



Accelerate J2EE™ Development Using Visual Studio® Resources

*Visual MainWin® for the J2EE platform:
An Efficient Solution for IT Developers*

December 2004
www.mainsoft.com

Accelerate J2EE Development Using Visual Studio Resources

*Visual MainWin for the J2EE platform:
An Efficient Solution for IT Developers*

Executive Summary

This white paper describes the business need in large IT organizations to use existing Visual Studio resources and skills to create enterprise applications that can be deployed on the Java™ 2 Enterprise Edition platform (J2EE).

Mainssoft's Visual MainWin for the J2EE platform is a cross-platform development solution that enables software developers skilled in C# and Visual Basic® .NET to create J2EE Web applications and Web services quickly, and at a fraction of the cost of transforming a Microsoft-centric development group into a highly-skilled J2EE development team.

Mainssoft Corporation, the cross-platform development company, enables businesses to develop business-critical applications with Visual Studio and to deploy them on UNIX®, Linux®, and J2EE. Since 1993, many of the world's largest independent software vendors (ISVs) have used Mainssoft's cross-platform products to deploy more than \$1 billion worth of software annually on multiple operating systems.

The Business Need

Today's IT professionals are under tremendous pressure to deliver high-quality enterprise applications with significant business value, while minimizing project costs and meeting aggressive production schedules. According to the Standish Group CHAOS report, few IT departments meet their productivity objectives¹. Less than a third of all software projects are completed on time and on budget. Overruns on challenged projects average 45 percent of the original budget and 63 percent of the original schedule, and nearly a quarter of all software projects fail or are cancelled.

The emergence of two conflicting software standards for developing and deploying the next generation of enterprise applications — Microsoft's .NET and J2EE from IBM, Sun, and BEA — tends to compound development risks and productivity issues.

While most CIOs favor the J2EE platform for deployment of business applications for its flexibility and scalability, the development challenges are significant, and the .NET/J2EE integration costs are steep.

The Gartner Group projects that the growing demand for large-scale J2EE projects far outpaces the availability and skills of J2EE developers, and the gap will continue to grow for the next three to four years. According to Gartner, this is because Java development needs to be further “dumbed down” before it can be widely accepted by mainstream enterprise developers, and the “growing complexity and ease-of-use concerns” remains the main weakness of the J2EE platform.²

Further confounding the adoption of J2EE is the lack of Java developers working in the enterprise. For many years, Visual Basic has been the mainstream enterprise development language, with more than three million enterprise developers³, and to date, none of the J2EE development tools support Visual Basic, thus compounding the learning curve for a significant part of the enterprise developer market.

¹ The Standish Group CHAOS report, 2001

² .NET versus Java, competition or coexistence, Mark Driver, Gartner ITXPO, Oct 2003

³ .NET versus Java, competition or coexistence, Mark Driver, Gartner ITXPO, Oct 2003

For large IT organizations, transforming a Microsoft-centric development organization into an effective J2EE development staff is an expensive and risky venture. It requires changing the development team's methodology, workflow, and development culture and may create internal tensions between business imperatives and technical aptitudes.

These drawbacks seem to suggest that hiring a second in-house development team or outsourcing J2EE development are more realistic alternatives to retraining and retooling existing staff. However, increased cost and time-to-market considerations may make these options less attractive.

In addition, large IT organizations with both Visual Studio® .NET and Java development teams have to manage conflicting software standards for developing enterprise applications and Web services and growing .NET/J2EE integration costs.

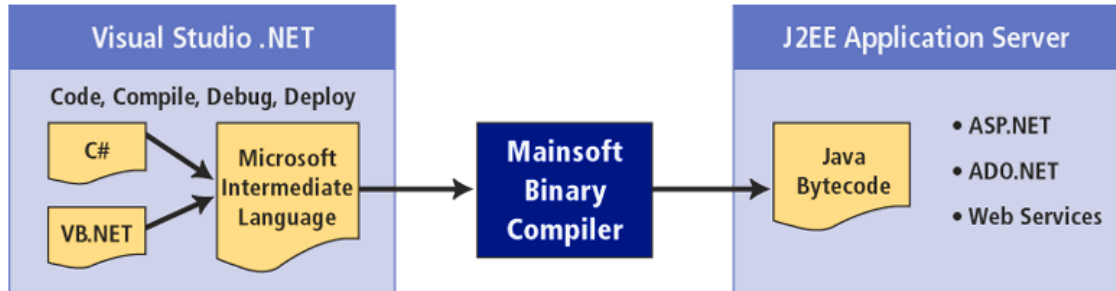
The disparity between J2EE's ability to meet business demands, and the productivity crisis of developing and integrating J2EE applications, has created the need for a development tool that enables an organization to leverage existing Microsoft skills and develop enterprise applications, which can be natively deployed on any standard J2EE application server.

