



New Securities-Trading Platform Delivers Rich User Experience and Greater Flexibility

Overview

Country or Region: United States
Industry: Financial services

Customer Profile

Founded in 1999, UNX provides electronic-trading tools and agency brokerage services for institutional investors, brokerage firms, and third-party vendors. The company is based in Burbank, California, and has another office in New York.

Business Situation

In developing its new execution management system, UNX was committed to delivering robust capabilities and reliable performance, within a dynamic and modular framework.

Solution

UNX used the Microsoft Visual Studio 2008 development system and the Microsoft .NET Framework 3.5 to build Catalyst, an open-technology desktop framework and set of add-ins for trade execution management.

Benefits

- Rich and responsive user experience
- Solid reliability
- Superior flexibility and extensibility through an SDK
- Rapid time-to-market
- Increased competitive advantage

“The result for our customers is a solution that is highly extensible and facilitates the rapid development of new functionality—including by developers who may not know C# or other managed-code languages.”

Alex Shindich, Vice President of Product Management, UNX

UNX LLC (UNX), which provides trading tools and other services for institutional investors and brokers, wanted to develop a next-generation execution management system (EMS). The company used the Microsoft Visual Studio 2008 development system and the Microsoft .NET Framework 3.5 to build Catalyst, a desktop application for the Windows operating system that UNX’s customers can easily adapt to meet their needs. Based on the Windows Presentation Foundation, Catalyst uses the Managed AddIn Framework to run all functionality as loadable add-ins, and it supports scripts in languages such as Ruby and Python through the use of the Dynamic Language Runtime. Through Catalyst, UNX is delivering the performance, flexibility, and reliability that its customers require along with the ability to rapidly extend and customize solutions—in turn helping UNX better meet customer needs and win new business.

“The .NET Framework 3.5 and Windows Presentation Foundation provided a way to deliver the performance, reliability, flexibility, and access to the latest programming technology that our customers require—and an opportunity to build an open platform in a manner that is more meaningful than what our competitors offer.”

Jay Hatho, Chief Technology Officer, UNX

Situation

Founded in 1999, UNX provides sophisticated electronic-trading tools and agency brokerage services for institutional investors, hedge funds, brokerage firms, and third-party vendors. The company’s solutions streamline trading, helping traders and buy-side investors find the best execution opportunities, minimize market impact, reduce transaction costs, and manage today’s fragmented liquidity to their advantage.

From its inception, UNX has employed its extensive knowledge of market structure and advanced programming techniques to develop sophisticated front-end trading tools and back-end services that work together to help institutional traders maximize their performance. Several of the company’s core technology solutions are already in their fifth generation, and UNX constantly evaluates new technologies to determine how they can be used to better meet client needs.

UNX set out to develop a next-generation execution management system (EMS) that would provide state-of-the-art execution management and active trading capabilities across a large network of broker-dealers and liquidity venues. “Our clients demand uncompromised performance and reliability, but they also need a very high degree of flexibility so that they can adapt to changing market conditions and structures quickly,” says Alex Shindich, Vice President of Product Management at UNX. “To meet those needs, we set out to build a dynamic application framework that uses open standards to achieve a broker-neutral, fully extensible, and highly customizable desktop trading environment.”

Solution

UNX built Catalyst, a desktop application for the Windows operating system that is implemented as an open-ended framework and a set of add-ins to help traders manage and execute their orders. Built using the Microsoft Visual Studio 2008 development system, Catalyst is based on Windows Presentation Foundation and takes advantage of the Managed AddIn Framework (MAF) within the Microsoft .NET Framework to run all functionality as loadable add-ins. This makes it easily configurable and extensible to meet most customer needs. Additionally, through the use of the Dynamic Language Runtime (DLR), developers can build new add-ins using dynamic languages such as Ruby and Python.

Shindich says, “Every Catalyst feature, whether it has a user interface or not, runs as an add-in, with some add-ins providing services that others use. We’re even running the Dynamic Language Runtime as an add-in, which facilitates the development of extensions using dynamic languages such as Ruby and Python. The result for our customers is a solution that is highly extensible and facilitates the rapid development of new functionality—including by developers who may not know C# or other managed-code languages.”

