eCIFS In Depth

Software solutions for applications and appliances

www.codefx.com
ph: (619) 269-4274
fax: (619) 269-4274
4545 Campus Ave. Suite #9
San Diego, CA 92116

Table of Contents

About CIFS[1]

CIFS, or the **C**ommon **I**nternet **F**ile **S**ystem, is a network protocol that allows computers to share files and printers. There are two distinct entities involved when a file or printer is shared: a client and a server. For sharing to occur, a user specifies a file folder or printer to be shared on the server. Sometime in the future, the client can then establish a connection and logon to the server by sending CIFS packets that indicate which resource the client wants access to, as well as the client's username/password combination. The client can then work with files or use the printer on the remote computer. All of these negotiations are accomplished using the CIFS protocol.

CIFS is an extremely popular network protocol because almost all Microsoft operating systems use CIFS for file and printer sharing. Microsoft includes CIFS client and server capabilities in Windows For Workgroups, Windows 95, Windows 98, Windows ME, Windows NT 3.5, Windows NT 4.0, and Windows 2000. In addition, Macintosh and Unix based operating systems can become CIFS capable using free or inexpensive third party software applications. For these reasons, CIFS is used extensively in home and enterprise environments for file and printer sharing.

New CIFS markets

Several developments and trends over the past 5 years have created new potential markets for the CIFS protocol. These developments and trends are listed and analyzed below.

1. Smart, connected consumer electronic devices are becoming a booming industry. Many of today's hottest gadgets such as PDA's, MP3 players, and cell phones are finding ways to connect to the PC and/or the Internet. Although not possible just a few years ago, the economic feasibility of such advanced products has been proven – connected devices are here to stay.
2. Many homes now contain a local area network (LAN) for file and printer sharing. Network access points in the home have increased dramatically; many construction companies now build new homes wired with ethernet, and the wireless ethernet standard is promising to offer many non-wired homes instant LAN access for affordable prices. In addition, standard network hardware prices for switches, hubs, and ethernet cards have dropped dramatically.
3. The CIFS protocol, as mentioned above, is ubiquitous. As a result of Microsoft's dominance on the desktop, the vast majority of PC's *already* speak the CIFS file sharing protocol.

[1] Much more detailed information on the CIFS protocol in general can be found in the "CIFS Explained" whitepaper available at www.codefx.com/products.htm.

Because of these trends, a new breed of CIFS-enabled electronic devices could offer exciting new services to the consumer. A CIFS-enabled device can gain access to any files or printers that are shared on the Internet or LAN. In order to better understand some of the possibilities with CIFS-enabled consumer electronics, a few hypothetical products are listed below.

Media terminal: A home stereo component could be manufactured which plays modern media file types, such as MP3 music and MPEG encoded video. The device could use local, on-board storage to hold these new multimedia formats, however, it would be more cost effective and elegant to leave the files on a local area network PC and simply use the CIFS protocol to gain access to the media. Utilizing IP connectivity and the CIFS protocol, the device can do away with storage, download, and transfer of the actual media files. The result is less engineering costs, easy storage upgrades (by adding more hard drive capacity to the PC), and centralized storage of all multimedia content. The new device simply acts as a thin client accessing and playing any multimedia content the user requests.

Digital video recorder / Set-top box: Similar to the media terminal above, DVR's and set-top boxes utilizing the CIFS protocol can gain access to a wealth of media files stored on computers attached to the network. Most of these devices already have ethernet connectivity, MPEG playback routines, and on-screen displays for the television. By adding the CIFS protocol for media access, all of these features could be used in conjunction to categorize and play the MP3 and MPEG media files stored on the network.

PDA feature: As PDA's are more and more commonly being used to access files on a PC, a CIFS client could be added to the PDA to allow the exploration of the PC file system directly from the PDA. This would allow the end user much more flexibility and greater ease of use.

In addition to these examples, there are many other potential uses for CIFS running on an embedded device. Cell phones could use a CIFS client to wirelessly access files located on a desktop PC or to download Outlook Express address book information in order to reprogram the cell phone's address database. Fax machines could use CIFS to allow users to specify files on the network to fax. A video game console could completely do away with cartridges or CD-ROM's by instead using CIFS to transfer a game from a PC on the network to the gaming console. ***CIFS opens a new world of creative, exciting and innovative applications for the connected consumer electronic device market***.

About eCIFS

eCIFS is a portable, 'C' source code CIFS client implementation that CodeFX has designed, developed, and debugged. CodeFX sells eCIFS to companies and individuals

who are enabling their smart devices and appliances to communicate on a CIFS network – essentially allowing that device access to any shared files and printers on the LAN.

Entry into the new CIFS markets specified above is best accomplished via the eCIFS product. The following list details the benefits of using eCIFS to penetrate these new markets.