Visual MainWin brings the Visual Basic .NET and C# programming languages to the J2EE platform. Until now, the Java platform has been a mono-language platform accessible exclusively to highly skilled Java developers. By opening the Java platform to an additional 2.5 million .NET developers, Visual MainWin enables IT organizations to leverage existing Microsoft-centric resources and skills to create composite applications for any standard J2EE application server using either Visual Basic .NET or C#.

Product Overview

Visual MainWin facilitates development of J2EE Web applications and Web services, enabling .NET developers to develop, debug, and deploy their code for J2EE directly within Visual Studio .NET.

Visual MainWin introduces a patent-pending technology enabling the binary compilation of Microsoft Intermediate Language (MSIL) into standard Java bytecode (JBC). Developers can write programs in their favorite language, such as C# and Visual Basic .NET and then compile their source code directly into standard Java bytecode.



Binary compilation from MS Intermediate Language to Java bytecode

Mainsoft designed and developed Visual MainWin to deliver the productivity advantages of the Visual Studio IDE, and provide native deployment of Web applications and Web services on J2EE. The product adheres to the following principles:

- **Preserve the Visual Studio developer experience** – Visual MainWin works seamlessly within the Visual Studio .NET development system and fully preserves the Visual Studio developer experience. Visual Basic .NET and C# developers can code, compile, run and debug J2EE applications using Visual Studio .NET, without having to learn to use a new development tool.
- **Support leading .NET development tools on J2EE** - Visual MainWin rehosts ASP.NET elements from Infragistics™ NetAdvantage™ 2004 Volume 2 presentation layer development toolset on the J2EE platform, offering .NET developers the unprecedented ability to quickly develop commercial-quality J2EE Web applications and Web services using NetAdvantage's easy-to-use designers and style presets. All the user interface richness and drag-and-drop productivity are maintained.
- **Comply with Java standards** – Visual MainWin generates pure Java bytecode, which can be executed on any Java Virtual Machine. Because the output is fully compliant with J2EE standards, Visual MainWin applications can be installed, deployed and managed from the application server administration console as a standard J2EE application.
- **Be open to other Java development components** – Java classes and components such as Enterprise JavaBeans™ (EJBs™), which have been developed with a Java development tool, can be imported into Visual Studio projects and used in applications written in Visual Basic .NET or C#. This powerful capability enables Visual Studio developers to leverage Java components for new applications.

- **Simplify .NET to J2EE porting** - Visual MainWin supports robust porting capabilities, including support for single-source code development for the .NET and J2EE platforms and a Visual Studio-integrated project conversion tool, which guides enterprises and independent software vendors (ISVs) in quickly porting existing .NET code to the J2EE platform.

Product Features

Complete Visual Studio .NET Developer Experience

Visual MainWin preserves the complete Visual Studio .NET developer experience when developing for the J2EE platform. The following Visual MainWin features are available as part of the integration with Visual Studio .NET:

- **C# Language Support**

Visual MainWin supports the C# language, as defined in [ISO specification 23270](#). To support the execution of the C# program in the Java environment, Visual MainWin provides an implementation of the .NET Common Language Infrastructure (CLI), as defined by ECMA-335. Microsoft, Intel, and Hewlett-Packard co-submitted specifications for C# and CLI to ECMA in August 2000.

- **Visual Basic .NET Language Support**

Visual MainWin supports the Visual Basic .NET language, a high-level programming language for the .NET Framework, as specified by Microsoft.

- **Enhanced Editing Capabilities**

Visual Studio .NET offers developers a large collection of useful editing features, such as syntax coloring, auto completion, and XML comments. Visual MainWin extends the powerful editing features of Visual Studio .NET to projects that output Java bytecode and to external Java code.

- **Support for Rapid Application Development (RAD)**

Visual MainWin fully supports the RAD design-time experience of Visual Studio .NET. RAD features such as the Web Forms Designer and drag-and-drop of Web and HTML controls, dramatically accelerating application development by automatically generating Visual Basic .NET or C# source code. Visual MainWin can then compile this code directly into Java bytecode without modifications.

- **Integrated Java Debugger**

Visual MainWin extends the Visual Studio .NET integrated debugger to support debugging of J2EE applications. Debugging can be done

at source level in Visual Basic .NET and C# while the application is running on a J2EE application server. Advanced features include a call stack display and a powerful expression evaluator.

- **Dynamic Help**

Visual MainWin enables direct access to all the J2SE™ SDK classes from Visual Basic .NET and C#. To increase developer productivity, the original help files of the Java 2 Standard Edition classes have been integrated into the Visual Studio dynamic help system and are displayed in native MSDN® format. This feature provides a quick and convenient way to get familiar with the J2SE SDK.

- **Project and Item Templates**

Visual MainWin offers class item templates along with project templates such as ASP.NET Web Service and ASP.NET Web Application. These templates provide the skeletons for the basic components that can be created and compiled for Java.

