Delivering Better Software, Faster

for the Microsoft® .NET Framework

Solutions to accelerate the application lifecycle

A Borland White Paper

June 2003

Borland®

Contents

Abstract

Efficient and flexible software systems are essential for successful enterprises. Yet, according to some research, at least half of all in-house development projects fail to meet expectations.

With Borland® solutions, organizations have the technologies to deliver better software, faster — software that is designed to strengthen customer relationships, to increase operational efficiency, to create market leaders.

Good systems are reliable, dependable, and secure. Their capabilities match the expectations of users and management, and they can be trusted to perform well under pressure. Good systems are flexible, so changes can be made to accommodate new ways of doing business.

To strengthen the development of good systems for the Windows® platform, Microsoft has developed the Microsoft® .NET Framework. This comprehensive platform includes features designed to make applications reliable and secure, and it simplifies maintenance by helping to make development and deployment less complex.

Already the Microsoft .NET Framework is promising to be widely accepted. In a September 2002 report, "*Leading Programming Languages for IT Portfolio Planning*," Gartner forecast that the number of programmers using .NET languages will exceed 4 million – more than half the number of current programmers worldwide.

Truly excellent systems development requires more than just a first-rate deployment platform. Meeting the needs of users and ensuring that systems are flexible requires first-rate capabilities from a development team. This team will need software that supports the entire application development process from requirements definition, through design and development, to deployment and maintenance.

The Borland application lifecycle solutions for the Microsoft .NET Framework provide an integrated approach to supporting the entire systems development process. With Borland, teams have the technologies to rapidly create systems that meet user demands and which

integrate with existing applications and data. No other solution for .NET development gives developers such an open and flexible development capability.

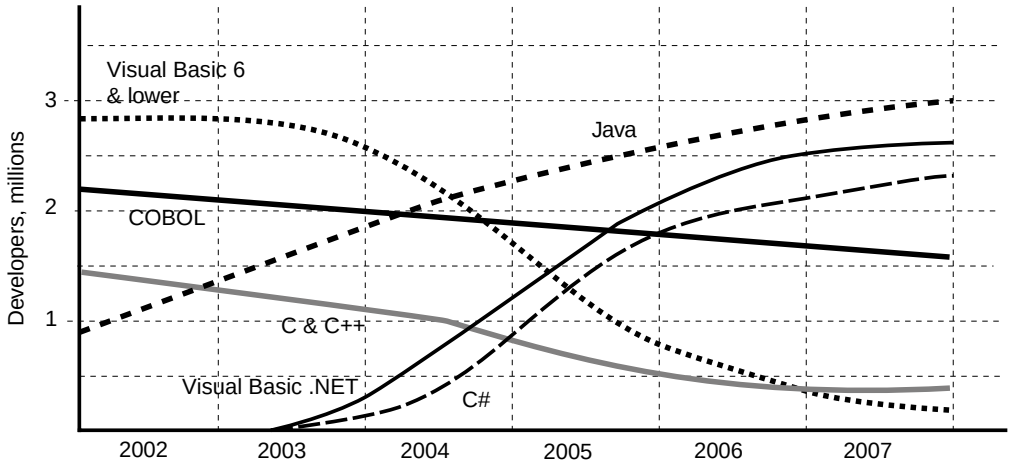


Figure 1: *Number of professional developers worldwide (Gartner, 2002)*

Solutions for the application lifecycle

Research conducted as long ago as 1992 by the Software Engineering Institute at Carnegie Mellon University showed that adopting simple repeatable processes dramatically increases software quality and reduces the time required to create applications by as much as 80 percent.

Software developers today are involved with every part of a business, and so more people in general are involved in delivering high-quality applications than ever before. This means it is more effective to build processes that involve all team members including designers, developers, testers, deployment teams, and managers. The ability to recognize, adopt, and repeat best practices is a good practice, as is good communication among all team members.

