



Microsoft®

Visual Studio® 2008

An Overview of Microsoft® Visual Studio® 2008

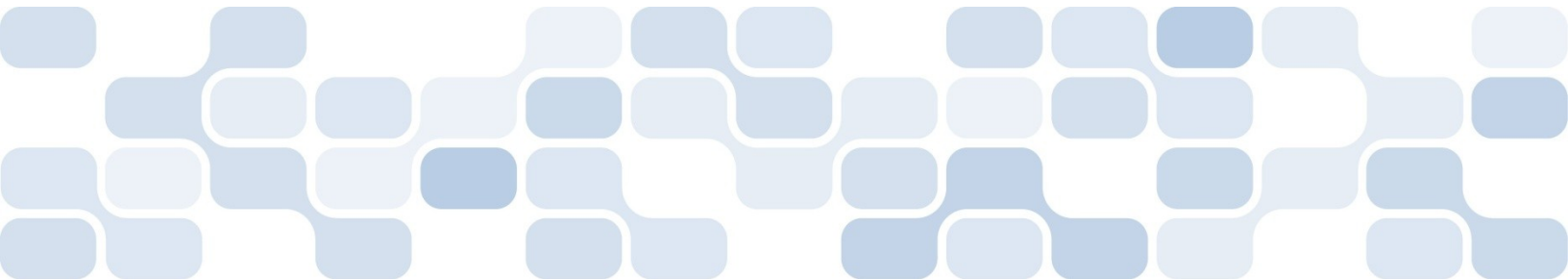
White Paper

July 2007

Tony Goodhew

For the latest information, see

<http://msdn.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, Excel, Expression, InfoPath, IntelliSense, Outlook, PowerPoint, SharePoint, SQL Server, Visio, Visual C++, Visual Studio, the Visual Studio logo, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

Contents

Overview 4

Develop Smart Client Applications 6

Create Microsoft Office Applications..... 8

Build Windows Vista Applications 9

Handle Data More Productively..... 10

Enable New Web Experiences 11

Better Developer Experience 12

Improve Application Life-Cycle Management (ALM)..... 13

Summary 14

Overview

Microsoft® Visual Studio® 2008 delivers on Microsoft's vision of smart client applications by enabling developers to rapidly create connected applications that deliver the highest quality, rich user experiences. With Visual Studio 2008, organizations will find it easier than ever before to capture and analyze information, which means they can make effective business decisions. Visual Studio 2008 enables organizations of every size to rapidly create more secure, manageable, and reliable applications that take advantage of Windows Vista™ and the 2007 Office system.

Visual Studio 2008 delivers key advances for developers in three primary pillars:

- Improve Developer Productivity
- Manage the Application Life Cycle
- Employ the Latest Technologies

This document will discuss the different customer experiences that deliver on these three pillars through seven different technology areas:

- Develop Smart Client Applications
 - Visual Studio 2008 delivers new and easy ways for developers to build smart clients by providing a comprehensive set of tools and classes that simplify integrating smart clients with new or existing web applications, and by handling local caching of data for disconnected scenarios.
- Create Microsoft Office applications
 - Visual Studio Tools for Office (VSTO) is now fully integrated into Visual Studio 2008 Professional Edition. Visual Studio enables developers to customize various Office applications, such as Outlook® and PowerPoint®, to improve user productivity and simplify deployment.
- Build Windows Vista Applications
 - Developers will be able to easily leverage new platform technologies, and deliver more compelling applications to their customers, by effortlessly incorporating new Windows Presentation Foundation features into both existing Windows Forms applications and new applications.
- Handle Data More Productively
 - The introduction of Language Integrated Query (LINQ), and various other data access improvements, now enable developers to deal with data using a consistent programmatic approach, perform data access with new data design surfaces, and use built-in classes for the occasionally connected design pattern.
- An Improved Developer Experience Overall
 - Visual Studio 2008 delivers a better developer experience overall through the combination of significant product quality improvements, changes to the way the most popular design surfaces return errors to the user, and simplifying the user's ability to adopt the toolset and framework separately.

