Success on Sun

Software Development Case Study

Federal National Mortgage Association (Fannie Mae)

Company Background

Chartered by Congress, Fannie Mae is the nation's largest source of home mortgage funds. Mortgages are purchased from primary lenders and sold to brokerage houses and investment funds. Headquartered in Washington, D.C., Fannie Mae includes five regional offices nationwide, with 3450 employees producing over $760 billion (U.S.) annually. Boasting over $2 billion (U.S.) in after tax profit, Fannie Mae received acclaim as the most profitable company per employee in 1994.

Over time, Fannie Mae has installed a new network of over 1000 Sun® SPARCstation™, SPARCserver™, and SPARCcenter™ systems. The company's central information system (IS) organization utilizes this predominantly Sun UNIX® environment to develop, maintain, and deploy mission-critical financial applications, such as its new Financial Management Information System (FMIS). Through the success of its new development environment and processes, Fannie Mae realizes that software development has, and will continue to play a key role in the organization's ability to maintain its edge in the financial services industry.

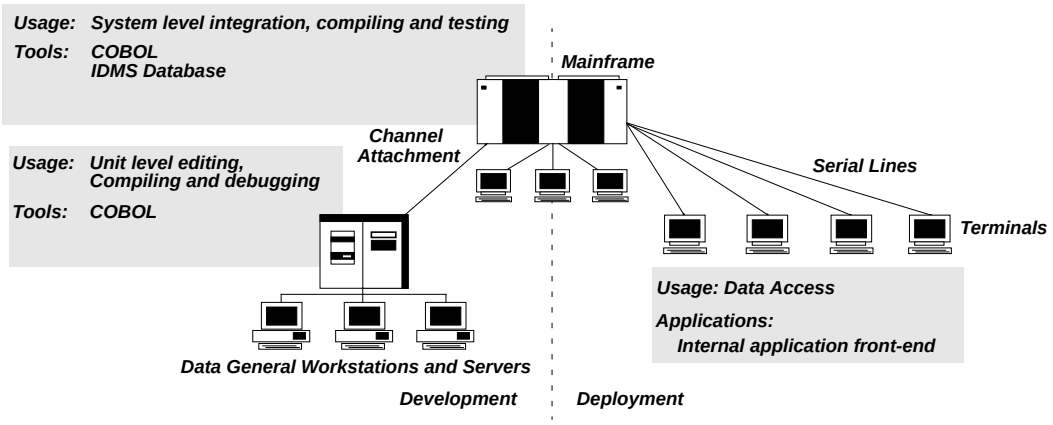
Company Name:	Fannie Mae
Industry:	Financial
Annual Revenues:	Approximately $760 billion (U.S.)
Headquarters:	Washington, D.C.
Key Development Issues:	• Need to get applications to market quickly • Need to adapt to rapidly changing market environments
Key Development Results:	• Shorter software development cycle • Faster problem isolation and resolution • Improved decision making and responsiveness
Application:	Financial Management Information System (FMIS)
Application Type:	Real-time decision support system
Development Environment:	**Hardware** SPARCserver 690 SPARCserver 10 SPARCserver 20 **Software** Solaris 2.3 WingZ SPARCworks C SPARCworks C++ SYBASE RDBMS SYBASE APT Workbench SYBASE APT DWB

Sun

Old Development and Deployment Environment

For years, Fannie Mae developed and deployed all of its applications in a timeshared mainframe environment. An IBM 3090 mainframe, several thousand Data General workstations and servers, and thousands of 3270 terminals were used to develop financial analysis, planning, and mortgage processing applications. Data General servers acting as gateways to the mainframe were connected via channel attachment, while the terminals used serial lines. Workstation users accessed the mainframe via 3270 emulation.

Old Development and Deployment Environment



The IBM 3090 was timeshared between development, testing, and deployment activities. Because they did not have dedicated system resources, developers were forced to off-load portions of their project development from the mainframe to servers and workstations. Using a traditional mainframe coding methodology, programmers edited, compiled, and unit tested their 3GL-based COBOL code on either the mainframe or a local Data General system. This process resulted in a fragmented development environment. Once compiled and unit tested, pieces were sent to the mainframe where they were assembled and the entire application tested. When verified, the application was moved to a designated production area on the mainframe. Data access was achieved through character-based 3270 terminals, limiting the manner in which data was presented to the user.