Application Lifecycle Management (ALM) regards the process of delivering software as a continuously repeating cycle of interrelated steps: definition, design, development, testing, deployment, and management.

A true application lifecycle solution

- shortens time-to-market as developers need focus only on business requirements
- improves quality, ensuring the final application meets both the needs and expectations of users
- increases productivity, as team members share best practices for development and deployment
- accelerates development through simplified integration
- cuts maintenance time by synchronizing application and design
- maximizes investments in skills, processes, and technologies
- and increases flexibility by making it faster to build and adapt applications that support new business initiatives.

Borland has a complete set of integrated development solutions for the Microsoft .NET Framework to support development teams throughout the application lifecycle. The integrated Borland ALM solution is designed to streamline the complete application lifecycle, speeding time-to-market while increasing quality.

Organizations typically have a mix of different systems, so when completed, a Microsoft .NET Framework system will most likely need to interoperate and exchange data with other applications. It will likely need to be deployed alongside other platforms such as Java™ 2 Platform, Enterprise Edition (J2EE™) and databases that are not from Microsoft, such as IBM® DB2® or an Oracle® database.

To support this increasingly common requirement, the Borland solution is open, supporting a wide range of development and deployment technologies such as J2EE and Enterprise JavaBeans™ (EJB™), CORBA,® Oracle, and DB2. This support reflects the Borland

commitment to platform independence and interoperability: *Borland does not tie organizations to any platform for development or deployment.*

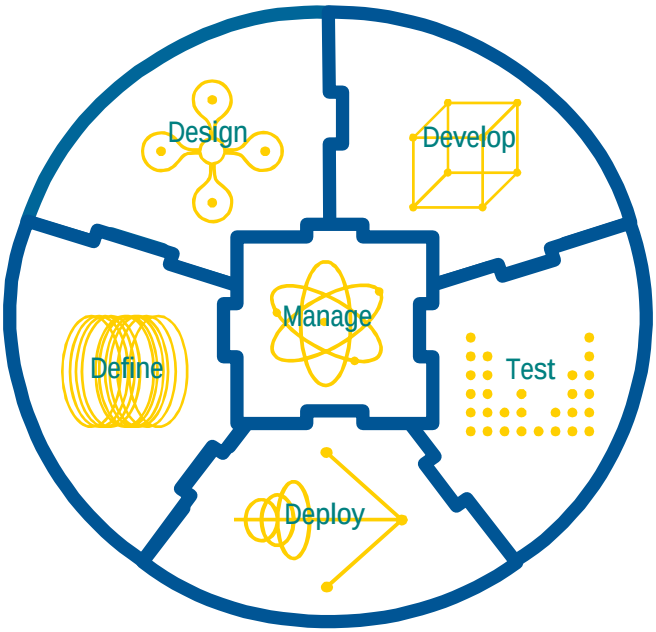


Figure 2: *The Borland view of the application lifecycle*

The Borland approach to managing the application lifecycle recognizes that software development is an iterative process and does not necessarily flow sequentially from one phase to the next. Rather, it can be expected to perhaps move back from testing to design, then to deployment, and then possibly back to definition as the needs of the business change. The Borland solution supports this need for flexibility by providing closely integrated components with powerful change management capabilities.