- Enable New Web Experiences
 - Beyond the secure, reliable and extensible infrastructure of IIS, developers can easily create efficient, interactive Web applications. The seamless integration and familiar programming model of ASP.NET AJAX enables more efficient client-side execution to deliver end-users a more responsive Web interface.
- Improves Application Life-cycle Management (ALM)
 - ALM provides great support, not only for managing the entire software development life cycle but also for the critical interaction with the final end-users and IT stakeholders of an enterprise application.

Develop Smart Client Applications

Visual Studio 2008 provides developers with new ways to build smart clients that enhance the functionality of the application, in addition to improving developer productivity. Many customers face the challenge of integrating their smart client applications with existing and new Web based applications through logon and personalization services. Visual Studio provides a new set of tools and classes that simplify this integration and enable developers to simply handle the local caching of data for disconnected scenarios.

Smart client development areas of focus include:

- Integrated UI designer experience
 - By supporting common formats (such as XAML) and giving designers more direct control over the layout, controls, and data binding of the application user interface (UI), Visual Studio makes integrating designers into the development process easier. Designers can use familiar tools like the Microsoft® Expression® Suite to create UIs and produce files that developers can work with directly in Visual Studio. Visual Studio supports a fully collaborative workflow that enables designers and developers to hand work back and forth directly and work in parallel. Designers and developers will also be able to build libraries of common UI designs, formats, and elements that can be easily managed and reused.
 - By taking advantage of the 2007 Office system as a development platform, designers can leverage the look and feel of both Office and SharePoint to create familiar, intuitive UI. Developers can then use Visual Studio Tools for Office to create the UI in their solutions quickly and easily.
- Improved ClickOnce application deployment
 - ClickOnce deployment in Visual Studio 2005 provided Web like deployment for smart clients. Users could go to a single URL and click a link to have the smart client application installed on their machine. IT Professionals could deploy new versions of the application by simply copying the new application to the URL. In Visual Studio 2008, ClickOnce now supports Firefox® as a browser and provides location independent signing and customer branding. In addition, ClickOnce now supports deployment through authenticated proxy servers and enables developers to distribute their application to multiple end-user companies from a central location.
- Office 2007 UI style support for native C++ applications
 - Visual Studio 2008 also provides developers with support for building applications that use the Microsoft Office 2007 UI style, including the Ribbon Bar, Ribbon Status Bar, and Mini-toolbar.
- Client Application-Level Services
 - Enable client application developers to use the same user profile and login services as their Web applications. This enables customers to use one method of backend storage for user personalization and authentication, regardless of the application type.

- Occasionally connected data access
 - The Microsoft Synchronization Services for ADO.NET provide an application programming interface (API) to synchronize data between data services and a local store. The Synchronization Services API is modeled after the ADO.NET data access APIs and gives you an intuitive way to synchronize data. It makes building applications for occasionally connected environments a logical extension of building applications where you can depend on a consistent network connection.
- Integrated support for Microsoft® SQL Server™ 2005 Compact Edition
 - Microsoft SQL Server 2005 Compact Edition is the next version of SQL Server Mobile, adding the desktop platform. SQL Server Compact extends the SQL Server Mobile technology by offering a low maintenance, compact, embedded database for single-user client applications, which is available for all Windows platforms including Tablet PCs, Pocket PCs, smart phones and desktops. Just as with SQL Server Mobile, SQL Server Compact is a free, easy-to-use, lightweight, and embeddable version of SQL Server 2005 for developing desktop and mobile applications.
- Leverage SOAs and WCF in mobile applications
 - Developers will be able to build a mobile device client application that works in a partially connected environment. The application will be able to send and receive data from a server, even if the device disconnects or roams. By providing the logic to solve the addressability and storage issues, a developer can focus on the mobile application functionality and not worry about the different identities, connection methods, or storage models required.