Key mainframe development characteristics

- **3GL**
- **Monolithic code**
- **Single-tasking development**
- **Fragmented development cycle, split between the mainframe and workstations**

Fannie Mae's development environment, and its associated development process, constrained application developers. The lack of windowing environments limited their ability to concurrently work on tasks. Batch mode transfers held developers at bay, causing significant delays in their edit/compile/test/debug cycle. Furthermore, as application functionality increased, Fannie Mae's procedural 3GL coding effort produced large and often unwieldy applications, consuming seventy percent of software development resources in maintenance. The use of both mainframe and workstation resources in Fannie Mae's development process fragmented the development environment and lengthened the software development process.

Development Issues

The inadequacies of Fannie Mae's development environment negatively impacted their ability to modify or create applications in a timely manner. This resulted in a series of development problems that hindered the effectiveness of their organization:

- **Lengthy development cycles**

Hampered by their development environment, processes, and lack of development tools, application developer productivity was compromised. Single-tasking operating systems and non-windowed environments prohibited developers from working on concurrent tasks, and eliminated their ability to simultaneously modify files and parallelize efforts.

Because development and production activities shared resources, programmers were forced to fragment their workspace, forcing the development process to be split among systems. With each edit / debug cycle, delays were experienced from batch mode transfers of application modules to the mainframe for compilation and testing, leaving developers idle for long periods of time. Since production applications had higher priority, turn-around times were slowed, further delaying development cycles.

- **Unable to respond quickly to change**

In the fast-paced financial services industry, new features are constantly needed in a short timeframe. Using a traditional coding methodology, the resulting programs were monolithic in structure. Without program modularity and tools to aid in problem isolation, it was difficult for developers to locate specific areas of code that needed modification. In addition, it was impossible for multiple programmers to concurrently modify source code files safely without potentially obviating fixes made by other developers, or cause new defects. This resulted in excessive edit / integrate / build / test cycles. Together, these issues resulted in a substantial backlog for requested changes and bug fixes to applications, resulting in the inability to quickly adapt to change.

- **Expensive software development costs**

Mainframe environments carry with them extremely high hardware and software maintenance costs. As more compute power was required, additional mainframe capacity or systems needed to be purchased. These costs, in combination with mainframe chargebacks for use of a shared resource, caused the computing platform to become the most significant cost in applications development.

Business Issues

Fannie Mae realized their development issues were having a direct impact on their success:

- **Lost business opportunity**

Fannie Mae's credit department handles underwriting—the process of deciding which mortgages should be purchased. Risk factors and demographics must be analyzed, and its marketing organization needs to be able to negotiate buy / sell terms.

The process of obtaining the data needed to make such analyses and decisions was cumbersome. Access to the mainframe was slow, and once retrieved, data could not be put in a format that was quickly understandable. The inability to present information in an intuitive, graphical manner meant increased analysis time and delayed decision making.

In addition, the inability to retrieve, analyze, and present data to customers managing the actual buying and selling of mortgages meant buy / sell decisions could not be accurately made within critical market windows. Fannie Mae discovered it was losing $1 million (U.S.) per day in lost opportunity and potential revenue.

- **Inability to quickly analyze data**

In 1989 Fannie Mae began applying financial modeling to aid their corporate-wide decision making process. Their complex models including hundreds of variables required extensive compute power. Since their mainframe was timeshared, financial models would take months to run, preventing Fannie Mae from taking maximum advantage of historical and current data.

Development Environment Analysis

When Fannie Mae concluded that their development environment was causing business losses, they began to reassess their requirements for a corporate-wide computing system. For a pilot project, Fannie Mae chose a new application, the Financial Management Information System (FMIS). If the solution worked, use FMIS as a model for the rest of the development organization.

> ***"Once we had demonstrated the effectiveness of client-server to quickly develop, deploy, and run powerful new applications, the floodgates opened and everyone was clamoring for new business solutions."***
>
> ***— Mike Williams***

FMIS was designed as a decision support system to help Fannie Mae with a new way to look at their existing business — referred to as their "Book of Business". It was critical that Fannie Mae have the ability to examine both financial data and entire business segments in a consistent, real-time manner. Developers had already identified the problems they encountered on a day-to-day basis. What was needed was an environment that was graphical, that let them perform multiple tasks at the same time, and that would allow them to develop applications faster.

