



MICROSOFT .NET FRAMEWORK CUSTOMER SOLUTION

Major Airline Takes New Approach to Application Development with the Microsoft .NET Framework and Visual Studio .NET



Published: June 2001

By developing a new mobile travel-planning application with the Microsoft .NET Framework and Visual Studio.NET, Continental Airlines is reducing deployment time by 90 percent, eliminating the overhead of tightly coupled remote-procedure-call implementation, ensuring rapid and flexible scalability, and providing transparent interoperability with host systems. Moreover, with the .NET Framework and Visual Studio.NET, developers envision an entirely new approach to creating and updating applications in the future.

Solution Overview

Profile

The fifth-largest airline in the U.S., more than 2,200 departures daily to 133 domestic and 92 international destinations, serving more international cities than any other U.S. carrier.

Business Need

A way to quickly build and deploy mobile applications for distributing vital flight information on diverse devices to high-value customers.

Benefits

Reducing deployment time by 90 percent, eliminating the overhead of RPC implementation, ensuring rapid and flexible scalability, providing transparent interoperability with host systems.

Microsoft Technologies/Products

Microsoft .NET Framework
— ADO.NET
— Common Language Runtime
— COM Interop
Microsoft Visual Studio.NET
Microsoft® Windows® 2000
Advanced Server
Microsoft SQL Server™ 2000

Scenario

B2C Web Services

Situation

Recently, application developers at Continental Airlines established the goal of learning how to efficiently build and deploy mobile applications that use a single code base and run on multiple platforms. Their motivation stemmed from a challenge to strengthen Continental's position as a global travel-planning resource for high-value customers by enabling them to receive essential flight information on the device of their choice.

To help meet their goal Continental developers are using the Microsoft .NET Framework and Visual Studio.NET to build a multitier application accessible on the front end by a variety of user devices. The application incorporates a .NET-based middle tier and is integrated on the back end with SQL Server as well as other databases, including legacy systems.

According to Continental Director of Application Development Ferdy Khater, technical managers based their decision to use the .NET Framework and Visual Studio.NET on four criteria. First, they wanted to boost developer productivity in response to constant competitive pressures. "Every day it seems like there's something new on the Web for travelers," Khater remarks. "So those of us building Web-based tools from inside the industry need to do what we can to ensure our applications get to market quickly."

The managers also wanted an extended platform that would interoperate transparently with the diverse technology that's commonplace in a large enterprise like Continental, and they wanted higher performance than they were able to achieve using applications based on the prior



environment. Finally, they wanted the kind of scalability that could enable them to efficiently assemble and distribute a challenging form of content—highly dynamic and typically residing at three or four separate locations—over wireless networks.

Khater says that after reviewing the .NET Framework and Visual Studio.NET with these concerns in mind, he and his colleagues were confident of their decision. “We became convinced that this technology was fully capable of addressing all our concerns.”

Solution

Today, two Continental developers are using the .NET Framework and Visual Studio.NET to create the new application. The application isn't due for production release until the third quarter of 2001, but even early in development Khater and his colleagues began realizing the benefits of using an environment based on the .NET Framework and Visual Studio.NET.

Benefits

Deployment in Hours, Not Weeks

According to Khater, the .NET Framework is helping his team to shave 90 percent off deployment time by using “XCOPY deployment,” which means that developers design, code, and debug on the development platform and then simply copy their code to a staging server. Once there, the code is ready to run without requiring that the developers register DLLs or build COM or export packages—that is, without anyone having to worry about dependencies. “With the .NET Framework installed on the staging server, we can be confident that code will execute in the Common Language Runtime, and for us that means a dramatic gain in productivity,” Khater points out. “Compared with similar projects in the past, we're measuring deployment time in hours instead of weeks.”

