



STEPHEN SHEFFIELD

exchange documents by e-mail rather than putting them on a central server. After all, sometimes we are connected to one of these data repositories on a fast network, sometimes we are connected on a slow network, and sometimes we aren't connected at all. People like having their own copies, and then keeping them up to the minute by invoking an incredibly powerful concept called "sync."

Sync—short for synchronization—is all about being able to take data from one location and intelligently copy it to wherever it might be needed. And most importantly, sync is about tracking changes. A good sync system allows you or others to freely edit either the original document or the copies, and then have your changes automatically propagate wherever they are needed. The best of these systems track changes as they are made and allow for sophisticated "undo"—for

## SUPER SYNC

For decades, future-gazing technologists and visionaries have assumed that technology would bring into being some sort of electronic Library of Alexandria. In this scenario, massive databanks would be centralized information utilities, with access granted by cheap, fast and ubiquitous data feeds. In *Star Trek*, for instance, crew members used a wireless network to link their tricorders with the starship *Enterprise's* onboard computer—that storehouse of all things interesting and relevant. And the original Internet was created not to distribute information, but as a massive remote-access system, enabling researchers in one place to tap into computers located somewhere else. Even George Orwell's *1984* envisioned a world in which records of news and current events were stored in relatively few places; that's how Winston Smith was able to edit the past so that Big Brother's pronouncements were never wrong.

But reality is heading in a different direction. Instead of ubiquitous connectivity to centralized databanks, we are instead building an infrastructure that's optimized for data replication. The same information is getting copied to dozens, hundreds or even thousands of places throughout the world, and it is kept current through continual retransmissions and updates. Humans instinctively work this way—that's why people collaborating on a project tend to

example, restoring the data to the way it was on a particular date, or removing all of the changes made by a particular author. Sync is not a new idea, but it has only recently begun to emerge as a crucial part of the way that computer systems are run. Sync technology really will set the tone for future computing.

Today the leading sync platform is the Palm operating system, which I've lauded in the past. In fact, one of the big reasons for the Palm's popularity is its "HotSync" technology. Palm users can add, delete or change their address books, appointment calendars or other databases on either their desktops or their Palm-based computers. Put the Palm into its cradle and press the "HotSync" button, and the changes are automatically replicated on the other machine.

But HotSync doesn't stop there. You can, for example, sync multiple desktop computers to the same Palm, allowing people at both your home and your work to access and update your calendar—even if there is no network connection between them. All you do is carry the Palm back and forth between home and office, syncing at both locations; the intelligent software does the rest. If you forget to sync, it's no problem—the next time, the system automatically adjusts.

Of course, sync technology goes way beyond the Palm—and way beyond one person's desktop PC. A synchronization program called the Concurrent Versions System is at the root of many successful software development projects, from

small corporations where a few programmers are working on the same project to large-scale open-source collaborative efforts enrolling hundreds or thousands of programmers. This technology makes it possible for many people to work on the same program at the same time. Every programmer has a personal copy of the software being developed. A programmer who adds a new feature or fixes a bug can “commit” that change to the project’s repository. Other programmers—either down the hall or across the globe—can then update their copies and automatically have the changes applied. It used to be cumbersome for more than one person to work on a single program at the same time. But now, simultaneous development and bug-fixing are the rule rather than the exception. The Concurrent Versions System, built on the concept of sync, has dramatically sped up software development.

For open-source programmers seeking to compete with commercial software, sync has been a curse as well as a blessing. It is tremendously difficult to design applications that get sync right. The advanced synchronization systems built into commercial database software like Oracle make it possible to build huge database farms by linking together large numbers of synchronized servers. So far, leading open-source databases like PostgreSQL and MySQL provide only limited support for database synchronization. The open-source systems will probably catch up one day soon, but the technology is inherently difficult to develop.

Sometimes the information source being synced is a moving target. Consider Usenet, the original global bulletin board system. When two Usenet servers connect, they essentially synchronize their articles. If an article is on one machine but not the other, a copy is made to eliminate the discrepancy. Back in 1991, John Gilmore, one of the founders of the Electronic Frontier Foundation, said, “Usenet interprets censorship as damage and routes around it.” What he meant was this: any university or business that doesn’t like articles posted on Usenet may delete them from its server, but because the articles flow through the network as a whole, no one institution can block information from getting out to the world.

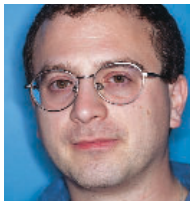
In recent years, Gilmore’s quotation has been reinterpreted by many journalists as referring to the Internet rather than Usenet. Sadly for both Gilmore and the cause of free speech, this alteration makes the quote inaccurate. When articles are published from a Web site, instead of through Usenet, they are indeed distributed from a central location—and that central location can be subject to censorship or other forms of political pressure.

Online file-swapping operations like Napster and Gnutella are really just very large synchronization services. Users have their own visions of what music or other files they want, and they sync and sync until they are happy. Here, “sync” applies

not to an individual file but to a collection. The results, however, are much the same.

Downloading music from a file-sharing service is fundamentally different from downloading information over the Web. In the case of the Web, very few readers keep and redistribute their own long-term copies. This is why Napster and its descendants are threatening to the music industry; as Usenet showed, it can be exceedingly difficult to stop the spread of data through a large-scale synchronization system. Indeed, one of the great advantages of sync is redundancy: even if the “master” copy gets erased, sync invariably leaves many other copies around. This phenomenon makes it hard for outsiders to eradicate or control information that is shared by sync.

Understanding the uses and power of sync is vital for accurately predicting the direction that the Internet and e-commerce are likely to grow. Most people like the safety that comes from having data in multiple locations, and the speed that comes from having the data immediately available on their own computers. Products and services that offer sync, therefore, will probably fare better in the marketplace than similarly priced services that offer high-speed access to data stored on remote systems. People don’t want to just tap into a data stream; they want to have their own copy of the information, and they want it kept up-to-date. This has broad impli-



**Instead of ubiquitous connectivity to central databanks, we are building an infrastructure of data replication: the same information is copied to multiple locations and kept constantly up-to-date.**

cations for everything from video on demand to home banking. I’ve been doing so-called Internet banking with Intuit’s Quicken software for years: every few days, I download my account’s most recent transactions and corrections over the Internet and add them to my register. My bank also lets me view my whole statement on the Internet. Would I give up downloading the transactions by themselves? Not on your life—I feel safer keeping my own copy.

Sync makes economic sense too. With sync, you aren’t so dependent on an expensive, always-on, high-speed Net connection. You can get much of the same effect with local storage and slow or even intermittent network connections. Sync really does mirror the way that the world has been built—as opposed to the way that pundits and engineers thought it would be.

In fact, even the Library of Alexandria was built through sync. Every ship that docked in Alexandria was searched for scrolls: if any were found, the ship was not allowed to leave until the scrolls were copied. Alas, the library’s hundreds of thousands of scrolls were lost when they were burned by Julius Caesar in 47 BCE because they didn’t sync a backup. ■