

Project Ace: A Better Way to Build Enterprise Applications

Project Ace Cuts the Time and Expense of Writing Applications, Delivers the Flexibility to Deploy on Any Architecture, and Improves Performance and Scalability

Contents

Executive Summary	2
Problems with Traditional Development	3
More Specialists Means More Expense	3
Lack of Architectural Flexibility Limits Options	3
Badly Distributed Code Slows Application Performance	3
Optimization is a Slow, Expensive, Manual Process	4
Ace: The Optimizing Compiler for the Network	4
Simplifies the Development Process	4
Keeps Architecture Choices Open	5
Optimizes Performance and Improves Scalability of Applications	7
Example: Java Pet Store Application	7
Summary: The Business Benefits of Ace	8
Ace Road Map	9
For More Information	9
About Sun Microsystems Labs	9
About Sun Microsystems, Inc.	10

Executive Summary

There is no question that software has become the key to competitive differentiation and profitability. The spoils of e-business go to the companies that have the agility to develop and deploy high-performance, highly scalable network services that improve customer satisfaction. At the same time, companies that are able to adapt their services quickly to meet changing customer requirements are more likely to maintain customer loyalty—a critical requirement in the era of e-business.

Yet most enterprises are still struggling to solve age-old IT dilemmas:

- ? How will we keep up with customer requirements that change by the day when development cycles are measured in weeks and months?
- ? How do we squeeze more productivity out of a development staff that is already severely overburdened?
- ? How do we deal with migration of applications to new waves of technology and software architectures?
- ? How can we optimize application performance without increasing the costs and timeframes of development projects?
- ? How can we make the IT department a strategic asset when it continues to be mired in day-to-day technical issues?

A research team at Sun Microsystems Laboratories has a simple answer: Project Ace. With Ace technology, developers can simplify and automate the development of enterprise Java? applications, create applications that are easy to migrate from one architecture to another, and optimize performance and scalability. The result: substantially shorter development timeframes, the flexibility to deploy new or existing applications virtually anywhere--regardless of the underlying architecture, and higher application performance.

Project Ace has proven its capabilities and its potential at Sun, and Sun Labs has a clear roadmap to further refine and enhance the technology. This paper describes how Ace can help companies solve core business and technical challenges; presents an overview of the key breakthroughs that give it a unique advantage over current methods of compiling applications; and shows how it is being used today.

Problems with Traditional Development

Application development has always been hard work. But today, it seems to be getting more difficult, not less. And the results are felt throughout the organization. Every developer working in an enterprise IT department knows the frustration of working with manual coding tools to create dynamic applications. Every IT manager has experienced the irritation of having a senior executive suddenly change the core requirements of a software project that has been in the works for six months. And every senior executive has felt the exasperation of seeing the company miss a key business opportunity because the IT department couldn't build the needed software in the required timeframe.

These are the challenges that the Ace research team at Sun Labs set out to address. This section examines the underlying problems with traditional coding methods; the following section shows how Ace provides solutions.

More Specialists Means More Expense

Object-oriented development and code re-use techniques were supposed to increase the productivity of developers. But today, programmers continue to face huge and growing backlogs of development projects. The sheer volume of code to write continues to grow exponentially as the diversity of hardware platforms continues to expand, and companies must now find specialists to write and re-write applications for each target platform. This in turn is leading to a shortage of programmers, higher costs to recruit and retain talented developers, and ever-increasing workloads for the development staff.

Lack of Architectural Flexibility Limits Options

While client-server is still the predominant computing model, many companies are moving some or all of their applications to two-tier, three-tier, or n-tier models that better meet the demands of Web-centric computing. These architectures vary widely in their capabilities, quality, and performance characteristics.