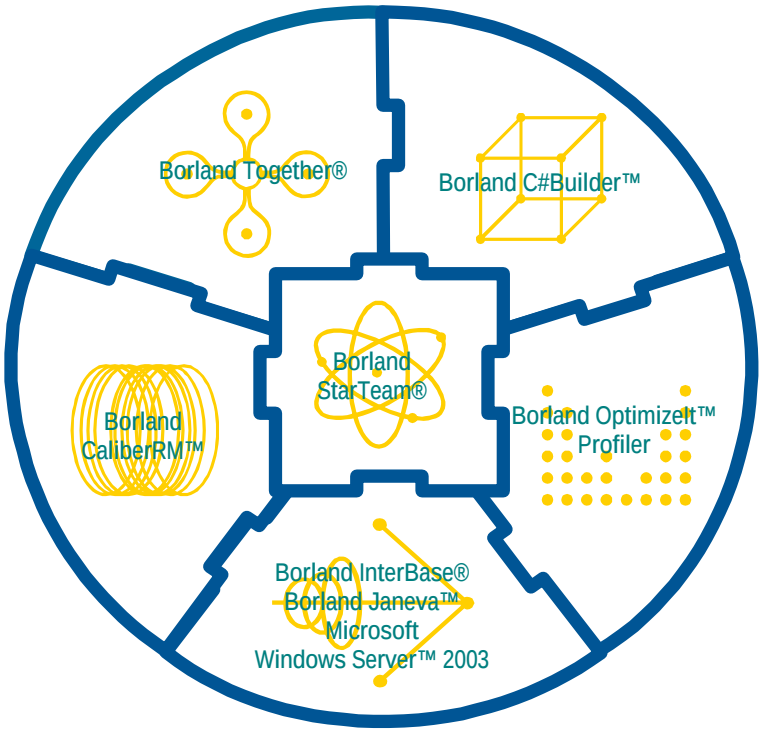

Borland Together®
Borland C#Builder™
Borland StarTeam®
Borland CaliberRM™
Borland OptimizeIt™ Profiler
Borland InterBase®
Borland Janeva™
Microsoft Windows Server™ 2003

Figure 3: *Each Borland solution is designed to work with each other to provide a complete application lifecycle solution*

Borland products are high-performance solutions that, when used together, are designed to deliver a competitive advantage through software, providing management teams with visibility and control of the complete application lifecycle.

Requirements definition

Without a formal definition of what an application is supposed to do, it is difficult for a development manager to allocate resources and plan time properly. It can be difficult to estimate the impact that changes will have on the development schedule unless the project plan ties back to the initial set of requirements.

The starting point for any application should be a formal definition of what it is supposed to do and the benefits that it is intended to deliver. When made easily available to team members, this definition unifies the business objectives of an application. This shared vision helps communication among management, development, and the eventual users of the system.

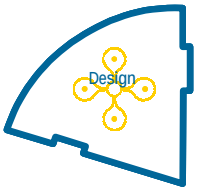
The scope of the application must be clearly defined, and the benefits of each feature considered against the cost of development. All the following phases of the application lifecycle refer back to the requirements as they communicate to analysts which parts of the system are mandatory and what the business implications of change may be.

***Borland*® *CaliberRM*™** is a complete solution for requirements management. It integrates closely with project management systems such as Microsoft Project and links with the *Borland*® *StarTeam*® configuration management solution to communicate the impact of changes. The requirements definition can be used as a reference for all the subsequent and interconnected stages in the lifecycle of the application.

Because requirements form the foundation for later design and development, referencing them from other phases of the application lifecycle must be an easy matter. CaliberRM provides requirements information to other technologies in the Borland ALM suite. StarTeam,

Borland® Together® technologies, and Borland® C#Builder™ are all integrated with CaliberRM. This integration supports the need for the requirements, which represent the scope and expectations for the application, to be properly communicated to all team members within their preferred environment.

When used as part of the Borland ALM solution, CaliberRM helps keep development teams focused on the features that are most important to the project.



Analysis and design

Like any complex engineering endeavor, software benefits greatly from proper analysis and design. Good design helps teams provide a strong foundation for their development activities. A poor design generally leads to software of lower quality and, over time, slows down a team's ability to make changes.

Effective design helps aid communication, which is especially important in large organizations. Through standard modeling languages such as UML™ (Unified Modeling Language™), software design tools can themselves provide an effective communications medium. A model can serve as a common language used to guide the software development process.

Understanding how applications work can be challenging, especially if they were written some time ago. In addition to helping build new applications, design tools can create visual descriptions of existing systems that help teams understand the architecture and organization of their applications. This knowledge enables developers to more effectively make quality changes.