- **Visual Studio-integrated project conversion tool**

Visual MainWin provides a project conversion, wizard-based tool integrated into Visual Studio .NET, which guides you in quickly porting existing .NET projects to the J2EE platform.

- **ADO.NET Runtime**

Visual MainWin provides a J2EE-based implementation of ADO.NET, allowing users to program directly with the standard Microsoft API and use Visual Studio .NET database tools to develop J2EE applications with data access. Since database access is a central part of today's critical business applications, a J2EE implementation of ADO.NET greatly simplifies the development of database code.

- **ASP.NET Web Forms Runtime**

Visual MainWin provides a J2EE-based implementation of ASP.NET that fully supports ASP.NET programming and the design of Web Forms using drag-and-drop or double-click techniques.

- **ASP.NET Web Services Runtime**

Visual MainWin provides a Java-based implementation of ASP.NET Web services, allowing developers to quickly develop Web services in Visual Basic .NET or C# and deploy them on J2EE application servers.

Full Compliance with Java Standards

Visual MainWin fully supports Java standards, including deployment on the leading J2EE application servers: IBM WebSphere®, BEA WebLogic®, and JBoss®. Applications developed with Visual MainWin

can be deployed and managed as any other J2EE applications. Visual MainWin complies with the following Java standards:

- **Java Compliant Output**

Java bytecode generated by Visual MainWin is compatible with the Java 1.4 specification. As a result, applications created by Visual MainWin can run on any compliant Java Virtual Machine.

- **ADO.NET Implementation Based on Java Data Base Connectivity (JDBC)**

Visual MainWin provides an implementation of ADO.NET on top of the JDBC API, enabling applications to connect transparently to any JDBC compliant database from any J2EE application server.

- **ASP.NET Web Pages Implementation Based on Servlets**

Visual MainWin provides an implementation of the ASP.NET engine on top of Java servlets. As a result, ASP.NET Web applications can be deployed on any compliant servlet container.

Open J2EE Development Environment

Visual MainWin boosts development productivity of J2EE development by integrating seamlessly with other J2EE development tools:

- **Access to any Third-Party Java Class**

Any Java class that is part of a Java library can be accessed from Visual Basic .NET or C#. The class is mapped to a .NET class, and the class methods, properties, and types are exposed with the same names in the .NET class.

- **Consumption of Enterprise JavaBeans (EJBs)**

EJB components running on a J2EE application server can be accessed from Visual MainWin applications. Visual MainWin will automatically generate a wrapper class that encapsulates the EJB and display it as a native .NET object. The wrapper class also handles the Java Naming and Directory Interface (JNDI®) lookup and the creation of an EJB interface. Because developers can directly access EJB business methods within Visual Studio, they do not need to deal with component lifecycle and method invocation issues.

- **Integration of Java 2 Standard Edition SDK Classes**

Any Java class from the J2SE™ 1.4 class libraries can be accessed from Visual Studio .NET, enabling developers to access the rich functionality of the Java APIs. Visual MainWin integrates the documentation of the J2SE classes into the Visual Studio help system.

- **Cross-language Debugging**

Visual MainWin allows developers to debug across multiple languages: C#, Visual Basic .NET, and Java code using the Visual Studio debugger. When the Java source code is available, developers can step through Java code and follow the execution flow of native Java modules within a single execution stack.

Product Benefits

Enterprises can reduce development costs and time-to-market significantly using Visual Basic .NET and C# developers to create Web applications and Web services for J2EE servers, without replacing or attempting to retrain their .NET or C# development staff.

Large-scale IT organizations with both .NET and Java developers can accelerate the development process significantly by enabling their .NET and Java development teams to co-develop applications for the J2EE platform.

Visual Basic .NET and C# developers rapidly create the front-end of the application in the Visual Studio development system, while Java developers provide the back-end business logic and J2EE components. Visual MainWin enables VB.NET and C# developers to easily reference and consume Java components such as Enterprise JavaBeans directly within Visual Studio .NET. The entire application can then be compiled and deployed natively on an industry-leading J2EE application-server such as IBM WebSphere®, BEA WebLogic®, or JBoss®.

Because integration between .NET and J2EE occur at the development stage, there are no application server integration costs.

About Mainsoft

Founded in 1993, Mainsoft Corporation, the cross-platform development company, enables businesses to develop mission-critical applications with Visual Studio development software and deploy them natively on J2EE, UNIX and Linux, dramatically reducing development costs and time-to-market. The company is a first-mover in cross-platform development. Its world-class research and development team has created patented cross-platform products that solve critical problems facing independent software vendors (ISVs) and IT organizations. Many of the world's largest ISVs, including Siebel, Computer Associates and IBM Rational, use Mainsoft's products to extend the productivity of Microsoft Visual Studio, deploying more than \$1 billion worth of software annually on multiple operating systems. For more information, visit www.mainsoft.com.

©Copyright 2004, Mainsoft Corporation

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.