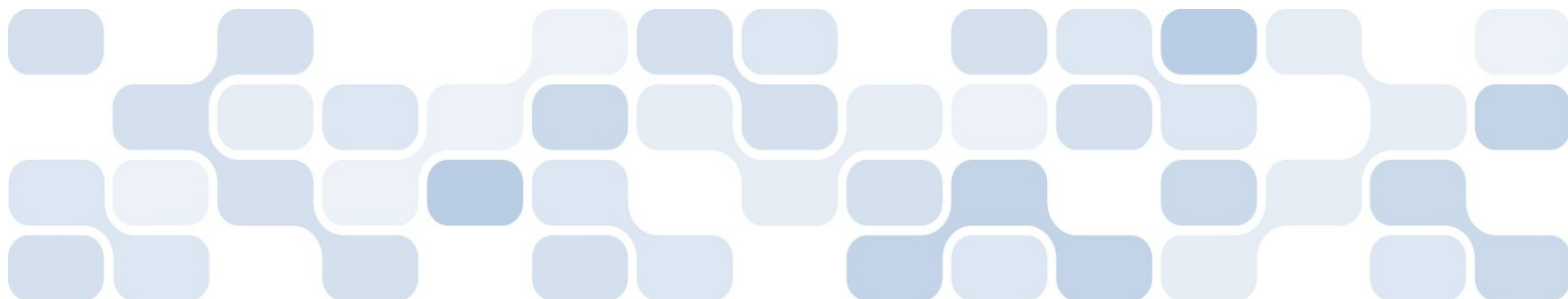




**Break Through Your Software
Development Challenges with
Microsoft[®] Visual Studio[®] 2008**
White Paper

November 2007

For the latest information, please see www.microsoft.com/vstudio



This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, Expression, Expression Blend, Outlook, Visual Basic, Visual C#, Visual SourceSafe, Visual Studio, and Windows Mobile are trademarks of the Microsoft group of companies

All other trademarks are property of their respective owners.

CONTENTS

Overview.....	1
Client Development.....	2
Web Development	3
Office Development.....	4
Mobile Development.....	6
Application Lifecycle Management	7
Summary	8

OVERVIEW

Microsoft® Visual Studio® 2008 enables organizations to take full advantage of the .NET Framework 3.5 and the 2007 Microsoft Office system to create client, Office, Web, and mobile applications¹—and through those applications deliver high-quality, rich user experiences that can help improve business processes and decision-making. Those capabilities are complemented by Microsoft Visual Studio Team System, a comprehensive application lifecycle management solution that provides tools, processes, and guidance to help development teams communicate and collaborate more effectively, ensure quality throughout the development process, and improve visibility into all aspects of the application lifecycle.

This document discusses key features of Visual Studio 2008 and Visual Studio Team System that raise the bar for client, Office, Web, and mobile application development, and that aid in application lifecycle management. It explains how Visual Studio 2008 can help development teams of all sizes to:

- Improve developer productivity
- Deliver breakthrough user experiences
- Effectively communicate and collaborate
- Improve software quality
- Enhance visibility into project status and quality

¹ Visual Studio 2008 Standard Edition supports client and Web development. Development of Office and mobile applications requires Visual Studio 2008 Professional Edition or Visual Studio Team System.

CLIENT DEVELOPMENT

Visual Studio 2008 enables developers to build rich client applications that run on the .NET Framework 3.5, including the ability to enhance existing client applications that are based on the Windows Forms classes in the .NET Framework 2.0. For example, a software-development group at a financial-services company might use an internally-developed desktop program for the analysis of currency trades. With Visual Studio 2008 and the .NET Framework 3.5, developers can deliver a new version of the application that provides richer analysis capabilities and the ability to execute modeled trades from within the application, instead of re-entering them into a separate program as they did in the past.

Some features of Visual Studio 2008 and the .NET Framework 3.5 that enable such enhancements include:

- **Windows Presentation Foundation (WPF)**, the graphical subsystem in the .NET Framework 3.5 for taking advantage of the capabilities of today's modern graphics hardware.
- **Windows Communication Foundation (WCF)**, the communication subsystem in the .NET Framework 3.5 for enabling applications or systems to communicate with one another easily.
- **Multitargeting**, which allows developers to use Visual Studio 2008 to target the .NET Framework versions 2.0, 3.0, or 3.5 from within a single tool-set.