Why Sun: The Choice for Client-Server Development

Based on the recommendation of consultants, Fannie Mae studied the cost of developing on mainframe versus client-server systems. They determined their current mainframe solution would take a team of 20 programmers 12 months to develop FMIS at a cost of nearly $7.8 million (U.S.). They also considered a client-server environment using a Sun SPARCserver 690 server, and 20 Sun desktops with a SYBASE relational database to handle back-end data collection, processing, and analysis. In addition, SYBASE's APT Workbench and APT Developer's Workbench (DWB) 4GL tools were selected for application development. Because information needed to be presented in an intuitive manner and they needed an application programming interface, the WingZ spreadsheet program was selected to present data to the user graphically. In the end, they determined the FMIS application could be developed by a smaller team in 6 months at a cost of only $1.5 million (U.S.)

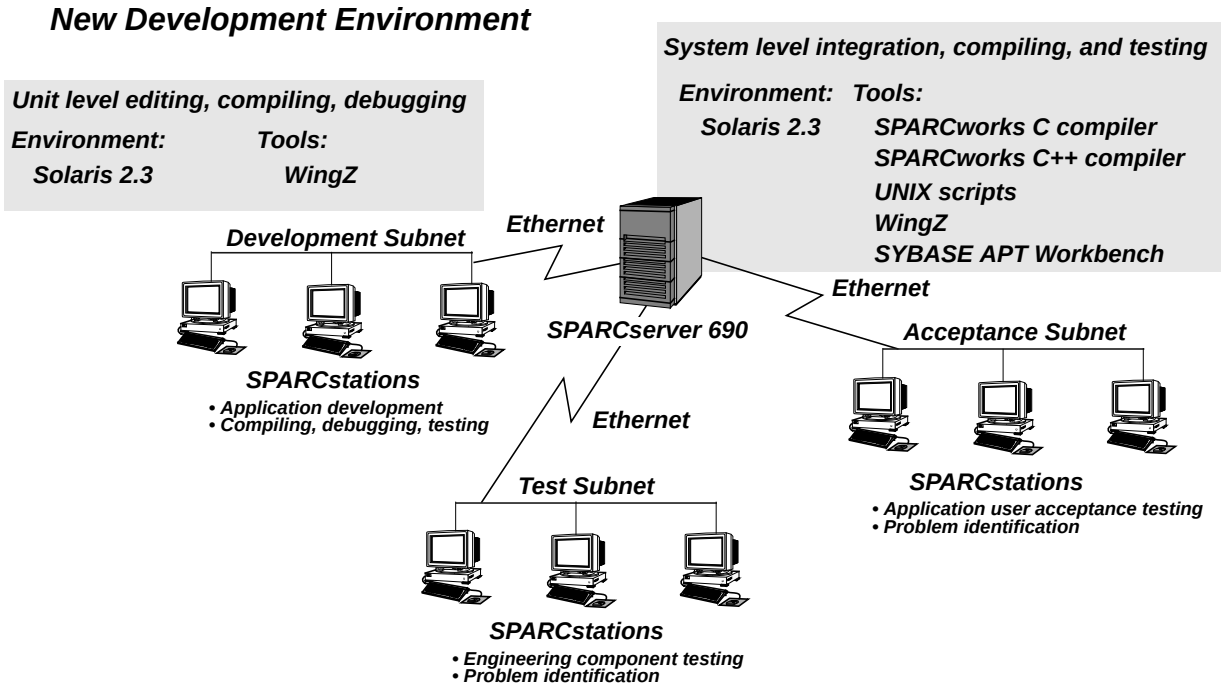
Fannie Mae realized the cost savings differential between their mainframe and a new Sun client-server environment had broad applicability. Lower cost client-server workstations significantly reduced capital expenditures for hardware, as well as hardware and software maintenance expenses. With a move to a distributed environment that provided dedicated computing resources, mainframe chargebacks would no longer be incurred, further reducing costs directly attributed to the hardware platform. Furthermore, using Sun's powerful Solaris multitasking operating system and advanced development tools, resource costs would be reduced. Smaller teams using flexible tools and an iterative development process could produce significantly more sophisticated applications in a shorter timeframe.

The New Development Environment

After successful pilot project, Fannie Mae decided to make a broader commitment to Sun systems for its entire development organization. Today, Fannie Mae's new corporate-wide development environment consists of 15 SPARCcenter 2000 and SPARCserver 690 servers, and 700 SPARCstation desktops divided into various subnets.