Create Microsoft Office Applications

Visual Studio Tools for Office (VSTO) is now fully integrated into Visual Studio 2008 Professional Edition. Developers can now easily target the more than 500 million users of Microsoft Office while using the same managed code skills that they've developed for writing Microsoft Windows applications or ASP.NET applications. Visual Studio enables developers to customize Word, Excel®, PowerPoint, Outlook, Visio®, InfoPath®, and Project to improve user productivity and take advantage of the many improvements in the 2007 Microsoft Office system. Visual Studio 2008 enables developers to perform tasks such as:

- Target the breadth of the new 2007 Microsoft Office system
 - Developers can use the tools in Visual Studio to create both application level and document level managed code customizations behind 2007 Office system applications quickly and easily. Visual Studio's built-in visual designers for key 2007 Office system UI features provide developers with a RAD development experience and enables them to deliver applications with a high-quality Office-based UI.
- Build applications that target Microsoft Office SharePoint Server
 - Visual Studio simplifies the task of building and debugging SharePoint Workflow projects, and also enables developers to build applications that provide easy access to back-end data stores and to data from enterprise systems such as SAP, CRM and ERP. Web Parts built for SharePoint are easily reusable in other applications.
- Develop Microsoft Office-based UI workflow solutions
 - Easily incorporate key Microsoft Office UI elements, including the Fluent Ribbon and Custom Task Panes, and integrate SharePoint Workflow Services into Visual Studio solutions. Developers can create applications that use Microsoft Office to manage process navigation and enterprise data surfacing, combined with SharePoint workflow services to manage collaboration requirements.
- Deploy applications securely and easily
 - Developers now have an easy to use and version resilient security model for their applications that will be compatible with future versions of Visual Studio and Office. With full support for ClickOnce deployment of all Office customizations and applications, developers and administrators now have the right tools and framework for easy deployment and maintenance of their Office solutions. This greatly simplifies the process of creating and deploying new solutions based on the 2007 Office system.

Build Windows Vista Applications

With Visual Studio 2008, developers will be easily able to leverage new platform technologies and deliver more compelling applications to their customers. Visual Studio enables customers to effortlessly incorporate new Windows Presentation Foundation features into both existing Windows Forms applications and new applications.

Developers can also move their applications to the new Windows Vista *look and feel* easily with enhancements to MFC and Visual C++®.

Visual Studio offers developers key improvements such as:

- Enable the construction of *rich experience* Windows applications
 - Visual Studio provides tools that enable developers who are early adopters of Windows Presentation Foundation to build *rich experience* applications quickly and easily. These tools include a designer and XAML editor, project templates, debugging support, deployment support, etc.
- Easily add the Windows Vista *look and feel* to native C++ applications
 - Developers can use Visual Studio to build applications that exhibit the Windows Vista *look and feel* and take advantage of the more than 8,000 new native APIs available in Windows Vista. A number of the Windows Vista *look and feel* features are available simply by recompiling an MFC application. Deeper integration that requires more coding or design work on the part of the developer is also simplified with Visual Studio's integrated support for the Windows Vista native APIs.
- Office 2007 UI style support for native C++ applications
 - Visual Studio 2008 also provides developers with support for building applications that make use of the Microsoft Office 2007 UI style, including the Ribbon Bar, Ribbon Status Bar and Mini-toolbar.
- Improved interoperability between native and managed code
 - Visual Studio makes it easier to build applications that leverage both native and managed code, and also delivers improved interoperability performance. C++ developers have access to a new marshalling library that simplifies data transfer across the native-managed boundary STL/CLR for extending the Standard Template Library (STL) into managed code.

Handle Data More Productively

Visual Studio 2008 significantly improves the way developers handle data. Traditionally, developers have manipulated data differently depending on where the data resides and how the user connects to it. With the introduction of Language Integrated Query (LINQ) and various other data access improvements, developers can now manage data using a consistent programmatic approach and perform data access with new data design surfaces. ADO.NET integrates with LINQ and supports an occasionally connected design pattern to simplify the development tasks for those application types.