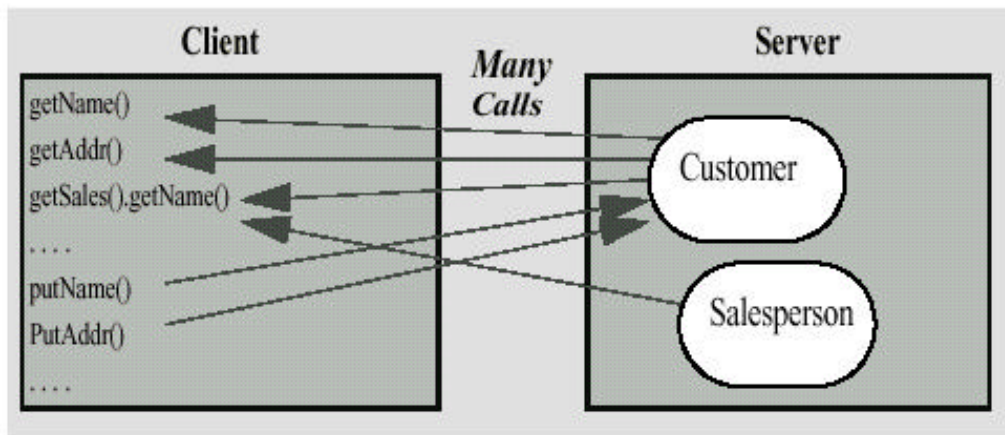
With traditional coding technologies, transitioning from one architecture to the next requires manually re-coding each application for each specific technology choice. And again, most companies simply cannot afford the time and expense of such transitions. The result: companies make technology choices that are tactically expedient rather than strategically advantageous; they do not adapt applications that have outlived their usefulness to customers; they do not scale successful applications to more appropriate platforms; and they do not pursue development projects that require an architectural transition.

Simply put, companies get locked into a software stack before coding begins, and they waste a substantial amount of time, money, and competitive opportunities trying to adapt later.

Badly Distributed Code Slows Application Performance

Most organizations today utilize some form of distributed computing, in which data and applications are distributed across the network on multiple servers, and specific pieces of functionality or data are called when needed by end-user clients. When a natural object-oriented coding style is used for client access to server data, however, application performance can bog down in a hurry because of the network traffic this model creates. For example, even a seemingly trivial segment of access code can require many network calls just to access a few fields from the

server. The resulting round-trip traffic can quickly degrade the performance and scalability of any enterprise application.



Inefficient code distribution results in multiple round-trip network calls, degrading application performance and scalability.

Optimization is a Slow, Expensive, Manual Process

It is possible to optimize applications to overcome the performance problems inherent in distributed object-oriented coding, but this requires the design of an optimized server API and the client proxies for each application. This task is much too difficult, time-consuming, and expensive to accomplish manually; yet with current development tools there is no alternative. The manual coding process is also error-prone, resulting in the need for time-consuming testing, debugging, and re-testing.

That is why IT departments do not optimize most applications today. Instead, they focus their resources on the most urgent development issues and try to catch up with an ever-increasing backlog of projects.

Ace: The Optimizing Compiler for the Network

Since its inception, Sun has proclaimed that “The Network Is The Computer.” While the computer industry was slow at first to understand and embrace this concept, the rise of client-server computing, the dot-com age, and Web services have borne out its merits. Ace technology—by improving developer productivity and time-to-market benefits, providing improved architectural agility, and optimizing application performance—adds a new dimension to Sun’s slogan. Simply put, Ace is the optimizing compiler for the network.

Simplifies the Development Process

The key breakthrough Ace brings to the development community is the ability to transform business requirements or specifications into automatically generated, correctly coded, smooth-running applications. The work of the Ace research team in this area has yielded more than 20 patent disclosures to date.

Ace is unique because it provides a natural way for developers to describe the “intent” of the application precisely, as opposed to manually writing the code that implements that intent. In other words, developers use Ace to create a high-level specification that provides enough information so that Ace can automatically generate the optimal code for the application. It completely separates the implementation details of a distributed application from its specification. This high-level specification consists of:

- ? A set of business objects that describe the entities, relationships, business rules, constraints, and procedures of the business. This collection of objects, called the **Business Object Specification (BOS)**, is typically shared across many applications.
- ? A set of task states and transitions that describe how the application and its user interacts with the underlying business objects, and a set of transactions that describe changes that the application makes to the values of these objects, called the **Application Usage Specification (AUS)**.