Borland Together technologies provide a comprehensive analysis and design solution for architects, designers, and developers. Together technologies provides a timely, effective solution for understanding and designing applications built using Microsoft .NET technology.

The unique LiveSource™ technology helps verify that the application design stays fully synchronized with changes made throughout the software lifecycle. As the C#, Borland®

C++Builder® or Microsoft® Visual Basic.NET program code changes, the model is kept up-to-date; conversely, as the model changes, the program is kept up-to-date.

Alongside the design capabilities, Together is also designed to audit software applications, helping enforce corporate standards and fix potential errors. This function is a great complement to software inspections, often enhancing the quality of applications.

Borland Together also offers capabilities to measure applications, providing data about size, complexity, and dependencies, which can help prioritize testing and re-factoring efforts. Additionally, by taking "snapshots" of the metrics, teams can better understand both the evolution of their applications and the quality of the application development processes and related activities.

Development

Once the requirements are understood and the design has been started, program coding can commence. Typically this is a continuous cycle – as the application is created, modifications will need to be made to the design.

Application developers require more than just a first-rate IDE for code editing and debugging. The IDE has to be fast, so there are no interruptions to the coding process. It has to have deep support for the platform, providing extensive wizards and pre-built components. And, as the heart (from a developer's viewpoint) of the application lifecycle, it should support the full range of technologies that the application will use.

Most importantly, the development solution should work directly with the application model. Without direct support for the model in the IDE, there will need to be extra manual steps involved in interpreting the modeling language. Thus, development is likely to be slow and prone to mistakes.

Borland C#Builder for the Microsoft .NET Framework is a development environment for C# and Visual Basic.NET, focused on the specific needs of enterprise developers. C#Builder is the development center for the Borland ALM solutions for the Microsoft .NET Framework.

With C#Builder, developers can integrate their applications for the Microsoft .NET Framework with a broad range of other technologies—including program code written for Java, J2EE, and CORBA middleware. Borland C#Builder also includes ADO.NET drivers for leading databases including DB2, Oracle, Microsoft® SQL Server® 2000, and *Borland® InterBase.®*

Borland C#Builder goes beyond simple integrated support for modeling tools. Not only does it support integration with modeling tools such as *Borland Together* for round-trip design, but it also supports model-driven development through *Borland Enterprise Core Objects (ECO™).*

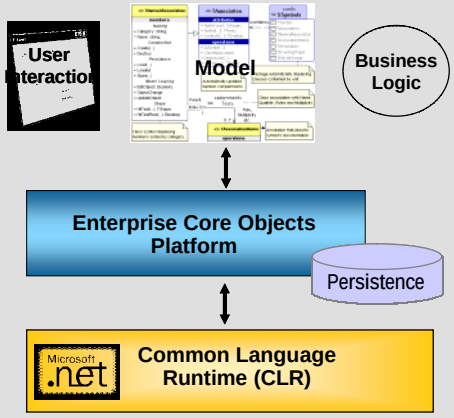
By using Borland ECO, development time can be greatly reduced as there is no need for actual programming. Because the model itself is used to drive the application, the quality of

the finished application is generally higher – and designers can be more confident that the program matches what was first defined in the model.

Borland® Delphi™ already supports the .NET Framework through a preview release of Delphi for the Microsoft .NET Framework. Borland has announced a new project, codenamed "Project Octane," designed to deliver Delphi™-language programming capabilities and a full component library for the Microsoft .NET Framework. The next version of Delphi is currently planned for release by the end of 2003.

Borland Enterprise Core Objects (ECO™)

Borland C#Builder supports design-driven development. The Borland Enterprise Core Objects (ECO) can be run directly from a UML model, without the need for programming. The only C# code necessary is used for the user interface: all the business logic is contained in the model and never actually seen as C# code.



Design-driven development of this kind can deliver productivity benefits, as there is no need to generate program code. The application matches the design, so even quite extensive changes to the system logic can easily be made without the potential to introduce programming errors.