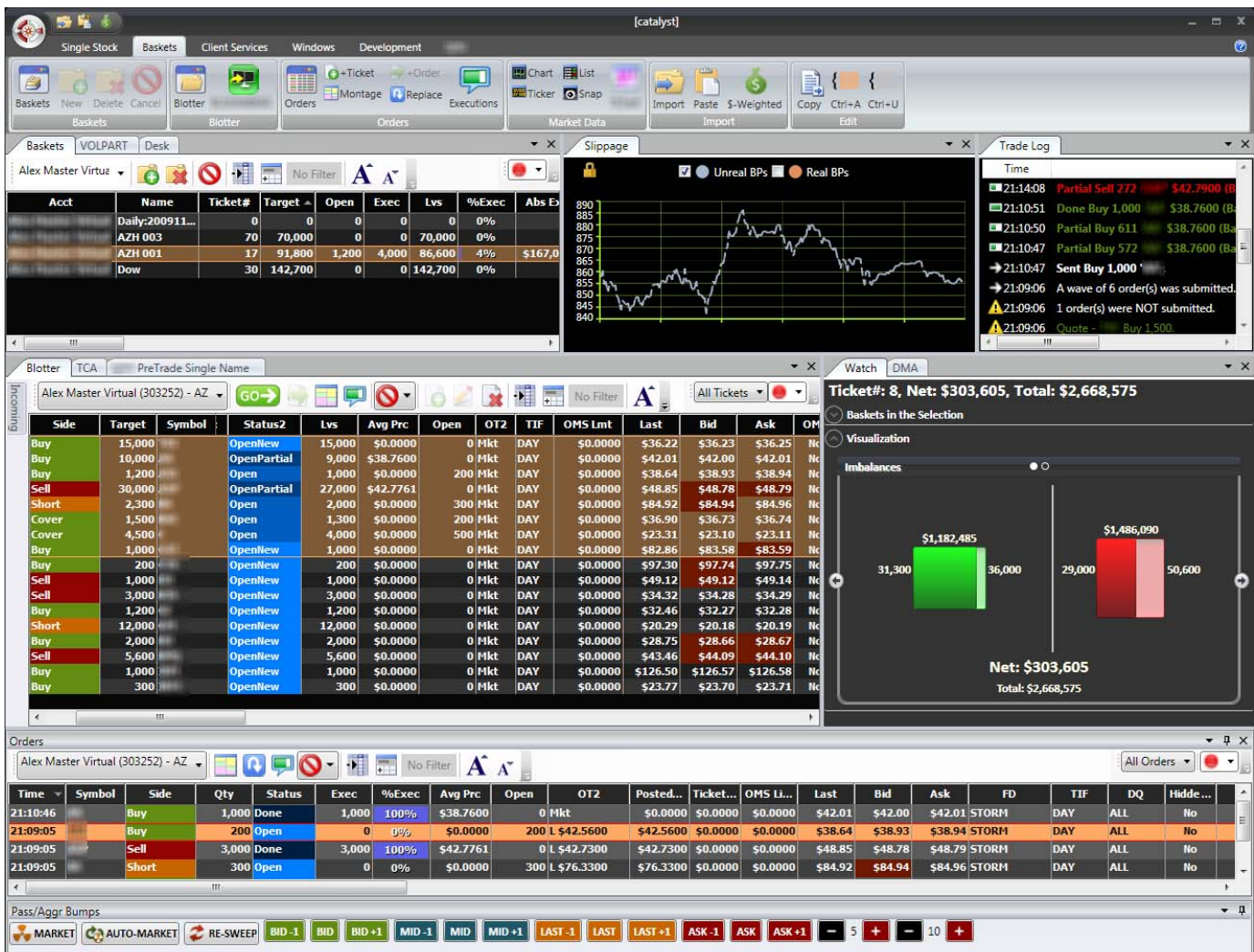
Platform Selection and Development Process

In planning to build Catalyst, UNX’s first step was to choose the right technology platform, and the company found several reasons to use the .NET Framework instead of Java. “Most of our competitors are heavy users of Java technology,” says Jay Hatho, Chief Technology Officer at UNX. “However, when compared to the .NET Framework, Java doesn’t support as

Figure 1. Because Catalyst add-ins can be developed using any number of programming languages, customers can easily extend the solution to meet their unique—and future—needs.

many other programming languages, makes it harder to find skilled developers for a project, and has slower user interface performance. The .NET Framework 3.5 and Windows Presentation Foundation provided a way to deliver the performance, reliability, flexibility, and access to the latest programming technology that our customers require—and an opportunity to build an open platform in a manner that is more meaningful than what our competitors offer.”