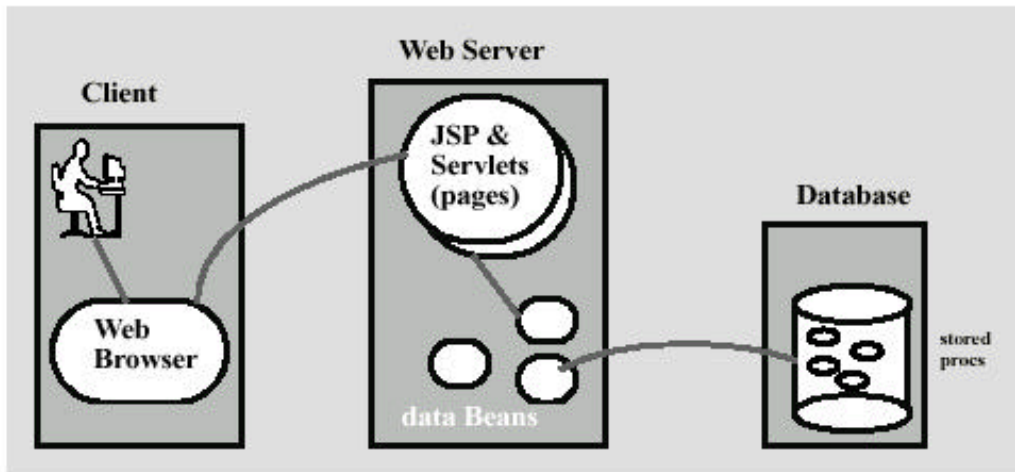
To better understand the roles of the BOS and AUS, consider an application that plays chess. The chess board, the chess pieces, the positions of those pieces on the board, and the legal moves that each piece may make would be described by the BOS. How the game itself is played, including how to set up the board, who goes first, and the conditions of winning the game would be described by the AUS.

The automatic generation of optimized code has huge implications for both developers and business executives:

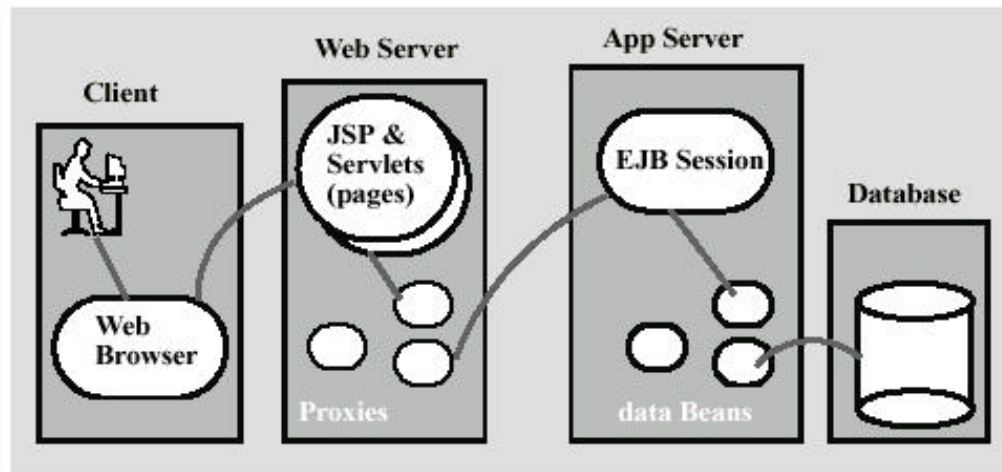
- ? Ace dramatically lowers the bar on the level of expertise that is required of application programmers. With Ace, the user is *not* required to learn or understand technologies such as J2EE, CORBA, or .NET, database schema design or management systems, relational query languages such as SQL, database APIs, application server environments, Web server environments, Web page programming, details of transactions and locking in a scalable environment, or how to transfer data from one type to another. That means businesses do not always have to hire specialists for every development project and every specific hardware platform.
- ? With Ace, application developers can now concentrate on the application semantics without having to worry about the details of distribution or the complex peculiarities of the software stack used to implement the application.
- ? Management can now understand and contribute to the application specification, so there is a tighter alignment between business goals and software capabilities.

Keeps Architecture Choices Open

The advent of Web-centric computing has given rise to new architectural models and new technology choices that companies want to take advantage of. For example, 2-tier models (client-Web server-database) may be best for small, lightly used applications, but companies might want to deploy a 3-tier architecture (client-Web server-application server-database) for complex, high-traffic applications and Web services.



