

# Mobile Solution with .NET Compact Framework Liberates Sales Professionals

Published: January 2008



***Mobitor implemented a new Sales Force Automation/ Customer Relationship Management system in C# using the Microsoft .NET Compact Framework running on Pocket PC devices, which communicates with a server and intermittently synchronizes its Microsoft SQL Server CE 2.0 database over a variety of wireless and wired connections. This enables users to work seamlessly in both a connected and disconnected mode. The new system is reliable and scalable, and helps agents close sales in the field.***

## Situation

According to Dr. Lakshman Watawala, CTO of Mobitor, the laptop computer has in general been a failure for insurance agents and other field salespersons. "You can do what you want to make the application useful, but most agents just won't carry the laptop with them to see clients. They say it's too heavy, and that it's a barrier to interacting with the client."

But this means agents have to return to their office to start and later complete the sales process. "Without some sort of computer in the field, the agent winds up working on paper at the client site, and then doing data entry back at the office. That's inefficient and error-prone. If there's more than one trip back and forth to the office, for example to look up premiums for various plans, it's a real impediment to closing the sale."

Fast Facts	
Number of developers to build application (total)	10
Number of months to build application (total)	18
Percent of development effort on Pocket PC Client	20
Maximum number of users per server	1000s

"That's showing up in the numbers. The ratio of sold life insurance policies to customer meetings has declined from 1 in 10 to 1 in 15 from 1990 to 2000. What's more,

Expense Ratios (this is a measure of cost to acquire new business expressed as a % of sales—it has nothing to do with claim payouts) for Property and Casualty insurance companies has increased significantly over the last 5 years. In many insurers, the ratio is up from 3% to 6%."

Mobitor hit on the idea of using a Microsoft® Pocket PC device in the field for insurance agents and other field sales professionals. The Pocket PC is capable of holding enough of a database to support the sales cycle even when disconnected. It can connect to a server over a variety of wired and wireless protocols, and run an attractive, "smart" client application. The Pocket PC can even capture signature data and support biometric security measures.

"We considered programming our mobile application in the Java language, specifically J2ME. We eliminated it due to its lack of native device support—we couldn't even do simple things with it like

## Solution Overview

### Customer Profile

Mobitor develops and sells an integrated suite of enterprise applications for conducting mobile interactions, transactions and services. Global enterprises use these applications to enable their mobile sales and service professionals to sell, service and provide information to their customers.

### Business Situation

Mobitor discovered that existing Internet- and laptop-based CRM systems were not effective for insurance agents. They wanted to build a mobile solution that would allow agents to close sales in the field.

### Solution

Mobitor built a new mobile field sales automation and customer relationship management system in C# using the Microsoft® Windows® .NET Compact Framework to run on Pocket PC, Pocket PC Phone Edition and Tablet PC devices. The devices synchronize with a server over WWAN, WLAN, or LAN connections, as available, using Microsoft SQL Server™ CE 2.0 synchronization and XML and Web services.

### Benefits

- Microsoft Visual Studio® .NET allowed Mobitor to integrate its XML-enabled Pocket PC components right into the IDE.
- SQL Server 2000 and SQL Server CE 2.0 deliver a reliable, high-speed, scalable solution for mobile support on the Pocket PC.
- Web Services allowed Mobitor to develop faster, reliable code to international standards.

### Software and Services

Microsoft® Pocket PC  
Microsoft SQL Server 2000  
Microsoft SQL Server CE 2.0  
Microsoft Visual Studio® .NET  
Microsoft Visual C#®  
Microsoft Exchange Server (opt.)  
Microsoft BizTalk® Server (opt.)

having a standard OK button on the title bar. Also, our application integrates and communicates with Pocket Inbox on the device to set appointments and contacts—that was not possible with Java.

"Subsequently we discovered that [Microsoft] Visual Studio® .NET is a much more productive development environment altogether.

## Solution

"At Mobitor we built a comprehensive mobile Sales Force Automation / Customer Relationship Management system called Mobile Business Manager™ (MBM). The MBM system is both customer- and process-centric. Sales processes are dynamically assembled by MBM based on customer information triggered from corporate systems. MBM delivers (or pushes) these sales process activities to the most appropriate mobile sales professional along with the requisite customer data, history, forms and documents to conduct business wherever the customer is. This ensures that not only is the most qualified and available mobile knowledge worker receiving the opportunity, but also that the tedious administrative task of searching for and collating the appropriate resource materials and forms is automated.

