



Overview

Country or Region: United States

Industry: Securities regulation

Customer Profile

Founded in 1940 and based in Washington, D.C., NASD employs nearly 2,000 people and is the world's leading private-sector provider of financial regulatory services.

Business Situation

NASD IT executives wanted a practical way to boost the functionality of an online tool popular with investors and to simplify "rebranding" of the tool for use by diverse audiences.

Solution

A team of three to four developers used the Microsoft® Visual Studio® 2005 development system to enhance automation, implement rebranding, and deploy a new version of the tool running against a Microsoft SQL Server™ 2000 database.

Benefits

- 20 percent faster design
- Significantly faster development
- Greater accessibility
- Twofold increase in tool usage
- Enhanced ability to fulfill the association's mission

Securities-Industry Regulator Streamlines Development, Delivers Unique Tool

"By enabling users to figure out expenses so easily, the NASD Mutual Fund Expense Analyzer has become a one-of-a-kind tool for investors."

Bob Adeli, Program Director, NASD

Seeking a practical way to boost the functionality and versatility of an online tool used by millions of investors, IT executives at NASD deployed a new version of the tool with the help of the Microsoft® Visual Studio® 2005 development system. Enlisting the product's support for the master pages feature of Microsoft ASP.NET version 2.0, the Enterprise Library for .NET Framework, and other productivity enhancements, NASD developers met their objectives in a development cycle that took just over half the time required by similar projects in the past. The updated tool is faster, easier to use, more accurate in its results, and more easily customizable for use in diverse environments. In addition, online visits to the tool have doubled. As a result, NASD has expanded its visibility to investors and is better able to fulfill its mission of investor education and protection.

“With ASP.NET 2.0 and Visual Studio 2005, we would finally have a practical way to implement rebranding capabilities.”

Michael Scheidt, Software Developer Lead, NASD

“We are providing greater functionality without a greater burden on computing or network resources and without any drawback in performance.”

Bill Cunnane, Director of Systems Engineering, NASD

Situation

As the world’s leading private-sector provider of financial regulatory services, NASD has worked to bring integrity to U.S. markets and confidence to investors for more than 60 years. Originally known as the National Association of Securities Dealers, NASD oversees the activities of more than 5,100 brokerage firms, their 100,000 branch offices, and 650,000 registered securities representatives throughout the United States. The association also provides an assortment of technology-based services, programs, and tools to stock markets, exchanges, brokers, and individual investors.

Among the most widely used of the NASD tools are online analyzers that help individuals to make smart investment choices. In response to user feedback and to recommendations from a task force convened by the U.S. Securities and Exchange Commission in 2003, NASD IT executives wanted to update a number of those tools for greater versatility, functionality, and ease of use.

In particular, the executives sought to enhance a tool known as the NASD Mutual Fund Expense Analyzer, which since its introduction in January 2003 had helped millions of investors to determine the total fees and expenses associated with their mutual funds and exchange-traded funds.

As NASD Program Director Bob Adeli explains, the NASD Mutual Fund Expense Analyzer is central to the mission of his organization. “With this tool,” he says, “we can provide better information to individuals so that they in turn can make smarter investment decisions.”

According to Adeli, the initial version of the NASD Mutual Fund Expense Analyzer was reliable, functional, and well received by users. But after gathering extensive user feedback, he and his colleagues discovered

an opportunity for improvement. Specifically, they wanted the tool to provide information in a more automated fashion—enabling users to calculate fees and expenses more easily and to see the results more clearly.

“In the original usage model, users entered information such as dollar contribution, investment period, fees, and expenses from the prospectus of a selected fund, and the analyzer calculated their total fee and expense liability over that period,” Adeli explains. “But we wanted to relieve users of having to look up the fee and expense information, ultimately saving them steps and ensuring more accurate results.”

Adeli and his team also wanted to incorporate “rebranding” capabilities into the tool—that is, to make it easily customizable for use in different environments. Such uses might include syndicated distribution to Web sites besides NASD.com and delivery to organizations and individuals in selected vertical-market segments. “By adding rebranding capabilities to the tool, we could make its powerful capabilities more available to more people and thereby better serve the association’s mission of investor education and protection,” Adeli explains.

