Inventor:

Peter Dickinson

CROSS-REFERENCE TO RELATED APPLICATION

This Application is related to the following Application, which ~~is filed of even date herewith,~~ was filed May 3, 1999, and is incorporated herein by reference:

"Methods and Apparatuses for Single Connection File Synchronization and Workgroup File Update", ~~Attorney Docket No. SYMA-1041 MCF/SES~~ Serial Application Serial Number 09/304,295. ~~still pending~~.

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

The present invention relates to the field of personal computers which access files on network drives and which utilize electronic mail systems. Specifically, the present invention involves the synchronization of the local copies of files on user's client computer hard disk to the current versions of the files on a network drive.

2. DISCUSSION OF THE RELATED ART

As more and more business information moves from analog to digital formats, the relatively newfound ability to create, amend, and revise information spontaneously and frequently has brought with it challenges for corporate users. Revenue results can now be easily aggregated and updated in near real time, sales presentations can be amended regularly, and changes made to business documents. However, causing these changes to effectively trickle down through the organization without causing a digital flood is a challenge.

Companies have responded to the threat by carefully creating structures for organizing, storing, and sharing these electronic files. Organizations have moved from file servers to intranet sites to combinations of both to meet the need of the corporate user. While these structures are clearly effective means of storing, sharing, and organizing information, they do not address the

face who do not have the time to spend looking for changes. An effective paradigm for addressing the problem of individual needs and delivering changes to documents can be found in the emerging category of "Push Technology."

As its simplest definition, push is the process of automatically delivering user-requested information electronically. It is not an application, but merely a function or feature in a product. There are clear distinctions between the three different categories of push-based application: content, software, and document.

Content push is the first mover. Conventional products focus on delivering breaking news and information to user desktops automatically. Instead of the user constantly surfing multiple Web sites for stock quotes, news, weather, etc., conventional products aggregate and broadcast information automatically according to individual user preferences. Many companies incorporate "push" functionality into their products.

Following acceptance by hundreds of thousands of early adopters, many push-based applications started the move into the corporate world. For IS Managers, "push-based technologies" were seen as an uncontrollable avenue for terabytes of graphics and HTML to come through the corporate firewall and network, filling local hard drives.

Microsoft and Netscape entered the fray with their own "push" clients — IE 4.0 Active Desktop and Communicator's Netcaster, respectively. Rather than spurring the growth of content delivery, however, the effect of the push entries has been to call into question the value of delivering Web content to user hard drives. The value is questioned not only in terms of relevance, but also its effects and load on corporate networks. The automated information flow becomes a flood through the Internet gate-ways of corporations threatening the stability and reliability of the network infrastructure itself.

Within the corporate world, the future of content push remains in limbo. Uncertainty over standards and overall value have caused the market to trip on the initial momentum and slow to a crawl. However, what is questioned here is not the value of automating delivery or "push," but rather the value of what is being pushed.

Software Push is another important objective. Microsoft and Marimba, among others, have recognized the importance and potential of "Electronic Software Distribution" (ESL) or "software push" as a way of addressing the need to seamlessly deliver software updates across

"content" in software delivery is, by its very nature, deemed critical. To reduce the impact of supporting multiple versions of products across the corporate network, near-simultaneous deployment is imperative. Companies that do not use some form of software push technologies require dedicated individuals to make the rounds updating software by reinstalling or applying patches for each personal computer and laptop.

Rather than aggregating and displaying information, software push transparently delivers one specific piece of information and applies it to user systems or applications. Files tend to be very large and the delivery of these files must be well-managed. Incremental downloading becomes important to reduce frustration and bandwidth associated with broken and lost connections. Management of software updating also needs to be centralized and MIS-controlled. In addition, the primary value of the application is to IS personnel and only indirectly to for the end-users.

A good example of software push is Marimba's Castanet, which allows Java applications to be distributed and updated seamlessly and automatically without user intervention. This same approach to Java programming can be, and is being, applied to C-language programs as well. The case of content push vs. software push makes it clear that the importance lies in the distinction between the data being delivered — and not the delivery mechanism itself.

The Next Phase is Electronic Document Delivery. The final frontier in digital push is "electronic document delivery"or EDD. It deals with delivering changes or "updates" to the same physical files (like software push), but the files themselves are highly personalized (like content push). Different from content push, these files exist in the form of sales presentations (PowerPoint), spreadsheets (Excel and Lotus 1,2,3), and reports and plans (Word or WordPerfect). These are the types of documents for which companies currently invest millions of dollars in file servers and intranet technologies in order to share among respective workgroups. The important distinction here between content and document push is the fact that EDD delivers data that currently exists in its native format within corporations and whose value is clearly understood by the company, MIS, and the end-user. With the recognized features, the willingness to invest in infrastructure is more likely.

Within conventional environments, users have access to files and can download or copy

is that these files change erratically and the user can never really know when a file has changed. As a result, those who need to have the most current documents are required to perform hit-and-miss network browsing and checking which is time-consuming, unproductive, and frustrating. Even if the changes are scheduled, the user is still required to manually access, retrieve and manage those changes.

For mobile users, the problem of knowing about and accessing changes to network-based files is compounded by infrequent access to the corporate network. In addition, when remote from the office, users need to establish connections to the network via dial-up networking technologies, then search and browse the network over an often slow, unreliable connection. The productivity losses and frustrations are simply multiplied.

As is apparent from the above discussion, a need exists for an efficient and effective mechanism for allowing a computer user to have copies of the current versions of network files on his client computer.

Conventionally, those who need to have the most current versions of computer files are required to perform hit-and-miss network browsing and checking which is time-consuming, unproductive, and frustrating. An object of the present invention is to provide a mechanism by which a user can be automatically provided with a current version of a file to which he subscribes. Another object of the present invention is to communicate the current version of the file in an efficient manner. According to the present invention, a server computer monitors network files and folders stored on the network for changes and then sends the user email notifications and updates when monitored items change.

According to the present invention, a server computer generates an update file for transmission to a client computer that permits the client computer to generate a copy of a current version of a subscription file from a copy of an earlier version of the subscription file. The server computer periodically reads the subscription file from the network drive and divides the subscription file into variable-length segments based upon a segment delimiter. The server computer computes a signature for each segment and stores the segment signature along with the beginning position and length of each segment in a current version of the signature list. The server computer also maintains the earlier version of the signature list.

For each segment of the current version of the subscription file, the server computer searches an earlier version of a signature list for an old segment signature which matches a new segment signature corresponding to the segment. When a match is detected, the server computer writes a command in the update file for the client computer to copy an old segment of the client computer's copy of the earlier version of the subscription file into the client computer's copy of the current version of the subscription file, where the old segment corresponds to the segment for which a match was detected. The command need only specify the location within the earlier version of the file where the old segment is stored, rather than the actual data that is stored at this position. This information is found in the signature list in the beginning location and size fields. The beginning location field is preferably expressed as a number of bytes from the beginning of the file. At the client computer, when this location information is combined with the offset of the beginning of the client computer's copy of the earlier version of the subscription file, the correct old segment can be copied into the client computer's copy of the current version of the subscription file. The size of the copy command is negligible in comparison to the size of the

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.