*"As a testimony to the success of our decision to use .NET and SQL Server CE, we are often accused of demonstrating functionality in our sales engagements that is too futuristic. Potential customers find it hard to believe that we have bundled so much capability in a mobile device."*

Dr. Lakshman Watawala  
Chief Technology Officer  
Mobitor



"As a result, salespeople can present the information necessary to encourage buying or cross-selling decisions exactly when the opportunity arises. Furthermore, MBM allows the sales professional to enter customer information and initiate a transaction remotely, thereby reducing downstream administrative burden and minimizing order errors. The result is shorter sales cycles plus a dramatic improvement in order value, customer retention and revenue.

"MBM is built on Microsoft .NET Compact Framework and makes extensive use of Web services, both to support selling activities via commercially available subscription-based web services, and as an integration conduit.

"MBM capitalizes on the resource packed Pocket PC platform via the comprehensive Microsoft .NET Compact Framework. The ease of writing Custom Controls in the .NET Framework enabled the creation of highly stylized rich user experiences that are optimized for the compact screen size of Pocket PC. MBM

takes full advantage of XML and Web Services to allow each customer to inject into the system their specific ways of doing business. This is a key aspect in creating a product (as opposed to a custom one-off application) that is readily configurable to handle the needs of widely varying customers. With the native support for XML in the .NET Compact Framework, this was a breeze.

## Issues Encountered and Resolved

"One of our big challenges was to target Tablet PCs, which run the full .NET Framework, along with Pocket PCs, which run the .NET Compact Framework. We use separate solution and project files in Visual Studio .NET for our Pocket PC and desktop versions. We conditionally compiled the code in shared source files where it made sense. We created separate code files (managed via Visual Studio) for the desktop and Pocket PC where that made sense.

"In some cases we found that the .NET Compact Framework was not a very clean subset of .NET Framework. For example, some classes were under different namespaces in the .NET Compact Framework, and we had to conditionally compile around that. We added and removed properties, methods and events and conditionally compiled to bring Pocket PC and [Microsoft®] Windows code in sync.

"The biggest issue we encountered with the .NET Compact Framework is the lack of OnPaint events in the controls. This forced us to write compound custom controls to get a distinctive look and feel for the PocketPC UI. Initially, that resulted in some significant page load performance issues, but we have been able to optimize the code and reduce the load time.

"We also found that the .NET Compact Framework is missing native support for Registry and COM, and that its support for XML is weaker than that in the .NET Framework. .NET Compact Framework is also missing some of the controls we wanted to use: HtmlViewerControl, InkX and the signature capturing control.

"We were able to work around all of these issues. Also, we manage different device form factors by isolating and abstracting a screen-size-dependant layer that provides a device-agnostic interface to the rest of the code.

## XForms

"The MBM suite includes a customer configurable forms package called MBM XForms. These are XML-enabled the .NET Compact Framework applets that are created via Visual Studio .NET extensions that we developed. MBM XForms go hand-in-hand with the customer-specific XML to form a system that is highly adaptable to individual customer needs.

## SQL Server CE 2.0

"The true relational database power of Microsoft SQL Server™ 2000 Windows CE Edition version 2.0 (SQL Server CE 2.0), combined with its merge replication capabilities, were essential ingredients in developing a powerful mobile application such as MBM that is fully functional even disconnected from the network. The ability to encrypt the entire SQL Server CE 2.0 database on the Pocket PC device is a key advantage in helping secure customer information locally stored on mobile devices that are susceptible to loss or theft. Furthermore, support for Blobs within this encrypted database helped us to even securely store documents such as Microsoft Word or Excel Files that could contain sensitive customer information as well.

"We initially encountered several performance issues with merge replication when large numbers of concurrent users were synchronizing. However, we were at SQL Server customer labs for a week in December and these issues were resolved to our satisfaction by the Merge Replication group.