To keep development and deployment costs to a minimum, team members decided to address both challenges—automation enhancements and rebranding—at the same time. However, according to NASD Software Developer Lead Michael Scheidt, this meant that the overall update had to wait until the team found an efficient way of tackling the traditionally tricky development issues required to make the tool rebrandable.

“Historically, we used a copy-and-paste approach, but that approach was problematic because it resulted in a split code base requiring server-side include directives for templating,” Scheidt explains. “Moreover, the

“Thanks to the support in Visual Studio 2005 for master pages, developers were able to focus on updating the tool’s functionality ... [helping] to cut 15 percent from the overall development time.”

Michael Scheidt, Software Developer Lead, NASD

include directives couldn’t be reviewed in design mode, and they, along with nested directives, significantly complicated debugging efforts.”

To summarize the problem, Scheidt says, “No matter how much we wanted to implement rebranding capabilities in the analyzer, we needed a more comprehensive and cost-effective way to do it.”

Solution

A breakthrough came when Adeli, Scheidt, and their colleagues discovered the master pages feature of Microsoft® ASP.NET version 2.0—and that the Microsoft Visual Studio® 2005 development system supported this feature. Although NASD was running an earlier version at the time, this discovery led the team to consider upgrading to Visual Studio 2005.

“We recognized that support for master pages would enable us to deploy a consistent presentation within a given version of the analyzer and a consistent functionality across all versions of the analyzer,” Scheidt says. “In other words, with ASP.NET 2.0 and Visual Studio 2005, we would finally have a practical way to implement rebranding capabilities.”

Team members also liked other aspects of Visual Studio 2005, beginning with the Enterprise Library for .NET Framework. As NASD Director of Systems Engineering Bill Cunnane explains, “The Enterprise Library was a plus for us because it provides a set of proven design patterns implemented in code and is available for use in any application. In addition, because full source code is available for the library, it can be modified to fit custom requirements—another plus for helping us to deploy timely updates.”

Scheidt and his colleagues also anticipated that enhancements to client-side validation

would enable them to minimize code volume and that the GridView control would enable them to more effectively meet business requirements for control of data formatting at the cell level. “In sum, we saw early on how the Enterprise Library, enhancements to client-side validation, and the GridView control would be very important to this project,” he notes.

Yet another reason that the NASD team members liked the idea of adopting Visual Studio 2005 was their familiarity with it, as the Mutual Fund Expense Analyzer was developed in earlier versions of the development system. That familiarity enabled a team of three and sometimes four developers to use the Visual C#® 2005 programming language in Visual Studio 2005 Professional Edition to create an updated version of the analyzer in just under six months.

Released in late 2005, the NASD Mutual Fund Expense Analyzer runs on the Microsoft Windows Server™ 2003 Standard Edition operating system with ASP.NET 2.0 and uses Web Forms pages. The analyzer runs against a Microsoft SQL Server™ 2000 Enterprise Edition database that stores information on more than 18,000 mutual funds and 160 exchange-traded funds and incorporates SQL Server Data Transformation Services for data scrubbing and data migration. Both Windows Server and SQL Server are part of Microsoft Windows Server System™ integrated server software.

Adeli says that NASD plans to release two more major online tools for investors and additional smaller tools updated with Visual Studio 2005. The team also will begin upgrading the SQL Server 2000 database to Microsoft SQL Server 2005. By upgrading to SQL Server 2005, Adeli and his colleagues hope to boost the scalability, availability, and performance of all the online tools.

“The Enterprise Library provided consistent error logging, consistent data access, and an intuitive way to implement encryption and decryption, making our development and deployment environment 30 percent more efficient.”

Michael Scheidt, Software Developer
Lead, NASD

Benefits

NASD IT executives report significant benefits from having used Visual Studio 2005 to update the NASD Mutual Fund Expense Analyzer. Development and deployment benefits include a more efficient development environment, a leaner code base, and greater flexibility to meet business requirements. Postdeployment benefits include the ability to offer investors a more powerful and accessible analyzer, an increase in investors' knowledge of NASD, and a greater understanding by investors of how the association can support them with education and information.