The FMIS development environment is representative of most Fannie Mae applications developed today. It consists of a SPARCserver 690 running Solaris 2.3 that is connected to each subnet via Ethernet. SYBASE APT Workbench and APT DWB 4GL are the central tools used for application development. When required, programmers utilize 3GL tools to augment functionality or integrate application components. The SPARCserver 690 server handles the compilation of application code using Sun's SPARCworks C and SPARCworks C++ compilers.

A SYBASE relational database stores financial information, and a WingZ API is used for the development of the application front-end that presents complex financial analyses in graphical form. Developers use a mix of SPARCstation 10s and SPARCstation 20s running Solaris 2.3 along with the SPARCworks toolset for developing, debugging, and testing application components.



Fannie Mae's new Sun environment has many characteristics that help its development organization to function more efficiently. Having moved from a mainframe to a UNIX client-server platform, developers now work in a multitasking, network-based, windowed environment that allows simultaneous access to files and the ability to work on concurrent tasks. Development and testing now use dedicated, high performance servers and desktops that offer faster compile times and increase productivity.

The company's shift from a centralized, monolithic mainframe environment to a distributed client-server computing platform has changed development team composition and function. Systems are divided among various financial application development groups, with each group partitioned into subnets. This has allowed Fannie Mae developers to specialize in specific tasks, such as development, testing, and user acceptance. This segregation of duties, combined with increased code modularity and an iterative development cycle, has allowed teams to perform their tasks in a parallel, collaborative manner. The resulting decrease in elapsed coding and testing time has enabled Fannie Mae to develop and modify applications more efficiently—significantly reducing backlog.

Fannie Mae not only changed its hardware configuration, it also switched from a traditional coding methodology to a 4GL rapid application development methodology. By altering the approach used to develop applications, Fannie Mae has seen dramatic changes in its development cycle. Dedicated desktop systems and SYBASE APT Workbench and APT DWB tools are used to develop 4GL applications. Additional lower level 3GL code is used to add functionality and integrate application modules via SPARCworks C and SPARCworks C++.

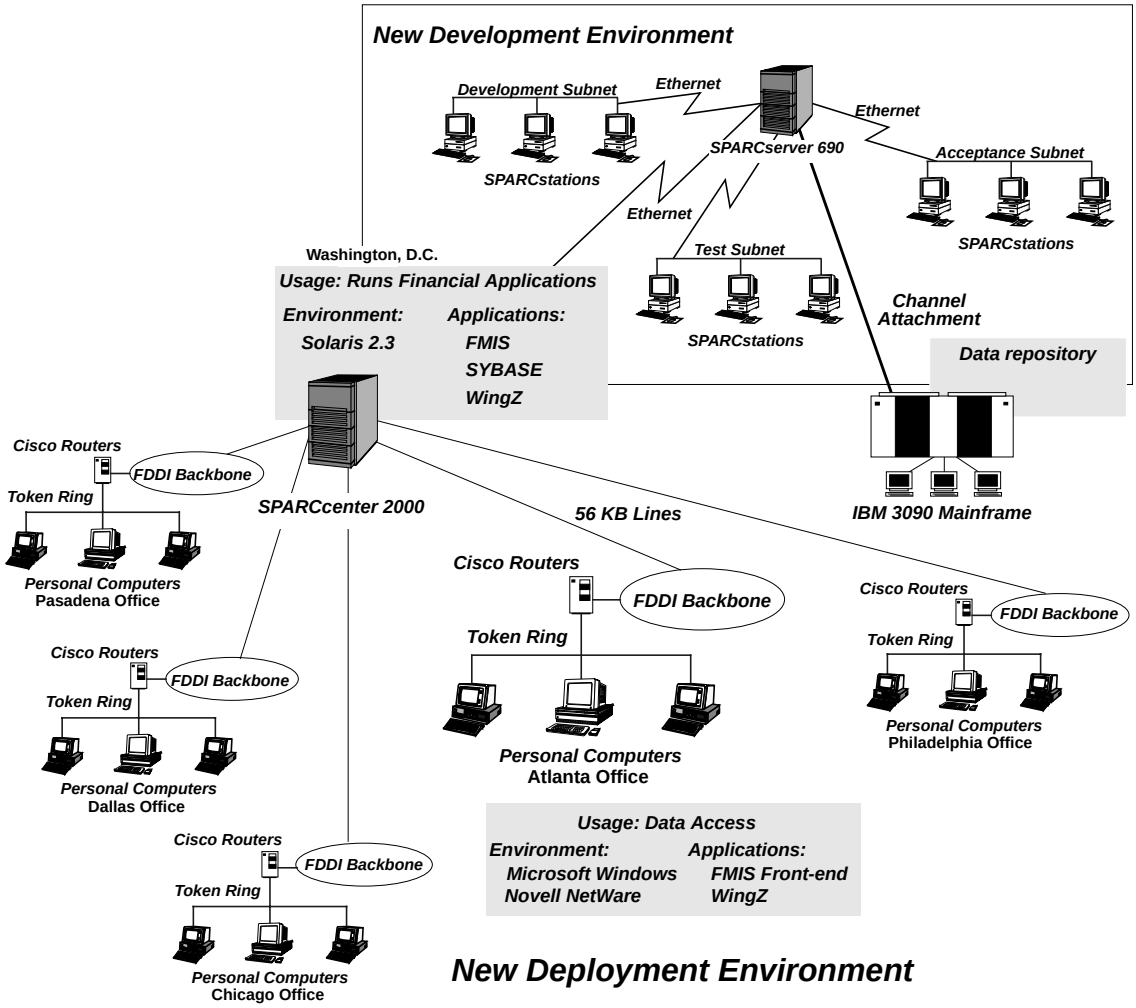
"There was a significant pent-up demand for more effective, timely delivery of business critical applications. With our evolving client-server network, we have not only experienced considerably faster development, but we are adding significantly more functionality to applications at a price that is comparable to — or less — than our proprietary approach."