The UNX development team used Microsoft Visual Studio Team System 2008 Team Suite as an integrated development environment and Visual Studio Team System 2008 Team Foundation Server for source-code control and bug tracking. UNX employed an agile development process, taking advantage of the Microsoft framework for test-driven development. Catalyst is now in production and is being actively used by a diverse group of customers.



“We have hundreds of production servers and are in the process of upgrading many of them to Windows Server 2008 R2 and SQL Server 2008, which we expect to deliver significant gains in performance and reliability.”

Jay Hatho, Chief Technology Officer, UNX

How It Works

As an open application framework, Catalyst provides comprehensive window hosting and management APIs, which developers documented using Sandcastle—an open source tool on www.codeplex.com. The solution makes heavy use of Extended Application Markup Language (XAML) to provide a fast, customizable method of defining graphical user interface (GUI) screens—an important capability for UNX customers.

“Every broker and execution venue tries to distinguish itself by offering extra options for order routing and execution,” says Shindich. “However, each of those options has to find its way to end users, which in the past required constant UI [user interface] updates. With Windows Presentation Foundation, we’re able to load XAML that describes UI screens on-the-fly from the file system, which means that the core application framework doesn’t need to change when the UI does. We also use Windows Presentation Foundation to implement core features such as tabbed, docked, and floating windows—as well as unique presentations of trading data, such as using tooltips to show dollar imbalances.”

Catalyst’s add-in architecture is supported through use of the Managed AddIn Framework, a set of .NET Framework 3.5 assemblies that provide a way for developers to build extensibility into client applications by enabling add-ins to be written for a host. MAF delivers that ability by constructing a communication pipeline between the host and the add-in.

“Using the Managed AddIn Framework, we were able to create a composite

application that allows the loading of dynamic extensions via dependency injection,” says Hatho. “When the core application starts, the Managed AddIn Framework searches for .NET Framework assemblies, loads those assemblies, and activates their classes. With the Managed AddIn Framework, we’re able to provide an extensible application framework that is solid and robust, but also very dynamic.”

UNX and third-party vendors have already built more than two dozen add-ins for Catalyst, and this number continues to grow. One UNX add-in uses the Dynamic Language Runtime to support other add-ins written in dynamic scripting languages such as Ruby and Python. “With the Dynamic Language Runtime, we’re enabling developers who prefer dynamic open source languages to create Catalyst extensions,” says Shindich. “This is exciting to me from a product management perspective because it greatly expands the number of people who can customize our product.”

Back-End Services

Because Catalyst is a software-plus-services solution, the desktop framework uses Windows Communication Foundation to communicate with Catalyst servers hosted at the UNX data center. The transaction engine was written using C# and is hosted within Internet Information Services 7.0 running on the Windows Server 2008 operating system. The Web servers communicate with database servers that run Windows Server 2003 and Microsoft SQL Server 2000 data management software.

“We have hundreds of production servers and are in the process of upgrading many of them to Windows Server 2008 R2 and SQL Server 2008, which we

“We have enjoyed a substantial increase in the speed of development that would not have been possible without the use of Visual C# 3.0, the .NET Framework 3.5, and Visual Studio 2008, all of which help to maximize developer productivity.”

Jay Hatho, Chief Technology Officer, UNX

expect to deliver significant gains in performance and reliability,” says Hatho. “In addition, because most of our servers are multicore and/or multiprocessor systems, we’re also looking at how we can take advantage of the Microsoft Parallel Computing Platform for further performance and scalability gains.”

Benefits

By building Catalyst as a desktop application for the Windows operating system, UNX is delivering the performance, reliability, and flexibility that its customers require. Furthermore, the approach makes it possible for the company to develop new functionality significantly faster than before and provides greater agility in releasing and deploying new features to customers. Catalyst is helping UNX to capture the attention of prospective customers and brokers because it is an open and flexible trading solution that promises to change the global EMS landscape.

Rich and Responsive End-User Experience

With Catalyst, UNX is providing a rich and responsive end-user experience. “As an application based on Windows Presentation Foundation, Catalyst automatically takes advantage of modern graphics hardware to provide a highly rich and responsive user experience—something that is of paramount importance to customers,” says Hatho. “What’s more, with Windows Presentation Foundation, we can do things that would have been virtually impossible any other way, such as showing dollar imbalances using tooltips.”

Solid Reliability

Because of its choice of technologies, UNX is able to deliver the reliability that its customers demand. “The Managed

AddIn Framework gave us a way to create an extensible application framework that allows new features to be added easily without jeopardizing overall stability,” says Hatho. “It builds on existing .NET Framework features to enable the core Catalyst framework to dynamically and reliably discover, load, and communicate with add-ins. Using Microsoft technologies from the desktop to the server allowed us to create an environment that meets the high level of reliability required of applications in the capital markets.”