Efficient Development Environment

Scheidt says it was no surprise that developers were able to tackle the update efficiently with the help of Visual Studio 2005 and, in particular, the feature that persuaded the team to select the product: support for ASP.NET 2.0 master pages.

“Thanks to the support in Visual Studio 2005 for master pages, developers were able to focus on updating the tool's functionality instead of on the Web-site shell or the navigational context, for example,” Scheidt says. “This alone helped to cut 15 percent from the overall development time.”

As for the Enterprise Library for .NET Framework, Scheidt reports that it enabled developers to work more efficiently by eliminating the need to code classes and functions by hand.

“The Enterprise Library provided consistent error logging, consistent data access, and an intuitive way to implement encryption and decryption, making our development and deployment environment 30 percent more efficient than what it was during earlier updates,” Scheidt adds.

Minimal Code Base, Cost-Effective Maintenance

As Scheidt envisioned, the enhancements to client-side validation available in Visual Studio 2005 helped developers to keep code volume lean. “Enhancements to client-side validation may be the single most significant factor in helping us to minimize lines of code,” he says. “In fact, the code base of the new analyzer is virtually the same as that of the prior version even though the tool is significantly more powerful. We are providing greater functionality without a greater burden on computing or network resources and without any drawback in performance.”

Even better, Adeli adds, by avoiding an expansion in the code base, NASD also is saving on long-term maintenance costs. “Less code written today means less code to maintain tomorrow,” he points out.

Scheidt also was happy with the advantages provided by the GridView control capability in Visual Studio 2005. “The GridView control provided an enormous amount of flexibility in formatting data at the cell level,” he reports. “As a result, we were able to apply different colors and font formatting to different kinds of data, making results easier for users to grasp.”

Still other capabilities of Visual Studio 2005 emerged during development to help the team work more efficiently and effectively.

“Support for Autocompilation in ASP.NET 2.0 enabled developers to avoid having to generate multiple builds,” Scheidt reports. “What's more, improvements to IntelliSense® technology saved keyboard entry in the early stages of development. This helped us to reduce design time by about 20 percent.”

More Powerful, Accessible Tool for Investors

Adeli says that with the updates his team implemented by using Visual Studio 2005,

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 who are deaf or hard-of-hearing can reach Microsoft text telephone (TTY/TDD) services at (800) 892-5234 in the United States or (905) 568-9641 in Canada. 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 NASD products and services, visit the Web site at: www.nasd.com

the new version of the NASD Mutual Fund Expense Analyzer is significantly more powerful than the preceding analyzer because of its more automated nature. For example, the new analyzer:

- Accesses data from a database that interacts with more than 18,000 mutual funds, thereby eliminating the need for the user to enter expense data.
- Incorporates complex factors such as fund conversion from one class to another.
- Calculates interest and expenses for a period of up to 20 years, in contrast to the prior version of the tool, which covered only 1 year.
- Displays results to users in both graphic and tabular formats and provides more intuitive displays of fund performance and other details.

Most significant, Adeli emphasizes, is the ease with which users can take advantage of the tool's functionality. "Now, users need only to enter the amount of their estimated contributions and rate of return, and within seconds they can know how expenses will affect the bottom line for up to three funds at a time," Adeli says. "By enabling users to figure out expenses so easily, the NASD Mutual Fund Expense Analyzer has become a one-of-a-kind tool for investors."

Not surprisingly, after the updated NASD Mutual Fund Expense Analyzer was released, early tallies showed monthly user visits to NASD.com up by 200 percent.

"More traffic to NASD.com means that investors are viewing NASD as a resource for information," Adeli points out. "This is good for NASD and, more importantly, good for the investor community."

Microsoft Visual Studio 2005

Microsoft Visual Studio 2005 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 2005 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 2005, businesses can deliver modern service-oriented solutions more efficiently.

For more information about Visual Studio 2005, go to: msdn.microsoft.com/vstudio

Software and Services

- Microsoft Visual Studio 2005 Professional Edition
- Microsoft Windows Server System
 - Microsoft Windows Server 2003 Standard Edition
 - Microsoft SQL Server 2000 Enterprise Edition

■ Technologies

- Microsoft Enterprise Library for .NET Framework