

The History of Notes and Domino

By Richard Schwartz

Eighteen years after Ray Ozzie struck a deal with Mitch Kapor, the founder of Lotus Development corporation, and formed Iris Associates to start developing a product based on his vision of collaborative software for networked PCs, Lotus Notes and Domino 6 has been released by IBM. Ray's vision, which has survived corporate changes and even his own departure to form a new company, Groove Networks, has grown into one of the most successful software products in history, with an installed base of some 80 million users. Notes and Domino owe much of their current success to the fact that Lotus and IBM (which acquired Lotus in 1995, shortly after Lotus acquired Iris) have kept up with emerging trends in information technology at every step along the way. The credit for this goes back to the original vision for the product.

The Notes Vision

The vision for Lotus Notes was never really about any specific technology. The proof of this is that software development done at Iris was cross-platform right from the very beginning. The code that Ray and the other engineers at Iris wrote was designed to be portable to many different operating systems and network protocols, and versions of Notes and Domino have been released for a half dozen different processors, five network protocols, and more than a dozen operating systems. The vision for Lotus Notes was a vision of the way people work and the tools they need to work together.

Ray Ozzie first began to piece together his vision when he was a student at the University of Illinois. While there, he had access to a system called PLATO, which was a mainframe-based timesharing platform for educational computing. In 1973, a programmer named David Woolley wrote a program that allowed users of the PLATO system to send problem reports to the system managers. The program also allowed the system managers to respond to the reports, and the

CHAPTER

In This Chapter

◆ ◆ ◆ ◆
Understanding the Notes vision

Understanding the evolution of Notes into Domino

◆ ◆ ◆ ◆
Introducing Notes and Domino 6

4 Part I ♦ Introduction to Lotus Notes and Domino

users to respond to the responses. As far as anyone knows, this was the first software implementation of a multi-user, online, threaded discussion system. The system, which was called Notes, was soon adapted for other purposes besides reporting problems.

A module called Personal Notes was added to PLATO in 1974. This was essentially an e-mail system for PLATO users. Nobody really knows who invented e-mail, and Personal Notes was far from the first e-mail system, but even in the late 1970s e-mail was a foreign concept to most people. Ray Ozzie may have been the first person to see both a threaded discussion and e-mail and realize that a system that combined the features of both could be an incredibly powerful tool for businesses.



Note

Nobody really knows who invented e-mail, but it is known that it dates back to the early 1960s. We have heard credible claims for unknown students at both MIT and Dartmouth College. Since most people credit MIT, we're willing to accept that, but the Dartmouth story is actually more interesting because it claims that a student developed the system to communicate with his girlfriend, who could log in to the Dartmouth computers from a terminal at her college more than 90 miles away. Even if this is true, however, the first e-mail system that worked by sending messages between two computers across a network, however, is credited to Ray Tomlinson, an engineer at BBN, the company that did much of the early development work on ARPANET.

Ray also saw a PLATO module known as Group Notes, which extended the original Notes program to allow users to create their own private threaded discussions. One of the key concepts of Group Notes was the Access List, which defined whether individual users could create new threads, read threads, and respond to threads. Ray's vision included similar security mechanisms, and more, because he realized that a system for sharing information among workers could be successful only if it could be trusted to enforce limits on sharing.

Another PLATO feature that influenced Ray Ozzie was networking. By the late 1970s, both Personal Notes and Group Notes had been adapted so that users on different PLATO mainframes in different locations could exchange messages and participate in the same threaded discussions. In the early 1980s, the IBM PC exploded onto the business scene, and Ray correctly predicted that PCs would eventually be networked, allowing software that worked like PLATO Notes to revolutionize the way office workers collaborate with each other.

Ray Ozzie was not the only person to see and be inspired by PLATO Notes. The list of software that traces its lineage back to PLATO Notes includes DEC Notes, which was written for Digital Equipment Corporation by another former U of I student, Len Kawell, who was one of the four co-founders of Iris with Ray. Another descendant of PLATO Notes is the Usenet News system, which provides tens of thousands of newsgroups to millions of users of Internet servers around the world.

Realizing the Vision

Lotus Notes, which was initially released in 1989, began the realization of Ray Ozzie's vision. Ray combined the e-mail, threaded discussion, security, and networking concepts that he had seen in PLATO with five more key concepts to complete the vision. These concepts include:

- ♦ The idea of the container object model, the "note" database
- ♦ The ability to create and manage loosely defined data schemas

- ♦ The concept of rich text, and creating a rich text object as a container for text with effects (for example, bold, italic, color) as well as objects
- ♦ The ability to create fairly robust database programs with user-friendly programming interfaces
- ♦ The concept of portability and remote access — what is commonly referred to as replication

The first of these new concepts was the “note” database.

Ray and the other engineers at Iris devised the Notes Storage Facility, or NSF file. It’s a bit of an oversimplification, but the best way to think of an NSF file is as a container full of different types of notes. Some notes are *data notes*, also known as *documents*, which contain the data that users work with. Some notes are *design notes*, also known as *design elements*, which contain meta-data that controls the formats in which users see and edit the data documents. We describe these types of notes, and others, more thoroughly in Chapter 4.

The second of the new concepts was loosely structured data. Most database architectures require that all data conform to a rigidly defined schema. The database management software validates all data to make sure that it follows all the rules laid out in the schema so that required data is always present, all data elements have the correct type, and all relationships between records are maintained correctly. Notes does not do this. Over the years, many critics picked on the fact that Notes is not relational, but the real root of criticism is that Notes doesn’t enforce a schema, and in Ray’s vision this was a strength of Notes, not a weakness. There are many good points for both sides of this argument, but the philosophy behind Notes is that enforcing a schema makes it harder for real-life Notes applications to include all the data that users really need. Managing a database schema, and managing the data to make sure that it conforms to any changes in the schema, takes time and requires a fairly high level of skill and discipline. Ray and his associates at Iris wanted the management of Notes databases to be simple so that applications could be created quickly and modified easily.