The benefits of such an approach include:

- **Rapid Application Development.** With multitargeting, developers can use Visual Studio 2008 to develop the new version of an application or maintain the existing one without having to switch tools. WPF and WCF improve productivity by enabling developers to do more with less code, and interoperability between Windows Forms and WPF can help minimize the amount of new code that developers need to write.
- **Breakthrough User Experiences.** With WPF, developers can easily take advantage of the graphical features in today's powerful desktop hardware to give application users intuitive modeling and visualization capabilities. WCF makes it easy to integrate client applications with existing back-end processing systems, thereby eliminating the need for users to access a separate application for that purpose.
- **Effective Team Collaboration.** Graphic designers who use Microsoft Expression Blend™ to create application UIs can collaborate directly with developers who use Visual Studio 2008. Designers and developers can share design artifacts without changes to the UI jeopardizing event or business-logic code behind the UI—and vice-versa.

WEB DEVELOPMENT

Most companies that leverage Internet technology for public Web sites and/or internal business applications are looking for ways to accelerate the delivery of such solutions and enhance the end-user experience. Visual Studio 2008 and the .NET Framework 3.5 give developers a versatile tool-set for building richer, more responsive, and more interactive Web sites and applications, along with the ability to enhance development team output.

Some features of Visual Studio 2008 and the .NET Framework 3.5 that developers of Web-based solutions can take advantage of include:

- **Full support for Asynchronous JavaScript and XML (AJAX)**, a programming technique in which a Web browser exchanges small amounts of data with a Web server behind the scenes.
- **Language-Integrated Query (LINQ)**, which enables developers to use the Visual Basic[®] and Visual C#[®] programming languages to write syntax similar to that of Structured Query Language (SQL)—and to use the same approach for object collections, ADO.NET, XML data, and SQL Server databases.
- **Integration with Microsoft Expression[®] Web** through the use of Cascading Style Sheets and shared project files.

The benefits of such an approach include:

- **Breakthrough User Experiences.** Improved support for AJAX enables developers to deliver richer, more-responsive Web applications—and a more-interactive user experience—by eliminating the need to reload an entire Web page when a user makes a change.
- **Rapid Application Development.** Developers are more productive when using AJAX because they can take advantage of IntelliSense technology in Visual Studio 2008 to discover variables, objects, and their methods. They can debug JavaScript code by setting breakpoints inside Visual Studio—a capability that works with the most popular Web browsers. Developers using LINQ benefit from design-time assistance in Visual Studio 2008, such as statement completion, smart compile auto correction, and IntelliSense technology, enabling them to invest less time learning syntax for languages like T-SQL or XPATH and instead focus on how to use the data.
- **Effective Team Collaboration.** UI designers and software developers can work together in a friction-free environment, sharing projects, code, and designs. Designers can use Microsoft Expression Web to design a UI and then turn it over to developers with confidence that the UI design and subsequently developed business logic code will both remain intact. Effective team collaboration allows the company as a whole to improve its focus on the user experience throughout the development process.

OFFICE DEVELOPMENT

UIs for enterprise line-of-business applications can be difficult to learn, slow to respond to user input, and often require people to be connected to the corporate intranet. Those drawbacks can hamper user productivity and be a disincentive to enter or maintain data, thereby reducing the value of an enterprise application to the organization as a whole.

Visual Studio Tools for Office (included in Visual Studio 2008 Professional Edition and Visual Studio Team System) provide a powerful tool-set for building Office Business Applications (OBAs), which extend the data in line-of-business systems to end-users through familiar Microsoft Office programs. Through such an approach, enterprises can integrate the front-office and back-office, making the information in ERP, CRM, and other enterprise systems directly accessible from within everyday business processes.

For example, developers might build an OBA that provides access to data in a CRM system through the Microsoft® Office Outlook® messaging and collaboration client—the same tool that salespeople already use throughout the day to interact with customers and colleagues. With such a solution, salespeople can view and update CRM activities, contacts, and opportunities while working offline and, upon connecting to the corporate network, synchronize their changes with the main CRM system database. The solution could easily also display data from applications used for sales-reporting and customer-support, providing busy salespeople with a single view of each customer across multiple line-of-business systems.

Some features that aid developers in delivering OBAs include:

- **Fluent Ribbon Integration for Office 2007.** Developers can add buttons to the Office 2007 ribbon interface for common tasks.
- **Outlook Form Regions.** Developers can use form regions—a new feature in Office Outlook 2007 for customizing the UI—to let users view and edit data in line-of-business systems.
- **ClickOnce Deployment.** Development groups can post a new version of an application for people to install, simply by copying it to a directory on a Web server. Users install the application by clicking on a link in an e-mail message.

