

# .NET vs J2EE

By John Papproth

**.NET** vs J2EE again! Yes, one more article on the virtues and vices of these two competing specifications for Enterprise Application Development. This article will compare the Microsoft .NET Framework (.NET) to the Sun Microsystems Java 2 Platform Enterprise Edition (J2EE). The goal of this article is to gain insight into both specifications, but not to identify the winner of a head to head match up! Besides, looking over the results of a search on articles already written on this topic, it becomes pretty clear that we've been trying to find that winner for at least a year or two.

It could be that we're trying to find the answer to the wrong question. Instead of asking which is the better choice based on performance benchmark data (<http://www.gotdotnet.com/team/compare/default.aspx>), we should be looking at how to best use each product given our own unique mix of talent, existing products and tools. It may just be a question of availability, preference and application.

While the .NET Framework and J2EE specifications have similar implementations, they also were developed with very different philosophies.

## WHAT IS .NET? MANY LANGUAGES, ONE SYSTEM

.NET provides a single development and deployment platform for many different languages. While the .NET product was designed to take advantage of distributed systems deployed over the Web, one other apparent goal is to minimize the impact of the choice of source language on the development of those systems. This minimizes language interoperability issues and maximizes the use of available skill sets. With .NET you are looking for a .NET Programmer, not a C# Programmer or a Visual Basic Programmer.

All sources, regardless of the language, are compiled into Microsoft Intermediate Language (MSIL). The Framework Class Library (FCL) is common to all .NET compliant languages. The FCL raises the level of support for all languages. For example, multi-threading capabilities are incorporated into the FCL, not inherent to the language itself. The Common Type System (CTS) defines how types are declared, used and

managed in the Common Language Runtime (CLR) environment. There is no need for questions like "Is this a C String or a VB String data type?" All of these components of the .NET Framework support language interaction.

The Visual Studio.NET product offers a common Integrated Development Environment (IDE) for each language. Construction, testing and deployment use a similar tool regardless of the development language.

Visual Studio.NET 2003 is delivered with support for the Visual Basic, C#, J#, and C++ languages. You can compare the benefits of each in the Microsoft white paper at <http://msdn.microsoft.com/vstudio/productinfo/whitepapers/default.aspx>.

Other vendors can and have followed the Microsoft Common Language Specification (CLS) in order to develop their own .NET-compliant languages. Built on the .NET framework, ActiveState's PerlASPX adds Perl to the list of supported ASP.NET languages. Fujitsu even offers NetCobol for .NET! And to think that I was worried about choosing between C# and Java!

The CLS attempts to ensure interoperability between different languages.

## WHAT IS J2EE? MANY SYSTEMS, ONE LANGUAGE

Java is a single language that can be developed and deployed on many different platforms. The J2EE specification written by Sun Microsystems has as its core the Java 2 Platform Standard Edition (J2SE). It has been implemented by a variety of vendors.

According to Sun Microsystems, J2EE takes advantage of many features of the Java 2 Platform, Standard Edition, such as "Write Once, Run Anywhere" portability, JDBC API for database access, CORBA technology for interaction with existing enterprise resources, and a security model that protects data even in internet applications. Building on this base, Java 2 Enterprise Edition adds full support for Enterprise JavaBeans components, Java Servlets API, JavaServer Pages and XML

technology. The J2EE standard includes complete specifications and compliance tests to ensure portability of applications across the wide range of existing enterprise systems capable of supporting J2EE.

The J2EE Version 1.4 SDK is now final and can be downloaded from <http://java.sun.com/j2ee/1.4/download-dr.html>. The SDK contains the Sun Application Server and J2SE v1.4.

The following is a list of vendors who offer tested implementations of J2EE v1.3.