While traditional coding techniques embed the architecture choice in the code, Ace's application specification has been very carefully designed to be high-level and architecture-independent. Therefore, developers can use Ace to regenerate the application's code for new architecture and technology choices, just like a conventional compiler can generate code for different processors, machines, and operating systems. For example, a developer could create an application initially for use in a 2-tier architecture because it would be much simpler to configure, install, and manage. As the volume of usage increases, the scalability of this architecture might limit performance. The developer could then use Ace to simply regenerate the application for use in a more powerful 3-tier architecture.



The architectural agility of Ace gives businesses the ability to “Write Once, Deploy Anywhere.” And this has multiple benefits:

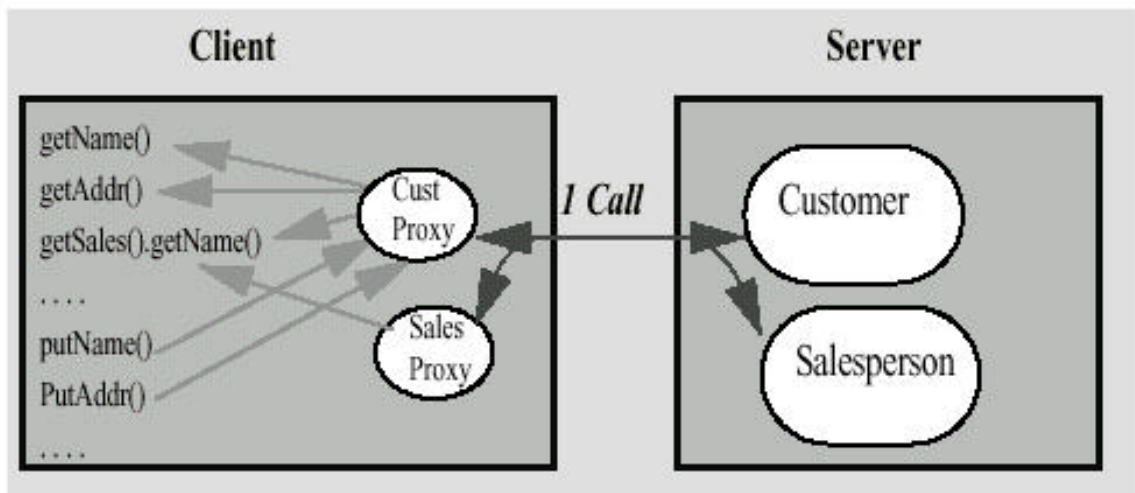
- ? Ace drives better architectural and technology choices because it gives companies the flexibility to evaluate more options and to take full advantage of what they learned from previous technology decisions
- ? Developers can use Ace to prototype enterprise applications, working collaboratively with management to optimize the design of the application without worrying about underlying platform or coding details

- ? Developers can design applications on the fly with full confidence that changes and enhancements can be implemented quickly and easily
- ? Enterprise applications become “future-ready,” so the IT department can begin thinking more strategically, less tactically

Optimizes Performance and Improves Scalability of Applications

The code that Ace generates is performance-optimized for a given architecture and technology choice. The key optimization is that Ace generates application components to minimize the network traffic that would otherwise create delays. For example, the components generated by Ace allow data retrieved from an application server to be cached and accessed on the Web server side as local proxy objects, so only one round-trip network call is required.

By dramatically improving application performance, Ace also increases application scalability. Servers running Ace-generated applications are capable of running many more applications than previously with the same resources.



Ace optimizes performance: In a 2-tier environment, Ace limits the network traffic between the application server and the Web server to one round trip using proxies.

Equally important, Ace can improve the performance and scalability of existing applications as well as provide optimized performance for new applications. Thus, programmers can reverse-engineer existing applications into Ace, making it easy to adopt different architectures (see below).

Example: Java Pet Store Application

The Java Pet Store is a reference application that provides typical e-commerce functionality: presenting users with various views of products and services for sale; taking and acknowledging orders; processing credit cards; and managing user logins, shipping information, and shopping

sessions. The Java Pet Store also includes administration functions, including inventory and order management. The application is also designed to support easy creation of interfaces to other systems, to leverage new technologies as they develop, and to provide business-to-business functionality.