1. **File/printer share access**: eCIFS allows products to speak CIFS immediately, thereby giving the device quick access to shared files and printers.
2. **Less time to market**: The development cycle is cut short by utilizing CodeFX and eCIFS. In-house engineers can concentrate on other issues, while CodeFX handles the CIFS capabilities.
3. **More product sales:** Utilizing innovative and useful CIFS features in a device sets the device apart from the competition. This promotes increased market share and greater sales.
4. **No local storage:** eCIFS allows remote file access capabilities that can eliminate the need for local device storage requirements. No flash memory units and no hard disk drives allows for less engineering requirements and more innovative form factors.
5. **Large user base:** Windows, Macintosh, and Unix based operating systems are already CIFS capable. This means new products utilizing eCIFS can interoperate with all the major operating systems.
6. **No in-house sharing server:** Because the major operating systems are already CIFS capable, additional software to enable sharing on the home computer does not have to be engineered or supported.
7. **Supports open standards:** eCIFS supports the IETF/SNIA CIFS1.0 draft specification. This is a publicly documented protocol and can therefore be understood and extended by anyone.

Because of these reasons, CodeFX feels that eCIFS will be a mainstream tool for developers of smart devices/appliances in the 21st century. A quick, clean, and easy entry into the CIFS connected appliance market is best accomplished via the eCIFS product.

eCIFS technical details

The eCIFS product is written in the 'C' programming language and presents a simple and clean 'C' API to the programmer. eCIFS keeps all of the CIFS packet exchanges behind the scenes and only allows basic file operations similar to "fopen" and "fread" to be called from the API. In using these function calls, a software developer does not have to be very familiar with the CIFS protocol and can concentrate on other programming tasks. A few typical API function calls are presented in Appendix A.

Most eCIFS API function calls can be either blocking or non-blocking. This allows a great deal of flexibility for the programmer. If desired, a given function, such as a remote file read, can block until all the data has been acquired. However, if the calling thread

cannot block, a non-blocking flag can be passed to the read function that allows the calling thread to continue execution, and a programmer specified callback function would execute when the read is completely finished.

In order for eCIFS to accomplish system-specific functionality (such as TCP/IP networking, inter-process communication, and thread spawning), an abstraction layer is used which keeps all of the system-specific utility functions wrapped in generic function calls. In order to port the eCIFS product to a new system, the actual contents of the generic function calls are changed to match the function calls of the new system (this porting process can be done by CodeFX or in-house engineers). These generic function calls are all contained in 4 porting files, and the bulk of the eCIFS source code is never modified in any way.

The CIFS protocol requires the TCP/IP network protocol for delivery of CIFS packets. eCIFS does not include a TCP/IP network stack, and therefore, the desired product must already have TCP/IP functionality. To better understand where the eCIFS protocol fits into a networked environment, the diagram below is included. The network protocols with the green background (NetBIOS and CIFS) are provided by the eCIFS implementation.

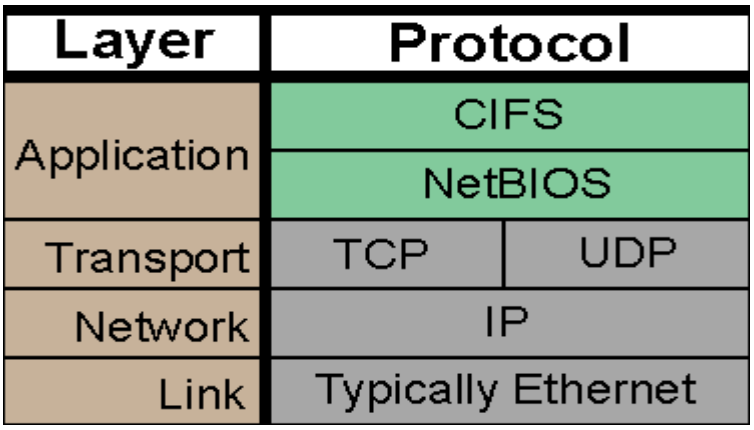

Figure 1- eCIFS implements protocols in green

To promote future compatibility, eCIFS uses the latest CIFS dialect, titled "NT LM 0.12", and is also capable of challenge-response password encryption. The eCIFS code is tested and compatible with Windows 95, Windows 98, Windows ME, Windows NT 4.0, Windows 2000, and the Samba Unix CIFS package. Because Microsoft makes new operating system releases backwards compatible with their previous operating systems, the eCIFS code will interoperate with future Microsoft operating systems (including XP) with no problems.

Conclusion

This concludes the discussion of the eCIFS product. More information on the CIFS protocol in general can be found in the CodeFX whitepaper titled "CIFS Explained". This document is available at www.codefx.com/products.htm. Questions or comments are welcomed and should be emailed to details@codefx.com.

Appendix A

A small subset of the eCIFS API is listed below. Please note that the call_mode argument can be either EC__BLOCK (for blocking calls), or EC__NOBLOCK (for non blocking calls).

int **Ec__ca_fopen**(int call_mode, Ec__ca_conn *conn, Ec__ca_fdata *f_data, char *name, int access_mode, int opts, int share_mode, int magic_num);

int **Ec__ca_fread**(int call_mode, Ec__ca_conn *conn, Ec__ca_fdata *f_data, void *buf, u_int32_t offset, u_int32_t *size, int magic_num);

int **Ec__ca_fwrite**(int call_mode, Ec__ca_conn *conn, Ec__ca_fdata *f_data, void *buf, u_int32_t *size, int magic_num);