Applications using ECO can be created by C#Builder, and subsequently the components can be integrated with programs written with Microsoft® Visual Studio.NET. This flexibility makes it practical to use C#Builder for portions of a large enterprise project, where the dramatic increase in productivity can be realized. To create the model, developers can use the Borland Together design environment; with ECO, powerful rapid-application development capabilities are added to the Together design features.

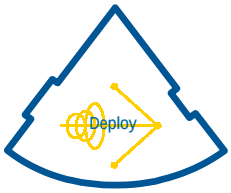
Testing and profiling

It's especially important that mission-critical applications used across an enterprise are fast, stable, and reliable. While the .NET Framework makes it easier to develop systems, it introduces an additional technology layer that limits the control developers have over the performance of their systems.

The regular use of performance tools throughout the development process is therefore critical for keeping performance and reliability issues under control.

Borland® Optimizeit™ Profiler for the Microsoft® .NET Framework is designed to help enterprises manage the performance of their applications. Tightly integrated with both *Borland C#Builder* and Microsoft® Visual Studio.NET, the Optimizeit Profiler supports all .NET managed code, including C#, Visual Basic.NET, and Managed C++. Optimizeit Profiler provides real-time performance profiling to help developers solve performance issues quickly.

By addressing performance management early in the development cycle, enterprises can deploy applications with greater confidence. This approach helps teams minimize the incidence of deep-rooted performance issues that can threaten an entire project. Teams can spend less time at the end of a project diagnosing and solving performance problems; using Optimizeit Profiler as part of the application lifecycle helps reduce risks in development and deployment.



Deployment

Platform interoperability

For most organizations, .NET applications will be deployed in a mixed IT environment where existing technology investments in J2EE and CORBA must be leveraged. To enable this platform interoperability, developers can use Web Services and similar bridging technologies to provide alternative ways to exchange data. However,these technologies potentially could be slow, insecure, and lacking in important capabilities such as transaction management.

Addressing these concerns, ***Borland® Janeva™*** is designed to provide seamless , high-performance interoperability between Microsoft .NET Framework applications and J2EE and

CORBA infrastructures. Using the highly scalable and secure Internet Inter-ORB Protocol (IIOP™), Janeva allows Microsoft .NET Framework client- or server-based applications to access J2EE and CORBA server-side components, leveraging the enterprise qualities of service these technologies provide.

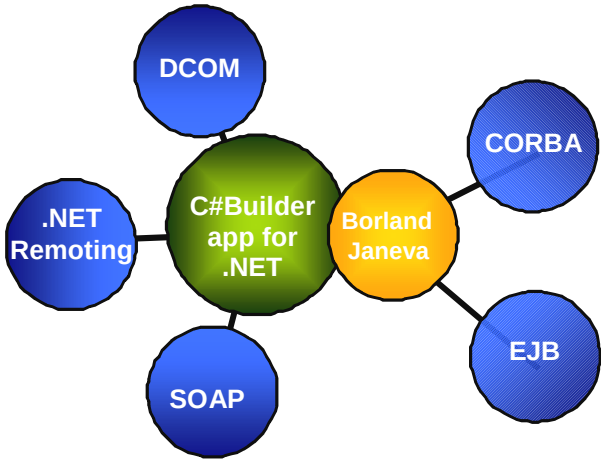


Figure 4: *Borland Janeva is designed to give .NET systems seamless access to EJB™*

Janeva is tightly integrated with popular development environments for the Microsoft .NET Framework, including C#Builder and Visual Studio.NET. Implementation requires no expertise with J2EE or CORBA technologies and no change to back-end systems. This low-impact interoperability helps simplify the integration of new .NET applications into existing enterprise infrastructures so businesses can take advantage of new technology while preserving their existing applications.

Databases

A database underlies virtually every business application, and database selection affects the cost of ongoing maintenance over the period—often many years—that an application is in production.