— Mike Williams, Senior VP Customer Applications and Technology

SPARCworks graphical tools are used to locate application defects in an interactive manner, easing the debugging task and reducing development time. As each modification is made, testing groups are able to test changes immediately after they are implemented and quickly provide feedback to developers. The ability to compartmentalize and parallelize development and testing efforts is a dramatic change from the sequential process used in the mainframe environment — a change that has resulted in the ability to quickly develop new applications or modify existing ones.

The New Deployment Environment

Fannie Mae's new distributed computing platform gives developers the ability to stage and deploy applications and functionality where they are needed. For example, the FMIS application is used throughout all Fannie Mae offices. The centralized financial information database is stored on a SPARCcenter 2000 located in Washington, D.C. On-line access to the information is obtained via a PC front-end running WingZ and FMIS in each regional office. Each office connects its PCs using a Token Ring network running Novell NetWare. These PCs directly access data located in Washington via an FDDI backbone that transmits data over a 56 Kilobit line. The current IBM mainframe is now principally used as a data repository.



New Deployment Environment

Development Benefits

With its Sun client-server solution, Fannie Mae solved a variety of development problems:

- **Shortened software development cycle**

Using a Sun client-server development environment, programmers can now work concurrently on a variety of tasks. The multitasking features of Sun's Solaris operating system and graphical windowing system, allow developers to simultaneously modify files, parallelize processes, and decrease compile and link times. By employing dedicated desktop systems and servers, all tools and data are available on-line, resulting in a cohesive development process rather than a fragmented one. The move to a 4GL development environment has also required less coding time, resulting in a software development cycle has been reduced by fifty percent.

- **Faster problem isolation and resolution**

The shift from a traditional coding to a iterative development process, and a new 4GL tool set has enabled better application modularity. Because applications are modularized, features and functions are segmented, eliminating the need to modify or replace a substantial number of lines of code each time a defect is found. Modularity eases the incorporation of new features by minimizing their impact on the rest of the application. Advanced graphical tools, such as Sun's SPARCworks compiler and debugger products, allow programmers to quickly modify applications as needs dictate, and to find and fix problems in less time.

- **Application flexibility**

Flexibility is the ability to easily modify or augment applications. Because they have become more modular, Fannie Mae's applications developers have greater flexibility in deciding how to add new functionality. Also, the move to a 4GL process allows Fannie Mae to accommodate new features without application redesign. The rapid prototyping features of the 4GL SYBASE and WingZ tools provide faster module development and validation.

- **Improved teamwork**

Fannie Mae's new iterative development structure has shortened the software development life cycle by using dedicated resources, and has improved teamwork by encouraging better communication.

Code modularity has increased the parallelization of software development tasks, making developers more efficient. The ability to rapidly prototype new applications and features also allows development personnel to quickly iterate toward a solution that meets end-user needs.

