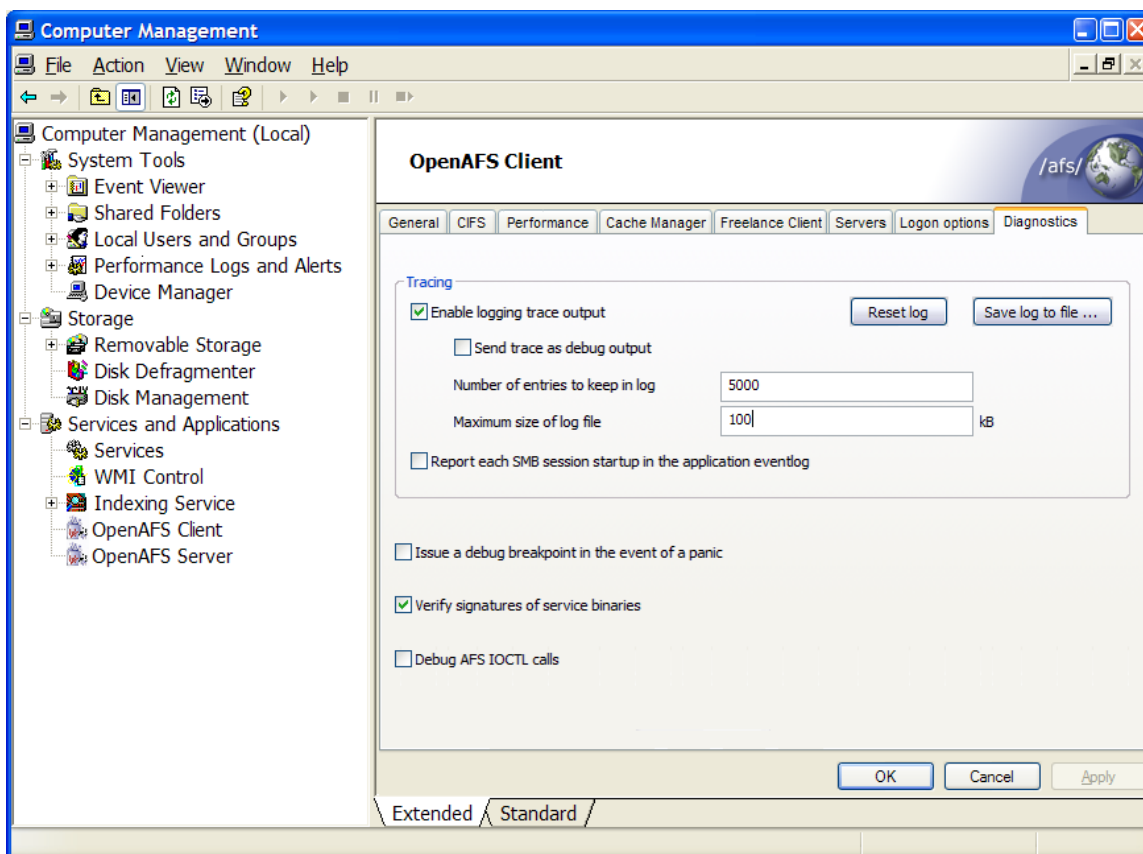


OpenAFS for Windows Status Report: July 2008



OpenAFS for Windows Status Report: July 2008

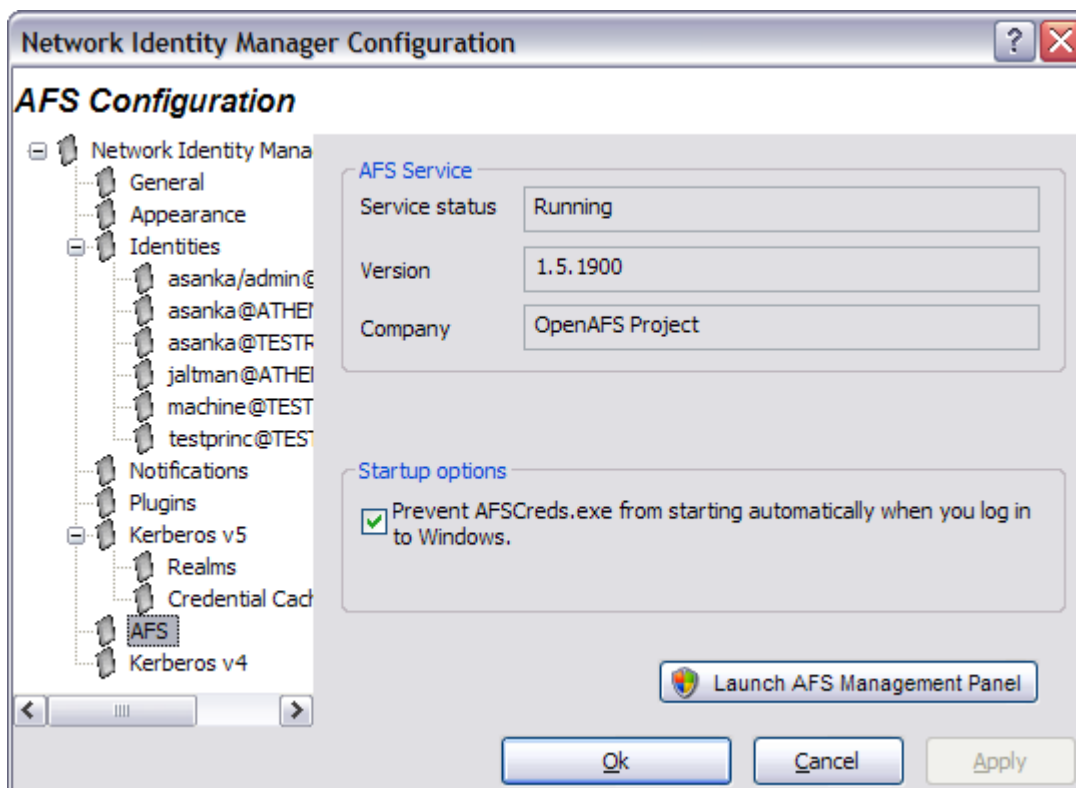




Estimated implementation time: 120 to 160 hours

6.4.1 Vista User Account Control Privilege Separation

Microsoft Vista's User Account Control requires that privileged operations such as starting or stopping services or modifying machine configuration cannot be performed by processes that are executed as normal users. This requires that the Network Identity Manager AFS Provider configuration page be modified to remove the privileged operations and provide a privileged access method to the Microsoft Management Console.



Estimated implementation time: 2 to 3 hours

6.5 AFS Servers on Microsoft Windows

As of the 1.5.51 release, the AFS Servers essentially work on Microsoft Windows provided that they are manually configured. The AFS Server Configuration Wizard can be used provided that nothing goes wrong during the initial configuration. Unfortunately, the wizard is not very robust and cannot be used to resume an install after an error condition has been corrected. Regardless, the wizard makes a number of assumptions about Microsoft Windows and the AFS client that no longer hold true. As a result its use is strongly discouraged.

What follows is a list of things that need to be done before Microsoft Windows can become a trusted platform for hosting AFS servers:

- The NTFS version of the "namei" file server does not include any of the bug fixes and improvements that the UNIX namei implementation has received over the last six years.
- Installation and administration manuals need to be written.
- The kserver (which has been deprecated) must be removed from the installation.
- The installation wizard must be removed from the installation and possibly replaced by a more functional tool.
- A Protection Server implementation that uses Active Directory as the data store should be implemented.

OpenAFS for Windows Status Report: July 2008

- A Protection Server implementation that makes use of native Kerberos 5 or GSSAPI name types should be implemented.
- The BOS Server should log events to the Windows Event Log
- AFS Server configuration information should be moved to the registry to permit its control via Group Policy
- AFS Vice Partitions should be mapped to arbitrary directories and not physical disks

Estimated implementation time: To be determined