Borland solutions give the developer the opportunity to select the database appropriate for each project. ***Borland InterBase*** is a powerful, embeddable database that is certified for

Microsoft® Windows Server 2003, integrated for exceptional productivity with *Borland C#Builder* and *Together technologies* and open for use with Microsoft Visual Studio. The power and small footprint of InterBase, together with its minimal requirements for maintenance, make it an excellent choice for applications that will be deployed widely.

Change and configuration management

Enterprise systems need to have a way to manage complex workflow and sign-off procedures to ensure accountability and maintain control. These systems should be backed up by effective communication tools such as discussion boards and defect tracking utilities.

Without effective change tracking, development managers do not have an effective view of the progress of their project development. This lack of visibility in turn means that they can lose control over the process, leading to slipped deadlines.

Effectual configuration management ensures that where necessary, maintenance teams can roll back the application to the point where it was deployed. Setting up the system to replicate the problem is simplified with an effective configuration management solution.

With an effective configuration management solution, developers can find it possible to maintain two or more parallel tracks of development at one time. StarTeam gives capabilities for control of the assets in a company's IT portfolio, so appropriate controls and audit checks can be created and maintained.

Borland StarTeam is a comprehensive solution to the change and configuration management needs of large organizations, including widely distributed project teams and enterprise-level projects. Designed from the outset to work with the Internet, StarTeam uses a central repository to store and monitor access to each project asset.

The StarTeam server's open, object-based architecture also supports linking of digital assets, enabling team members to more easily access and track changes to the lifecycle deliverables associated with their development.

StarTeam includes powerful workflow capabilities to give managers a comprehensive audit trail throughout the application lifecycle.

Configuration management is a central part of the Borland application lifecycle solution. Borland StarTeam integrates directly with the other Borland solutions, including the *Borland CaliberRM* requirements management solution, *Borland Together* for modeling and design, and *Borland C#Builder* for application development.

The strength of integration

When choosing software for phases of the application lifecycle, managers often select products that are from different vendors. Some suites offer benefits such as strong ties between each product for each phase of the application lifecycle, but they frequently lock the organization in to a particular technology framework. Often, support from these suites for third-party deployment platforms is weak –and frequently, it is nonexistent.

For the Microsoft .NET Framework, Borland products integrate well with each other but also integrate well in combination with third-party products that may be in use. This means that there is no lock-in to proprietary systems that restrict freedom of choice. Borland solutions allow developers to work flexibly with a broad range of technology.

Each Borland solution supports touchpoint integration both with Borland products and also with tools from other vendors. *Touchpoint integration* puts each solution a menu option away, supporting compatibility and automatically translating between different file formats. This increases productivity and reduces the risk of error, also encouraging developers to use the breadth of capabilities offered by the joint solution.

For example, a comprehensive interface to the Borland StarTeam configuration management solution is available directly from Borland Together technologies, Borland C#Builder, and Microsoft Visual Studio.NET. This both encourages developersand makes it easier for them to use the capabilities of source code control – vital if development managers are to be confident in the integrity of their systems.

Many Borland solutions use *synergistic integration* to make the development process coalesce. With this approach, the capabilities of two products are blended together transparently. The developer may not even realize that two products are being used.

The relationship between Borland C#Builder and Borland Janeva is an excellent example of this synergistic integration. Once installed, Janeva directly translates between J2EE and the Microsoft .NET Framework without requiring developers to learn complex translation mechanism. By making interoperability more straightforward, .NET developers need no additional skills or training; Janeva is designed so integrated solutions can be developed much faster.

Independent solutions with the strength of integration

By choosing Borland, project teams know they are using exceptional solutions for application lifecycle tasks. Teams also gain great benefits in usability and ease of learning through integration of each separate element. The result is a smoother, more predictable process that leads to the building of better software, faster.