- **Lower development costs**

With its new client-server platform, development processes and tools, Fannie Mae has changed the architecture of its development environment resulting in lower development costs. In the older, less flexible environment, $7.8 M (U.S.) and 240 man months would have been needed to develop FMIS. Fannie Mae was able to complete the effort in a Sun client-server environment in 120 man months at a cost of $1.5 M (U.S.) — a cost savings of $6.3 million (U.S.)

Business Benefits

The new development environment had direct impact on Fannie Mae's business:

- **Increased responsiveness**

By investing in a dedicated development environment, more CPU cycles are now available to production systems. As a result, data access is improved, and compute power is available for financial modeling. Today, financial models are run in days rather than months, enabling Fannie Mae to take better advantage of historical and current data, increasing their responsiveness to changing business conditions.

- **Improved time-to-market**

The shortened software development life cycle has enabled Fannie Mae to produce applications more quickly. In the last 4 years they have developed and deployed over 60 financial applications, reducing application time-to-market to one-half that observed in the old mainframe environment. Not one mainframe application has been developed since 1991.

- **Increased revenue**

An easy to use GUI and the graphical presentation of financial data has financial data to be directly accessed from on-line, dedicated servers. Now, data can be analyzed more quickly, permitting better decisions to be made quickly. With their new environment, Fannie Mae is better able to seize market opportunities, and today generates approximately $5 million (U.S.) per day from recaptured business opportunities.

Summary

Fannie Mae recognized that access and distribution of accurate, up-to-date financial information, together with powerful analysis tools, was a critical factor in running a successful business. Financial information, such as interest rates and market conditions needed to be integrated into their decision making process as quickly as possible. The ability to quickly access this data directly for buy/sell decisions, financial modelling, or business segment analysis, was essential to reduce missed opportunities and lost revenue.

These observations drove Fannie Mae to restructure their development environment, the way financial applications were developed, and how information was presented to the user. To accomplish these goals, Fannie Mae replaced its mainframe environment with distributed client-server systems for both development and deployment. By employing client-server technology, simultaneous access to a wide variety of financial information, and the ability to quickly develop applications that can take advantage of a powerful computing environment are now possible. By exploiting the multitasking and windowing capabilities of s Solaris, Fannie Mae's programming staff is able to work concurrently on several tasks, decrease application development time and time-to-market.

The end-user graphical user interface used in the FMIS application is one example of where the use of new technologies has had a significant impact on business revenue. By employing a SYBASE relational database and APT Workbench 4GL tools, FMIS is able to graphically present time-sensitive and critical financial data using WingZ as a user interface. Able to directly access financial information from an on-line, dedicated server, data can be analyzed more quickly, allowing better informed decisions to be made quickly.

The higher throughput and flexibility offered with Sun systems, combined with powerful development tools, dramatically reduced the cost of developing financial software. Fannie Mae's new Sun-based development and deployment environments also reduced the cost of obtaining, managing, and analyzing business information. Able to parallelize efforts, business can continue without interrupting development. The Sun client-server solution combined with SunSoft SPARCworks tools has turned Fannie Mae's development bottleneck into an effective, efficient, and flexible application development environment.

Sun Microsystems
Computer Corporation

A Sun Microsystems, Inc. Business

Australia: (02) 844 5000
Belgium: +32 2 716 79 11
Brazil: 011-55-11-887-9011
Canada: 416 477-6745
C.I.S.: 7-502-256-5470
Finland: +358-0-525561
France: (1) 30 67 50 00

Germany: (0) 89-46 00 8-0
Greece: +30-1-689-2210
Hong Kong: 852 802 4188
Hungary: 36-1-202-4415
Ireland: +353-1-6684377
Italy: 039 60551
Japan: (03) 5717-5000
Korea: 822-563-8700

Latin America: + 1415 688-9464
Mexico: 011-525-580-5229
The Netherlands: 033 501234
New Zealand: (04) 499 2344
Nordic Countries: +46 (0) 8 623 90 00
PRC: 861-8492828
Poland: 48-2-658-4535
Singapore: 224 3388

South Africa: (2711) 805-4305
Spain: (91) 5551648
Switzerland: (01) 825 71 11
Taiwan: 2-514-0567
UAE: +971-4-366-333
UK: 0276 20444
United States: +1 800 821-4643
Venezuela: 011-582-285-6640

Worldwide Headquarters:
+1 415 960-1300
Intercontinental Sales:
+1 415 688-9000

Printed in USA