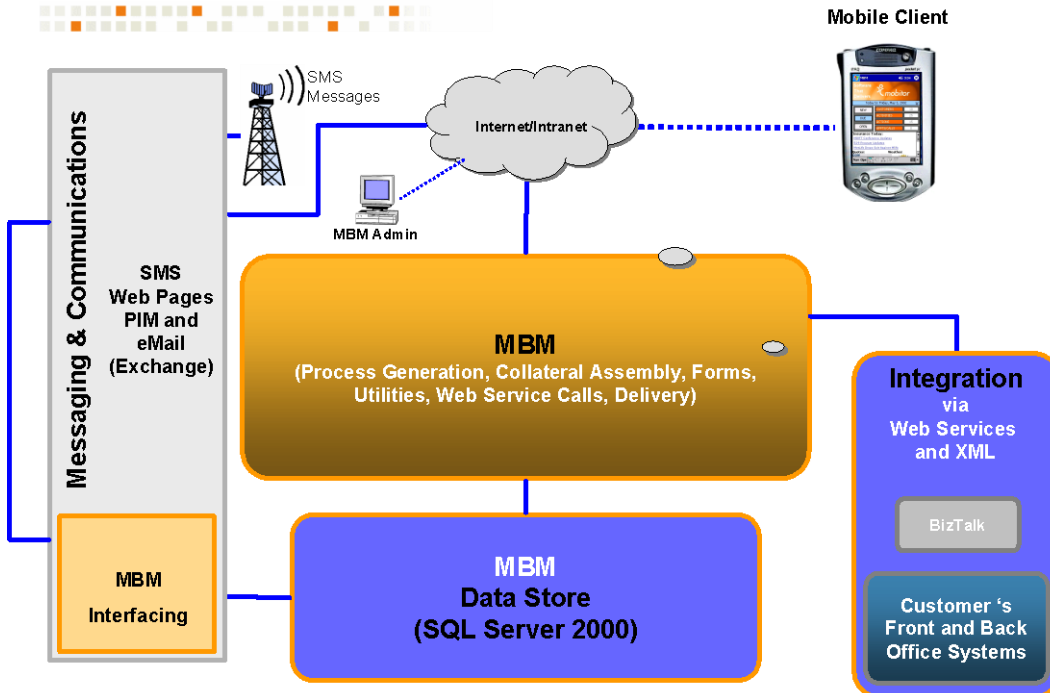
## Benefits

"For Mobitor, the selection of the Microsoft .NET Development Suite and SQL Server CE was extremely critical from both a business and a technical perspective. Since Mobitor is a product company, our software products are held to the highest standards over a long life. One thing we have learned throughout our years of experience in building software products is that our success and our reputation in many cases hinge on the underlying quality of not only the development tools but the tools company as well. Our partnership with Microsoft and our decision to build our Mobile solution using the .NET Compact Framework and SQL Server CE 2.0 has paid off handsomely. When we started our mobile product development over two years ago, we spent many man-weeks researching the development platform on which we were going to build our

*"In our opinion the Windows .NET Framework's native support of XML and XML Web Services did for software developers and integrators what USB and Plug-n-Play did for the PC peripheral hardware market."*

Dr. Lakshman Watawala  
Chief Technology Officer  
Mobitor

### X-TIER ARCHITECTURE – HIGH LEVEL



solution. This decision was extremely critical to the success of our new mobile product and of our company.

"We selected Microsoft, the .NET Compact Framework and SQL Server CE 2.0 because we wanted to partner with a company that was a market leader and had a long track record of delivering the highest quality tools and training. We also identified Microsoft as a company that truly had mobility computing as part of its core vision, and it offered a complete suite of development tools and environments that would allow us to develop our mobile vision without compromise. As a testimony to the success of our decision to use the .NET Compact Framework and SQL Server CE 2.0, we are often accused of demonstrating functionality in our sales engagements that is too futuristic. Potential customers find it hard to believe that we have included so much capability in a mobile device. Mobitor was able to build such a 'futuristic' mobile product because of the .NET Compact Framework, its native support of XML and Web Services, the flexibility to extend the Visual Studio .NET, and the comprehensive product management development environment.

## **Support of Multiple Mobile Devices**

"Since the .NET Framework and Compact Framework support a wide variety of devices like the Pocket PC, Pocket PC Phone Edition, Tablet PC and Desktop, they enabled us to write common base code that runs on many different devices. We just had to write a thin device-specific layer we call *decorators* to take advantage of device-specific features. It helped us to develop the application for one device and then quickly enable it for many different devices.