The third new concept that Ray added to the vision was rich text editing. By giving users word processing style features for control over colors, fonts, and text layout, as well as the ability to insert non-text elements, such as graphics or file attachments, Ray elevated Lotus Notes’ functionality far above the simple text processing offered in PLATO and the other collaboration systems that preceded Notes.

The fourth new concept was programmability. By creating what you see is what you get (WYSIWYG) design tools for generating forms for displaying and editing documents, views for displaying lists of documents, and a simple programming language for doing computations, Ray made Lotus Notes much more than a set of single-purpose applications. It became a platform for developing custom, client-server applications.

The fifth new concept was remote access. Ray extended the networking concept in PLATO to include mechanisms for dialing in to connect to a Notes server, and for replicating databases between servers and clients. This allowed users to take Notes with them when they traveled with their laptop computers, and to connect with a modem in order to resynchronize the data. Local area networks (LANs) were just beginning to become commonplace during Notes’ early years, and dial-in network access (Microsoft’s RAS) was not in widespread use. Inexpensive high-speed network connections didn’t exist either. Notes users, however, could take their work with them wherever they went, and this was an extremely important advantage for Lotus.

The Evolution of Notes into Domino

Groupware is a term that evolved in the mid 1980s to refer to software that helps users in workgroups collaborate more effectively. When Lotus Notes hit the market for the first time, adoption was a little slow. But by the time Release 3 came out in 1993, Notes was clearly the dominant groupware platform. The early releases of Lotus Notes were clearly targeted at workgroups, but with Release 4 in January 1996, it became obvious that Lotus was positioning Notes as an enterprise solution and that this positioning was going to be successful. Groupware became something of a passé term, and the Notes became the leader in the new, less limiting buzzword: collaboration.

At about the same time that Notes was experiencing rapid growth as an enterprise collaboration platform, the World Wide Web was taking the world by storm. There was a great deal of hype about how the Web was going to be a Notes killer. Lotus recognized that Web technology had certain advantages over any system that relied on a proprietary fat client, but that it also had many weaknesses. The standardized technologies used on the Web addressed only a small part of what Notes did. Still, it was clear that tremendous momentum was building up behind Web technologies, and it was only a matter of time before someone overcame the limitations and developed solutions that delivered a rich set of collaboration features via the Web. Lotus decided to be that someone.

In December 1996, less than a year after releasing Notes R4, Lotus changed the name of the Notes server, and released versions 4.5 of the Lotus Notes client and the Lotus Domino server. This release included an HTTP server that could make Notes applications accessible to users of Web browsers. The HTTP portion had actually been released earlier in the year as an add-on to the R4 Notes server, and was an immediate hit with developers. Domino made six of the seven key features (all except replication, and Lotus later released Domino Off-Line Services to take care of that), which had made Notes into a success available to browser users.

The next releases of Notes and Domino continued to embrace evolving standard technologies. By the time Notes and Domino Release 5 shipped in 1999, Domino was a mature and robust platform supporting proprietary technologies such as @formulas and LotusScript and standard technologies such as HTML, JavaScript, Java, and XML, delivering solutions to both the proprietary Notes client and to standard Web browsers. Few, if any, software products have as long a history as Notes and Domino. And few, if any, software product visions have been so successful or so successfully adapted to new technologies.

Notes and Domino 6

Notes and Domino 6 follows in the tradition of preceding releases, adding more power and flexibility for end-users, system administrators, and developers, and adding support for even more standard technologies such as SOAP, UDDI, and WSDL. The IBM development team continues to build on the foundations that were established by the earlier releases, maintaining a level of backward compatibility that is probably unparalleled in any other application development environment. Almost all applications written for previous releases, all the way back to Release 1 from 1989, run on Notes and Domino 6 exactly as they did before. This is all the more remarkable for Notes and Domino 6 because IBM engineers completely rewrote a key component of almost all applications, the engine for executing @Formulas. The new engine is considerably faster than the old version, and it adds many new features to make programming easier, and yet fully compatible with previous versions.

Some of the key new features in Notes and Domino 6 are

- ♦ Looping in the function language
- ♦ Accessing list elements with array syntax in the Formula language
- ♦ Functions @GetField, @ThisName, and @ThisValue for referring to a field and field value without any hard-coding
- ♦ Improved list processing functions in the Formula language
- ♦ Improved error processing in the Formula language
- ♦ Remote debugging of agents
- ♦ Trusted agents accessing data on remote servers
- ♦ Boolean and Byte data types in LotusScript
- ♦ NotesXML, NotesXSL, NotesDOM, NotesSAX, and NotesStream classes in LotusScript for processing XML data
- ♦ NotesUIScheduler class in LotusScript for embedded schedulers
- ♦ NotesNoteCollection class in LotusScript for working with design elements or other types of non-data notes in a database
- ♦ New NotesRichText classes in LotusScript for navigating and editing rich text items
- ♦ Separate event handling for Notes clients and browsers
- ♦ The InViewEdit event allowing users to edit documents in views without opening or previewing them in the Notes client editor

All of these features, and many others, will be covered in-depth in later chapters. The preface contains a listing of these features, so if your primary interest in this book is catching up on what's new, you may want to make that your focal point. However, even developers with extensive Notes and Domino R5 development experience will find valuable insights into many of the older Notes and Domino programming features throughout this book.