Another productivity advantage comes from the .NET Framework type safety, as Applications Developer Derrick Blanco explains. “Whatever language we're working in, type safety lets us safely assume that external objects will be returned in a format we can use, helping to make our code and our development work a lot more efficient.” Also helping to streamline both code and development are several Visual Studio.NET capabilities. “The IntelliSense editor, for one, is very useful in getting new programmers up to speed,” Blanco says. “Moreover, the product's built-in designers, tools, and templates enable easy customization of the development environment.”

Further helping to reduce programming overhead, as Khater points out, is the fact that the entire .NET Framework is built on XML and SOAP. This means the developers can use XML Web services to expose flight schedules and related information without having to implement RPCs (remote procedure calls) between platforms. “We can simply expose our XML interfaces through schema definitions for anyone who wants to subscribe to the services,” he reports. “This includes not only our users but also vendors with whom we have joint-marketing agreements, such as our frequent-flyer program or flight-schedule sharing.”

Outstanding Scalability, Transparent Interoperability

With the inevitable growth of Continental's joint-marketing agreements and related services, and the customers who rely on them, it's essential that the mobile travel-planning application scale easily. And with its basis in .NET and the help of XCOPY deployment, that's exactly what the

“With the .NET Framework SDK installed on the staging server, we can be confident that code will execute in the Common Language Runtime, and for us that means a dramatic gain in productivity. Compared with similar projects in the past, we're measuring deployment time in hours instead of weeks.”

Ferdy Khater
Director of Application Development
Continental Airlines

“[Using the COM Interop tool to wrap internally developed C++ components] gives us managed code that calls and interoperates effectively with any unmanaged code on the mainframe. To do this without the .NET Framework would require a utility that easily could take us an additional six weeks to implement.”

Chris Wang
Application Developer
Continental Airlines

application does. "With XCOPY deployment, adding a server is easy because we don't have to bring down the whole site to do it," Khater says. "So during peak usage periods we can throttle the system by adding hardware and, once the peak has passed, redeploy it to where it's needed more."

In fact, the scalability of the mobile travel-planning application has been proven in recent stress-testing benchmarks. Simulating 200 concurrent connections, developers generated 80,000 simultaneous hits with a connection success rate of over 98 percent. CPU utilization was favorable as well—less than 50 percent under a maximum load. As Khater explains, two factors in particular contribute to these excellent numbers: having managed code and avoiding memory leaks, both of which are supported by the .NET Framework Common Language Runtime.

"Whatever language we're working in, type safety lets us safely assume that external objects will be returned in a format we can use, helping to make our code and our development work a lot more efficient."

Derrick Blanco
Application Developer
Continental Airlines

Growing the Technology Dynamically

Based on the results they've observed so far, Khater and his colleagues intend to move virtually all development work in their group to the .NET Framework and Visual Studio.NET as soon as it's practical. As Khater is careful to point out, however, his team's motivation for the move is based not so much on what the .NET Framework and Visual Studio.NET have already helped them to do, but on what they envision as an entirely new approach to developing future applications at Continental.

"Traditionally, it might take weeks or months to update applications in response to customer feedback, but with .NET we can work dynamically, growing our technology as it's being used," Khater says. "That's the best way I know how to bring our customers the kind of solutions they demand and deserve."

The Microsoft .NET Framework is a platform for building, deploying, and running XML Web services and applications. It provides a highly productive, standards-based, multi-language environment for integrating existing investments with next-generation applications and services as well as the agility to solve the challenges of deployment and operation of Internet-scale applications. The .NET Framework consists of two main parts: the common language runtime and a hierarchical set of unified class libraries that includes a componentized version of Active Server Pages called ASP.NET, a loosely coupled data access subsystem (ADO.NET), and an environment for building rich Windows applications (Windows Forms).

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 Resource Centre at (800) 563-9048. Outside the 50 United States and Canada, please contact your local Microsoft subsidiary. To access information using the World Wide Web, go to <http://www.microsoft.com/>. For more information on Continental Airlines, go to <http://www.continental.com/>.

© 2001 Microsoft Corporation. All rights reserved.

This case study is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Microsoft, BizTalk, the .NET logo, Visual Studio, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