These new capabilities include:

- Programming the model and design experience around language/data access unification
 - LINQ aims to reduce complexity for developers and help boost their productivity through a set of extensions to the C# and Visual Basic programming languages as well as the Microsoft .NET Framework, which provide integrated querying for objects, databases, and XML data. Using LINQ, developers will be able to write queries natively in C# or Visual Basic, without having to use specialized languages, such as SQL and XPath.
- Simplify the construction of data driven Web sites with LINQ to SQL
 - With deep support for incorporating .NET language integrated query (LINQ to SQL) into ASP.NET Web applications, Visual Studio makes the creation of data-driven Web sites more productive, more efficient and more fun. Developers can use familiar programming semantics to access all data sources in a unified and uniform manner.

Enable New Web Experiences

Microsoft offers organizations a robust, end-to-end platform for building, hosting, and exposing applications over the Web. Beyond the secure, reliable, and extensible infrastructure of IIS, developers can easily create Web applications with more interactive, responsive, and efficient client-side execution by using the seamless integration and familiar programming model of ASP.NET AJAX along with Internet Explorer browser extensions and enhancements.

Teams targeting the Web can collaborate more effectively and obtain faster results by integrating the advanced designers and editors of the new Expression tools into their development workflow, and by incorporating the broad functionality of Windows Live services into their solutions.

Visual Studio enables the creation of new Web experiences by empowering Web developers and simplifying Web development. Visual Studio 2008 gives developers the tools they need to build the next generation of Web experiences more quickly, efficiently, and easily than ever before:

- Enabling Web developers to program “AJAX-style” interactive Web user interfaces
 - Visual Studio provides developers with all the tools and framework support required to create compelling, expressive, AJAX-enabled Web applications. Developers will be able to take advantage of these rich, client-side and server-side, frameworks to easily build client-centric Web applications that integrate with any back-end data provider, run within any modern browser, and have complete access to ASP.NET application services and the Microsoft platform.
- Designing and implementing a Web service
 - As the concept of a Web service evolves, Visual Studio will enable developers to configure service endpoints, using the same tools and code, regardless of what wire protocol (HTTP, TCP/IP) is used, to transmit messages and test the service without code. Developers and partners will be able to extend the underlying protocols to handle any definition of a Web service.
- Easily consume Windows Communication Foundation (WCF) services
 - Developers can use RAD tools to quickly and easily create client connections and proxies to existing services, and test them without needing to write code. In addition, developers can use the same techniques and tools for consuming WCF services, no matter where they are located.
- Service Implementation integrated with workflow design
 - Visual Studio enables the developer to orchestrate behavior across services with Windows Workflow Foundation (WF) to visualize, create, edit, and debug workflow tasks and dependencies.

Better Developer Experience

Visual Studio 2008 builds on the productivity and developer experience improvements delivered in Visual Studio 2005. The development process used to create Visual Studio 2008 enabled the individual feature teams to focus on the final quality of the feature throughout the entire development process. This approach raises the overall product quality significantly. In addition, the overall developer experience with the Visual Studio 2008 is improved through the ability of Visual Studio to build and target all the platforms that developers have been using in their projects over the last few years. This enables development teams to adopt Visual Studio 2008 without a corresponding IT cost in deploying new framework components.

Developer experience includes:

- Build using Visual Studio 2008, target different .NET Framework platforms
 - Developers have traditionally required tools that are bound to the runtime platform that they are building against. With Visual Studio they can now use one toolset and target the desired platform.
- Improvements to the Windows Forms designer
 - Windows Forms continue to provide developers with the features and functionality required to build compelling Line-Of-Business applications. With Visual Studio 2008 the design-time experience for Windows Forms UI and component developers has continued to improve. Improvements in the performance of the designer enable developers to be more productive when building a Windows Forms application. In addition, the simplification of a design-time error list provides developers with more usable feedback, and an ability to *ignore and continue*, thus easing the overall development task.
- Product Quality Improvements
 - Continued focus on improving product quality, and fundamental changes in the way Microsoft builds developers tools, enables Microsoft to deliver a high quality tool with significant servicing investments over previous versions.
- Leverage existing UI investments
 - Visual Studio not only provides new designers for building the next generation of applications, but also makes it easy for developers to extend existing applications to deliver high-quality user experiences. Enhancements to the existing Windows Forms designer enable task-oriented designs for creating applications that leverage Windows Forms, .NET Framework 3.0, and XAML in one application. This includes providing a design-time experience for using this Windows Presentation Foundation content in an existing Windows Forms application by enabling the Windows Forms visual designer to place and visually layout the new content in relation to the other controls on the form. This ability also enables Windows Forms content to be placed into a new application built on the Windows Presentation Foundation.