Initially, the Java Pet Store application was created by Sun developers as a means of displaying the basic features and benefits of using J2EE technology and best practices in system design. The intent of the Java Pet Store was to cover as much of the platform as possible, as clearly as possible, in a relatively small application. The first implementation of the Java Pet Store contained 14,273 lines of code and took six months to develop.

Following the successful deployment of the Java Pet Store, Microsoft engineers decided to use the reference application to show off the capabilities of C# and .NET, and rewrote the application using 3,484 lines of hand-crafted code.

By contrast, a single developer has created an Ace specification of the Pet Store application consisting of just two files and only 224 lines of hand written code—less than one tenth of the size of the Microsoft version. The Ace Pet Store application can be deployed using either a 2-tier or a 3-tier architecture. Best of all, the new Ace specification took Sun's developer only a week to complete.

Summary: The Business Benefits of Ace

Ace technology has shown significant potential in solving some of the most vexing challenges IT departments face today. Using Ace, developers can:

- ? Simplify the initial process of building and deploying enterprise applications
- ? Migrate applications to new architectures and technologies without lengthy and expensive recoding cycles
- ? Improve performance and scalability without increasing cost or development timeframes

For businesses, the advantages of Ace go far beyond technical capabilities. Because of Ace, enterprises can now:

- ? Increase the number and the quality of applications under development without increasing the IT budget
- ? Keep up with constantly changing customer requirements
- ? Create competitive advantages and revenue streams by getting to market faster with new, high-quality, software-driven services
- ? Increase the productivity of the current development staff rather than continue recruiting, hiring, training, and re-training new developers
- ? Leverage current technology resources in migrating to more powerful, sophisticated architectural models
- ? Improve customer satisfaction and loyalty by improving the performance and scalability of enterprise applications

? Get business goals more tightly aligned with software capabilities—without requiring executives to become software programmers

Ace Road Map

Ace is a research project, not a product. The team has developed an implementation of the Ace technology that runs within the Forte/NetBeans Integrated Development Environment (IDE). It includes GUI editors for the two specification languages (BOS and AUS) as well as a way to view the specifications in a natural textual way and synchronously edit either the textual or GUI view. The team has also developed generators for both 2-tier and a 3-tier implementations of an application, and is able to produce working applications.

The next major milestone release of Ace technology will be usable by cutting-edge users who wish to try it out for prototyping applications. The Ace team is interested in receiving feedback on the current capabilities as well as capabilities that may need refinement or additional ease of use features.

The ultimate destination for Ace technology is a product division within Sun, where its proven features and capabilities can be further enhanced and made available to application developers globally.

For More Information

Ace was demonstrated in a Keynote speech with James Gosling at the JavaOne 2002 conference. The on-line video of this Ace demo can be found at the web site (<http://servlet.java.sun.com/javaone/sf2002/conf/keynotes/index.en.jsp>). Ace is shown in the Keynote webcast titled "*Monday Keynote - March 25, 9:30 AM - 11:30 AM*" and its demo presentation starts at about 1 hour 38 minutes into this webcast.

For additional details about Ace technology or examples of how Ace has been used at Sun, please contact the Ace team (ace-info@sun.com). An instructional guide for programmers entitled "*A Practical Guide to Application Specification Using the Ace Application Builder*" will be available through the Ace team.

About Sun Microsystems Labs

Sun Microsystems Laboratories was established in 1991 with the charter of keeping Sun at the leading edge of technology. Today, we've expanded from a small group of scientists into a separate division with a staff of almost 200 at research sites in California, Massachusetts, and Europe. Together, these centers take advantage of a tremendous pool of scientific and engineering talent and enable our people to collaborate with other researchers from a wide range

of industries. Sun Labs also works closely with major universities worldwide, jointly funding and staffing applied research and advanced development projects. In the years ahead, Sun Labs will continue to play a major role in the development of Sun, the computer industry, and the Net Economy.

About Sun Microsystems, Inc.

Since its inception in 1982, a singular vision, "The Network Is The Computer" has propelled Sun Microsystems, Inc. to its position as a leading provider of high-quality hardware, software, and services for establishing enterprise-wide intranets and expanding the power of the Internet. With more than \$18 billion in annual revenues, Sun can be found in more than 170 countries and on the World Wide Web at <http://www.sun.com>.

#

? 2002 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Java, J2EE, and EJB are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.