The benefits of such an approach include:

- **Rapid Application Development.** By building OBAs on top of Microsoft Office applications, developers can deliver new business solutions faster. Features such as Fluent Ribbon Integration and Outlook Form Regions made it easy to extend Office applications to add needed functionality instead of developing a separate application from scratch.
- **Breakthrough User Experiences.** By enabling employees to more easily access the information they need, OBAs can help employees

work more productively and contribute more effectively to business goals—even during times when connectivity with the corporate network is unavailable. Features such as Fluent Ribbon Integration further enhance the user experience, providing the same model for working with line-of-business data and other types of information.

MOBILE DEVELOPMENT

Mobile employees are often forced to work inefficiently, such as spending additional time in the office transferring paper-based records to a PC-based program. Companies can help those employees be more productive by using Visual Studio 2008 Professional Edition or Visual Studio Team System to easily extend the functionality of desktop applications to Windows Mobile®-based devices.

Some features that developers can take advantage of include:

- **.NET Compact Framework 3.5**, which lets developers reuse new technologies that they've used to build an application's desktop version, such as Windows Communication Foundation WCF and LINQ.
- **Unit testing support**, which enables developers to reuse some of the same unit tests they've developed for the desktop version of the application and write new ones where needed.
- **Enhanced mobile Windows Forms controls**, which make it easy to modify and optimize mobile applications' screens to support mobile devices' smaller displays.

The benefits of such an approach include:

- **Rapid Application Development.** The effort required to extend the functionality of an existing desktop application to mobile users is significantly reduced because developers can reuse a large part of the architecture, code, and unit tests from the desktop application, with most of the work going into developing a new UI for the smaller screen.
- **Breakthrough User Experiences.** A Windows Mobile-powered solution puts the capabilities and information that mobile workers need at their fingertips when it is most useful—while they are on the move. New mobile Windows Forms controls enable developers to optimize the mobile application's UI for particular job roles to enhance worker productivity.

APPLICATION LIFECYCLE MANAGEMENT

Most development teams use a number of standalone tools to manage the application lifecycle. Specifications and requirements are stored on file shares or SharePoint sites; tasks assigned to developers are managed using e-mail; bugs are tracked with spreadsheets; and source code resides in one or more version control systems. With important information in so many different places, it's hard for team members to effectively work together.

Comprehensive application lifecycle management capabilities provided by Visual Studio Team System make this scattershot approach obsolete. At the core of Team System is Visual Studio Team System 2008 Team Foundation Server, which provides a unified repository for all project data and artifacts, along with the tools to define, enforce, and automate desired processes.

Some features of Team Foundation Server that development teams can take advantage of include:

- **Work Item Tracking.** All work items are managed centrally, with status updated automatically as part of everyday workflows.
- **Team Build.** Continuous-integration builds and automated tests are done whenever a check-in occurs. Immediate e-mail notifications inform everyone if a problem exists.
- **Reporting.** Prebuilt reports—such as Quality Indicators and Remaining Work—put the information needed to assess progress and software quality at peoples' fingertips.

The benefits of such an approach include:

- **Enhanced Collaboration.** All members of the development team—including architects, developers, testers, database professionals, and project managers—can work together more effectively. Version-control policies enforce desired processes, such as requiring all checked-in code to be associated with a work item. Testers are notified when new code is checked in and which features or bugs have been marked as complete, and they can immediately begin testing those items.
- **Improved Software Quality.** Continuous-integration builds and automated testing enable broken builds and integration issues to be identified immediately, helping to ensure that valuable testing time is not lost. Prior to checking in code, developers use the integrated code analysis and profiling tools in Team Suite to help detect potential security and performance problems.
- **Better Visibility.** With all project data in one place and gathered as part of normal workflows, everyone has deep, real-time visibility into project status, enabling them to spend less time finding and manipulating information and more time acting on it. A Web-based portal provides access to that information for remote workers and project stakeholders outside of the development team.

SUMMARY

Visual Studio 2008 and Visual Studio Team System raise the bar for client, Office, Web, and mobile application development, as well as for application lifecycle management. Through their adoption, development teams of all sizes can improve developer productivity, deliver breakthrough user experiences, communicate and collaborate more effectively, improve software quality, and enhance visibility into all aspects of the application lifecycle.

For the latest information about Visual Studio 2008, see the Visual Studio Web site at www.microsoft.com/vstudio.