"The .NET Framework has also helped us create a branded look and feel by creating custom controls, forms, and wizards with minimal effort. Although this kind of development is possible without the .NET Framework, it was a tedious, cumbersome, and error prone process. With .NET, creating custom controls and libraries is straight forward and has enabled us to build a much more powerful product with a rich end-user experience.

## **Native Support of XML and Web Services**

"In our opinion the .NET Framework's native support of XML and XML Web Services did for software developers and integrators what USB and Plug-n-Play did for the PC peripheral hardware market. If you can remember the day before Plug-n-Play and USB you needed to be a hardware jock just to connect a scanner. The days of manually setting jumpers, interrupts, IRQs, and DMA channels are long gone in the hardware world. Now, with the native support of XML and Web Services in .NET, we will be able to dispense with proprietary customer-specific flat files, metadata, and proprietary metadata parsers (the software worlds equivalent of jumpers, interrupts, IRQs, and DMAs). Just as any consumer of a new Tablet PC or desktop computer would not tolerate anything but Plug-n-Play and USB support; corporate consumers of enterprise software products like MBM will soon be unwilling to tolerate any thing but the modern advantages and easy of integration and support of XML and Web Services.

## **Flexibility to Extend Visual Studio .NET**

"The flexibility of readily extending Visual Studio .NET with custom controls, custom properties, and custom designers is what enabled us to create an XML-enabled WinForms infrastructure we call MBM XForms. We created user interface elements (custom controls) that are optimized for the constrained real estate of the Pocket PC screen. Furthermore, we were able to develop sophisticated Custom Designers using XML DOM objects that allowed us to associate user interface components with underlying customer specific XML schema. The Custom Designers provided our end users with a point and click interface to generate extensive business forms that are delivered to the Pocket PC by MBM.

"The ease of creating Custom Controls, Properties, Property Attributes and Designers along with the ready inclusion of these assemblies into the Visual Studio .NET IDE made the development process highly productive. There is no other environment that provides independent developers this much flexibility, power, and easy of use.

## **Product Management**

"When you develop products, you have to develop a master plan for long term support of multiple releases in production by many customers. When developing a new product many newer development teams/companies don't plan for this and end up with huge support and customer satisfaction issues. Mobitor has been through the school of hard-knocks surrounding these kinds of long term support issues. We knew when we selected the Visual Studio .NET development environment that its integration with Visual SourceSafe® and its integrated HTML documentation

features would work well with our master plan for version control, release management, and documentation practices. Also, due to .NET technologies and our object orientated development practices, Mobitor is able to keep clean branches with product release code separate but in sync with minimal, highly isolated customer specific integration code. This soup-to-nuts Microsoft-centric development environment allows us to not only develop our solutions faster, it also allows us to maintain our existing code base and leverage it more effectively than ever before."

**The Microsoft .NET Compact Framework**, a subset of the .NET Framework, extends the benefits of developer productivity and XML Web services integration to resource-constrained devices, such as Pocket PC and other Windows CE-based systems. It provides a highly productive environment for development and integration with virtually any data source, ranging from local SQL Server CE databases to enterprise back ends. The Compact Framework's unified, hierarchical class library enables software developers to rapidly construct rich user interfaces, access native operating system methods, and interoperate with assets on the device, such as the infrared data port and other hardware extensions.

**The Microsoft .NET Framework** is an integral Windows component that supports building and running the next generation of applications and XML Web services.

For more information about the .NET Framework, go to:  
<http://msdn.microsoft.com/netframework/>

**Microsoft Visual Studio .NET** is the rapid application development (RAD) tool for building next-generation Web applications and XML Web services. Visual Studio .NET empowers developers to rapidly design broad-reach Web applications for any device and any platform. In addition, Visual Studio .NET is fully integrated with the .NET Framework, providing support for multiple programming languages and automatically handling many common programming tasks, freeing developers to rapidly create Web applications using their language of choice. Visual Studio .NET includes a single integrated development environment with RAD features for building Web applications and middle-tier business logic, and RAD XML designers for working with data.

For more information about Visual Studio .NET, go to:  
<http://www.visualstudio.net/>

## ***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:  
<http://www.microsoft.com/>

For more information about Mobitor's products and services, call (925)552-8230 or visit the Web site at:  
<http://www.mobitor.com/>

© 2003 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, MSDN, the .Net logo, Visual C#, Visual SourceSafe, the Visual Studio logo, Windows, and the Windows logo 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.