Estimated completion dates for these projects are highly dependent upon resource availability. The complete OpenAFS Road Map can be reviewed at <http://www.openafs.org/roadmap.html>. The road map includes descriptions of each work item and implementation time estimates whenever possible.

7. OpenAFS for Windows Needs Your Support

The future of AFS is dependent on a successful future for the Windows client. Without a first class client that seamlessly integrates with Windows, the pressure to migrate to a Microsoft based technologies will be overwhelming to many IT support organizations. In order for OpenAFS.org to meet the needs of the community the projects listed in the road map must be completed. As an open source project, it is the contributions of the community that determine what will or will not be achieved. As the efforts of the last year have demonstrated, great things can happen when the community comes together to provide the necessary resources.

7.1. *Financial Contributions*

Although financial contributions are not the only way to support the development of OpenAFS for Windows, they are the most useful. Financial contributions can be made in a variety of ways.

7.1.1. **Secure Endpoints Inc.**

Secure Endpoints Inc. provides development and support services for OpenAFS for Windows and MIT Kerberos for Windows. Secure Endpoints Inc. is owned by the Windows Gatekeeper for OpenAFS. Donations provided to Secure Endpoints Inc. for the development of OpenAFS are used to cover the OpenAFS gatekeeper responsibilities and providing support to the OpenAFS community via the OpenAFS mailing lists.

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 products.

7.1.2. **The USENIX OpenAFS Fund**

USENIX, 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.

Donations can be made by sending a check, drawn on a U.S. bank, made out to the USENIX OpenAFS Fund to:

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

or by making [a donation online](#).

7.2. Direct contributions of code and/or documentation

Organizations that use OpenAFS in house and have development staffs are encouraged to contribute any code modifications they make to OpenAFS.org via openafs-bugs@openafs.org. Contributions of documentation are highly desired.

Interested parties should contact the OpenAFS Gatekeepers at openafs-gatekeepers@openafs.org. Architectural designs should be discussed on the OpenAFS for Windows Development mailing list: openafs-win32-devel@openafs.org.