Superior Flexibility and Extensibility

Catalyst provides the flexibility typically demanded by financial institutions that require solutions tailored to their needs. “Most of our customers aren’t happy with packaged software,” says Shindich. “Instead, they want solutions that are customized to their businesses. One of the distinguishing features of Catalyst is the ease with which it can be customized and extended to meet customer needs. All UNX trading products developed on the new platform will offer APIs for automation and data manipulation. Additionally, any developers of custom functionality will be able to register extension points and APIs for their own add-ins. Using our software development kit, developers of other add-ins will be able to make use of registered APIs and extension points to modify the application’s behavior and to create new custom functions.

“For example,” continues Shindich, “a customer could build an add-in that creates custom buttons for filtering a basket of names according to the liquidity of those names. Similarly, a customer could develop a custom confirmation prompt that replaces the standard prompt, or could develop a custom

“The extensibility of Catalyst is one of the major draws for prospective customers; we’re seeing lots of interest because of how Catalyst provides increased control over trading screens that are presented to users.”

Alex Shindich, Vice President of Product Management, UNX

submission screen that uses custom analytics. Catalyst comes with an SDK that streamlines the process of add-in creation. Developers also can utilize the entire wealth of .NET tools and libraries, including those offered by Microsoft, third parties, and open source projects.”

Rapid Time-to-Market

Using the latest Microsoft technology, UNX created a highly functional, extensible solution much faster than was possible with the use of previous-generation technology. “We have enjoyed a substantial increase in the speed of development that would not have been possible without the use of Visual C# 3.0, the .NET Framework 3.5, and Visual Studio 2008, all of which help to maximize developer productivity,” says Hatho.

Hatho also expects a 30 to 40 percent decrease in development times for both new add-ins and enhancements to the core Catalyst framework. “Even more important,” he adds, “we can do things that we would never have contemplated in the past. For example, we never would have tried to provide unique presentations of trading data through tooltips because it would have taken entirely too much work. With Windows Presentation Foundation, the implementation becomes almost trivial. We were able to accomplish more with Windows Presentation Foundation in six months than we ever achieved with Microsoft Foundation Classes.”

Increased Competitive Advantage

Catalyst is generating a great deal of interest among major institutions in the financial services industry, including several leading Wall Street firms. “The extensibility of Catalyst is one of the major draws for prospective customers;

we’re seeing lots of interest because of how Catalyst provides increased control over trading screens that are presented to users,” says Shindich. “In addition, customers realize that the ability to develop add-ins using a broad range of languages will greatly reduce the learning curve. Users of our competitors’ products frequently have to hire specialists in proprietary scripting languages or development frameworks, whereas our use of .NET technology enables mainstream developers to rapidly create value-added content for the application.”

For More Information

For more information about Microsoft products and services, call the Microsoft Sales Information Center at (800) 426-9400. In Canada, call the Microsoft Canada Information Centre at (877) 568-2495. Customers in the United States and Canada who are deaf or hard-of-hearing can reach Microsoft text telephone (TTY/TDD) services at (800) 892-5234. Outside the 50 United States and Canada, please contact your local Microsoft subsidiary. To access information using the World Wide Web, go to:

www.microsoft.com

For more information about UNIX, visit the Web site at:

www.unx.com

Microsoft Visual Studio 2008

Microsoft Visual Studio 2008 is the world's most popular development environment for designing, developing, and testing next-generation Windows-based solutions and Web applications and services. By improving the development experience for Windows, the Web, mobile devices, and Microsoft Office, Visual Studio 2008 helps organizations deliver a variety of solutions more productively than ever before. Visual Studio Team System expands the product line with new software tools that enable greater communication and collaboration throughout the development life cycle. With Visual Studio 2008, businesses can deliver modern service-oriented solutions more efficiently.

For more information about Visual Studio 2008, go to:

msdn.microsoft.com/vstudio

Software and Services

- Microsoft Visual Studio
 - Microsoft Visual Studio Team System 2008 Team Suite
 - Microsoft Visual Studio Team System 2008 Team Foundation Server
- Microsoft Server Product Portfolio
 - Windows Server 2008
 - Microsoft SQL Server 2008

• Technologies

- Dynamic Language Runtime
- Managed AddIn Framework
- Microsoft .NET Framework 3.5
- Windows Communication Foundation
- Windows Presentation Foundation