- ▼ BEA WebLogic Server 7.0
- ▼ BEA WebLogic Server 8.1
- ▼ Borland Enterprise Server, AppServer Edition
- ▼ Borland JBuilder 6.0
- ▼ Fujitsu INTERSTAGE Application Server
- ▼ IBM WebSphere Application Server 5.0
- ▼ IONA Orbix E2A Application Server
- ▼ Macromedia JRun Server Technology
- ▼ Novell exteNd Application Server 5
- ▼ Oracle 9i Application Server
- ▼ Pramati Server 3.0
- ▼ SAP NetWeaver Web Application Server
- ▼ SAS AppDev Studio 2.0.2 Preview Release
- ▼ Sun Java System Application Server 7
- ▼ Sybase EAServer 4.1
- ▼ Tmax Soft JEUS 4.0
- ▼ Trifork Application Server 3.1
- ▼ J2EE SDK v1.3

## INTEROPERABILITY BETWEEN .NET AND J2EE

Both .NET and J2EE support the creation and deployment of Web Services. It is here that we have an opportunity for communication between the platforms.

A Web Service is a method that is located on the server, which receives and sends messages to clients using HTTP. The messages are wrapped in the Simple Object Access Protocol (SOAP). SOAP uses Extensible Markup Language (XML) to encapsulate the message and parameters.

Did we have the same opportunity for communication before Web Services? Yes. ODBC allowed intercommunication using the Database Management System of your choice and Stored Procedures. The business logic was encapsulated within the Data Layer as a Stored Procedure Call.

With Web Services, things are just a little cleaner, and the connection is a lot longer and less complicated using HTTP.

## NOTABLE

The .NET Framework is required on any machine where a compiled .NET application will be executed. It can be downloaded and installed using the Windows Upgrade facility. This is very similar to the Java Virtual Machine that is required by Java applets and applications. At the time of this writing the JVM successfully runs on MAC, Windows and UNIX operating systems. Framework successfully runs on Windows 98 and above.

However, Microsoft had submitted the Common Language Infrastructure (CLI) and its C# language to the Ecma for standardization. Ecma International is an industry association founded in 1961 and dedicated to the standardization of Information and Communication Technology (ICT) Systems.

TABLE 1: FEATURES OF .NET AND J2EE

Feature	.NET Implementation	J2EE Implementation
Languages supported	VB.NET, C#, C++, J#, others	Java
Development environment	Visual Studio.NET	Java Studio Standard 5 update 1, IBM WebSphere, others.
Runtime environment	.NET Framework	Java Runtime Environment (JRE)
First compilation output from source (both systems use Just In Time (JIT) compilation at execution)	Microsoft Intermediate Language (MSIL) and Bootstrap code in a windows executable (.EXE)	Bytecode (.CLASS)
Execution management	Common Language Runtime (CLR)	Java Virtual Machine (JVM)
Support library	Framework Class Library (FCL)	Java API
Database support	ADO.NET both SQLServer and OleDb objects	JDBC
Web support	ASP.NET	Java Servlet JavaServer Pages (JSP) Enterprise JavaBeans
Desktop support	Windows Forms	SWING and the AWT
Web Services	Integrated into Visual Studio.NET using UDDI, WSDL, SOAP, and XML	Java API for XML Processing (JAXP) Java API for XML Registries (JAXR) Java API for XML-based RPC (JAX-RPC) SOAP with Attachments API for Java (SAAJ)

This implementation of the CLI and C# is available for download from Microsoft and builds and runs on Windows XP, the FreeBSD operating system, and Mac OS X 10.2. It is released under a shared source initiative. For further information see the Microsoft Shared Source Initiative (<http://www.microsoft.com/resources/sharedsource/default.mspx>).

## MICROSOFT'S MIGRATION PATH (C# OR J#?)

Microsoft has always been helpful in supplying tools for wayward programmers who may still want to follow the Microsoft way. Concerning the Java language, however, there is a fork in the migration path and the decision is left to the programmer.

Microsoft's Java Language Conversion Assistant 2.0 is a tool that automatically (beware of the magic word "automatic") converts existing Java-language code into Microsoft Visual C# for developers who want to move existing applications to the .NET Framework.