The power of independence

Most organizations now have a mixed environment where software for Microsoft Windows works alongside Java, Linux,® and other platforms. Only Borland has solutions for development on each platform in common use today. Borland Delphi and C++Builder deliver streamlined development for the Windows® 98, Windows NT,® and Windows XP® platforms, and Borland® Kylix™ is a powerful component-based solution for Linux development. For Java and Enterprise JavaBeans, Borland JBuilder is the leading solution. New Borland C#Builder is the first independent development solution for the Microsoft .NET Framework (and includes the new Borland Janeva developer software).

Borland also has powerful solutions for every other phase of the application lifecycle for these three platforms. For example, Borland Together provides powerful UML design and modeling for Java as well as for C#, and the Borland StarTeam configuration management solution supports Windows, Linux, and UNIX® clients. Borland Janeva provides interoperability during the deployment stage: the Janeva runtime component is embedded in the deployed thin- or thick-client Microsoft .NET application.

Using Borland® solutions with Microsoft® Visual Studio®.NET

Microsoft® Visual Studio .NET is a central part of the development plans for many organizations. The Borland solution for application lifecycle management complements Microsoft Visual Studio .NET in several significant areas.

Overall, the Borland solutions go beyond Microsoft Visual Studio.NET by providing integrated support for requirements gathering, design, application profiling and interoperability. Each of these solutions can work well alongside Microsoft Visual Studio.NET, offering both touchpoint integration and synergistic integration where appropriate.

The development solution Borland® C#Builder™ also complements Microsoft Visual Studio .NET. Besides providing productivity enhancements (such as live data in designers), C#Builder supports component-based development, making it easier for architects to pre-build components ready for other developers to use as high-level building blocks.

Borland Janeva is also tightly integrated with the Microsoft Visual Studio .NET development environment, speeding the development of .NET applications that need to interoperate with J2EE and CORBA infrastructures.

Direct integration across enterprise platforms

Mixed environments are here to stay, and so forward-looking organizations must find ways to directly integrate systems across multiple platforms. Borland provides integration solutions at several different layers:

- The highest level of integration can be performed through Web Services. This standards-based approach can be used between most technology platforms and is supported by each of the Borland products solutions as well as by the Microsoft .NET Framework.
- At a lower level, Borland Janeva is architected to communicate using IIOP, which has been shown to provide higher levels of performance over the Simple Object Access Protocol (SOAP) used by Web Services. By basing its framework on IIOP and adhering to the J2EE, CORBA, and .NET standards, Janeva not only supports high performance, it also delivers interoperability with other J2EE and CORBA infrastructures that are compliant with IIOP.

- An alternative approach is to create a model of the business system, then create different implementations of parts of the system for different platforms. For instance, Borland Together supports code generation of both C# and Java, so it can create systems for either language from a single UML model.

Borland has proven technology solutions to address this breadth of mixed solutions. For Microsoft .NET in particular,. no other development solution is as broad or as open.

Summary

For 20 years, Borland has been dedicated to creating world-leading development and deployment products. This expertise is now leveraged across the entire application lifecycle – from requirements management, through design and development to testing and deployment.

This heritage is now brought to the Microsoft .NET Framework with a set of integrated products designed to positively transform the productivity of development teams. The Borland ALM solutions that complement the Microsoft .NET Framework include:

- *Borland® CaliberRM™* for requirements management,
- *Borland® Together®* for application design using UML,
- *Borland® C#Builder™* for design-driven development with C#,
- *Borland® Optimizeit™ Profiler* for application profiling,
- *Borland® Janeva™* for interoperability with EJB, J2EE, and CORBA,
- *Borland® InterBase®* for low-maintenance embedding alongside Microsoft Windows Server 2003
- and *Borland® StarTeam®* for configuration and change management.

Each of these solutions integrates with one another to provide developers with one of the highest development productivity solutions available.

For more information on the Borland solutions for the Microsoft .NET Framework, see the Borland Web site at http://www.borland.com/dotnet.