Improve Application Life-Cycle Management (ALM)

In Visual Studio 2008, Microsoft is continuing to invest in the market-leading Visual Studio Team System technology. Visual Studio 2008 provides great support for not only managing the entire software development life cycle but also the critical interaction with the final users and managers of an enterprise application. In addition, it is designed to expand the collaborative benefits of the Visual Studio Team System to more roles on the project team.

By addressing the needs of a wide range of customers, from the smallest independent developers to the largest enterprise customers, Visual Studio 2008 will make delivering quality solutions easy regardless of the size of the project or team.

With this release the new capabilities include:

- Integrate the database professional into the software life cycle
 - Creating off-line database representations and database projects brings the database professional into the development life cycle. Visual Studio 2008 provides a full suite of tools source control, testing and test data generation, rename refactoring, and a deployment solution that includes visual diff/merge and deploy script generation.
- Extend Visual Studio's Unit Testing capabilities
 - Unit testing is easily one of the most sought after abilities incorporated into Visual Studio. This release extends its capabilities, improving its performance and broadening its reach: unit tests now run faster whether they're executed from the IDE or from the command-line; test inheritance enables users to reuse inherited methods; usability improvements enable users to execute a test directly from its definition; unit tests are now available to all Visual Studio Professional Edition users and can also be used to test mobile applications.
- Load testing for the enterprise
 - Visual Studio 2008 improves on existing load testing capabilities by simplifying the load testing interface and providing a multiple machine graph view that brings together the test results, performance, and health of all the machines under test. Additional improvements enable better management of the test results, rich load modeling, and the integration of results from non-PerfMon sources.
- Enable performance tuning and diagnostics of enterprise applications through testing
 - A new area of support in Visual Studio is the ability to drive system performance tuning and diagnostics through the Visual Studio test tools. This enables developers to run profiling during tests, so they can run load and test procedures against a system, see how it behaves, and use integrated tools to profile, debug, and tune. Also included is performance base-lining, so that users can save a baseline profile and then, if the performance degrades, compare up-to-date traces to identify the source of the regression.

Summary

In summary these seven technology areas:

- Develop Smart Client Applications
- Create Microsoft Office Applications
- Build Windows Vista Applications
- Handle Data More Productively
- Enable New Web Experiences
- Better Developer Experience
- Improve Application Life-cycle Management

In summary, this white paper provides a view of the features that support Microsoft's primary investment pillars. These three pillars in Visual Studio 2008 allow developers to:

- ***Improve Developer Productivity***

In Visual Studio 2008, developer productivity doesn't end with the code editor and wizards. By extending this concept to application architectures and the underlying platform, Visual Studio 2008 delivers not only a productive development tool, but also enables developers to tackle new business problems while decreasing the total cost of solution construction. In Visual Studio 2008 developers, designers, and database professionals will see new tools and frameworks become available to simplify their tasks.

- ***Manage the Application Life Cycle***

Visual Studio 2008 enhances the end-to-end value of Visual Studio Team System by increasing its role-based coverage and delivering enhanced traceability throughout the software development life cycle. With deep integration across roles in the software life cycle and the Team Foundation Server, Team System enables customers to amplify the impact of their teams and improve software quality.

- ***Employ the Latest Technologies***

As users look for new ways to comprehend and retain information, developers must still grapple with basic desktop and application security. Visual Studio, Windows Vista, and the 2007 Office system, enable developers to deliver a safe, robust and compelling user experience in any type of application.

These fundamental advances enable customers to rapidly create connected applications that deliver the highest quality, rich user experiences regardless of project complexity or organization size.

For the latest information about Visual Studio 2008, see the Visual Studio Web site (<http://msdn.microsoft.com/vstudio>).