For Java programmers, J#.NET may feel like an easier option. The J# language supports the Java language syntax through Java version 1.1.4. However, the current release of the J2SE is 1.4.2 with version 1.5 being reviewed. Supported Java API calls are handled by the Framework Class Library.

For any of you who had used Visual Studio 6.0, Microsoft originally offered a Java Development platform called J++, although it was never fully accepted into the Visual Studio family. It was installed as an additional product that could be downloaded. With the announcement of .NET J++ was abandoned and J# was born. Although J# is now included with Visual Studio.NET, it is my opinion that C# is the "favored" language.

## CERTIFICATION

Learning is a lifelong process and in Information Technology a lifetime is about 12-16 months!

Take a look at yourself and your current staff. Do you know what your skill base looks like today? Is your shop a large group of trained and certified Java developers? Do you exclusively develop applications for Visual Basic.NET? Do you have a diverse development and client environment, consisting of Microsoft Windows, Sun, Linux, and IBM AS400 WebSphere? Is your development and client environment all the same release of the Microsoft XP Professional Operating system?

My guess would be that the majority of your staff are operating in a very diverse environment. Diversity and size of the company seem to be directly related.

The issues surrounding a move from one technology to another involve training and possibly hiring decisions. In a diverse environment, we need to seek to optimize our current skill base and focus on the issue of interoperability.

Acquiring certification in a given product or development area is a benefit for the individual and for the employer. For the individual, it reinforces the skill that is gained in working with a particular product or platform. For the employer, certification offers a common rating system on what has become a global employment pool.

Brainbench (<http://www.brainbench.com>) offers vendor neutral testing on many Information Technology (IT) and non-IT topics. There are excellent tests on Java 2, J2EE, ASP.NET, ADO.NET, C#, VB.NET, and .NET Framework.

Microsoft offers certification for developers who use the .NET Framework (<http://www.microsoft.com/learning/mcp/default.asp>):

- ▼ Microsoft Certified Application Developers (MCADs) use Microsoft technologies to develop and maintain department-level applications, Web or desktop clients, or back-end data services.
- ▼ Microsoft Certified Solution Developers (MCSDs) design and develop leading-edge business solutions with Microsoft development tools, technologies, platforms, and the Windows architecture.

Since J2EE is a specification that can be implemented by a variety of vendors, it should not be surprising that certification can be obtained not only from Sun Microsystems (<http://suned.sun.com/US/catalog/>

[java/j2ee.html](http://suned.sun.com/US/catalog/)) but also from the vendors that have implemented the J2EE specification.

Sun offers the following Java technology professional certifications:

- ▼ Sun Certified Programmer for the Java 2 Platform
- ▼ Sun Certified Developer for the Java 2 Platform
- ▼ Sun Certified Web Component Developer for the Java 2 Platform, Enterprise Edition (J2EE)
- ▼ Sun Certified Business Component Developer for the Java 2 Platform, Enterprise Edition
- ▼ Sun Certified Enterprise Architect for the Java 2 Platform, Enterprise Edition

IBM also offers certification in its J2EE Product Offering, WebSphere.

## RESOURCES:

---

<http://java.sun.com/j2ee/index.jsp>

<http://suned.sun.com/US/certification/java/index.html>

Microsoft .NET and J2EE Interoperability Toolkit by Simon Guest (Microsoft Press)

<http://msdn.microsoft.com/vstudio>

<http://msdn.microsoft.com/netframework/>

<http://www.microsoft.com/learning/mcp/default.asp>

<http://www.gotdotnet.com>

<http://www.ecma-international.org/publications/standards/Ecma-334.htm>

<http://www.ecma-international.org/publications/standards/Ecma-335.htm>

---

*NaSPA member John Papproth is a both an MCSD.NET (Microsoft Certified Solution Developer) and SCJP (Sun Certified Java Programmer). He is a consultant for Bass & Associates Inc. a Business Solutions Provider in Omaha NE. John has taught courses in both the Microsoft .NET language suite and the Sun Microsystems Java Language.*