

Users' Choice IDEs - 2008

A comprehensive study of over 1,200 software developers

May 2008

Evans Data Corp, 740 Front St, Santa Cruz, CA 95060
www.evansdata.com (800) 831 3080

INSIDE THIS REPORT

- 1** Overview
- 4** Overall IDE User Satisfaction Rankings
- 6** Eclipse
- 8** IBM Rational Application Developer
- 11** My Eclipse
- 13** Delphi
- 15** JDeveloper
- 17** Adobe Macromedia Studio/Creative Suite
- 19** Microsoft Visual Studio
- 21** Net Beans
- 23** Sun Studio
- 25** Attributes

Overview

In the spring of 2008, Evans Data Corp conducted a worldwide survey of developers with the intent of finding user satisfaction with the features of various IDEs. This was the third year that we've conducted this user satisfaction survey. For 2008, over 1200 developers participated with a proportional spread across regions. The survey was conducted via email and was translated and posted in French, German, Portuguese, and simplified Chinese, as well as in English.

Only users of an IDE were asked to rank the features of each IDE. Thus we've restricted rankings to only those developers who are actually users of the environments.

The developers were asked to rank fifteen different features commonly found in IDEs. Each feature could be ranked as "excellent", "very good", "adequate", "needs improvement" or "NA". During processing, the first four rankings were assigned a numerical value according to their relative significance. The last value ("N/A") was discarded. Values were then combined to produce a score for each element for every IDE and also for an overall total score. The relative values for each element ranked are displayed in graphs with the same scale, so that comparisons between graphs can be made. The numbers in the graphs only display relative values.

IDEs that were user-ranked and had enough users to be included in the study were:

Adobe Creative Suite / Macromedia Studio
Codegear's Delphi
IBM Rational Application Developer
Microsoft Visual Studio
MyEclipse
NetBeans
Oracle JDeveloper
Sun Studio

Fifteen IDE features and capabilities were rated by the users. These were:

Quality of tech support
Size/quality of developer community
Documentation
App modeling tools
Web design tools
Sample apps
Profiler

Compiler performance
Performance of resulting apps
Debugger
Make/Build Functions
Editor
Ease of Use
Ability to integrate tools
Availability of 3rd party tools

The goal was to shed light on the different characteristics of each IDE and toolset and to show the strengths and weaknesses of each. The design philosophy behind integrated toolsets can differ significantly as vendors target different market segments and provide different capabilities focused on the needs of those segments. Some IDEs are aimed at large enterprise developers, while others are designed for individuals or small teams. Some are Open Source and grow and mature organically, while others have received the benefit of millions of dollars worth of research and development, in addition to market research and user testing.

This study shows how well each has done in pleasing users

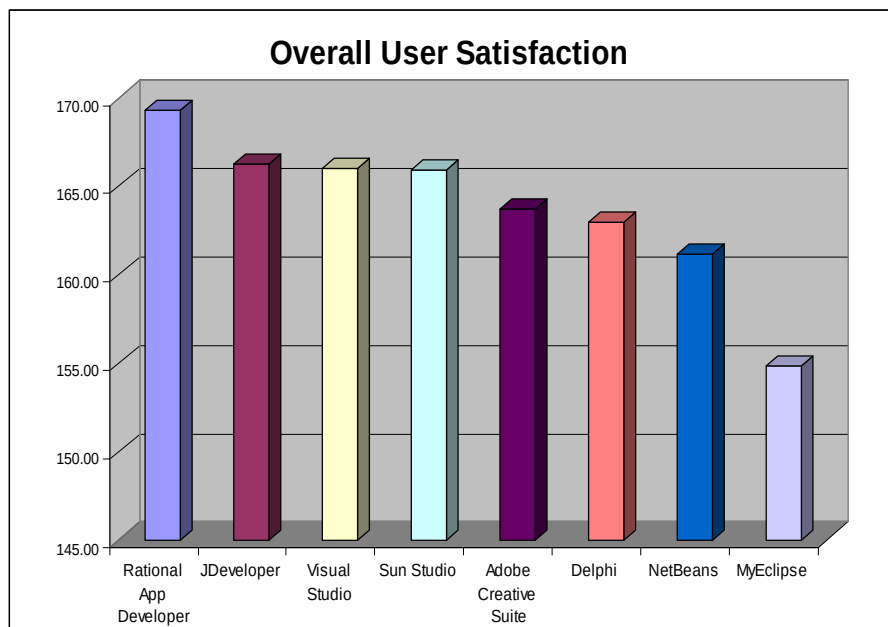
Overall IDE User Satisfaction Rankings

In the pages that follow we look at each individual IDE and how their features/functions were rated by their users, and then we look at each feature/function individually and see how each of the IDEs were ranked by their users relative to each other. In the graph at the bottom of this page we show how the total scores of each IDE are relative to each other.

In addition to asking users to rate the attributes of the IDE and toolset they use, we also asked all of the developers to rank the relative importance of each of the attributes. The ranking table for that can be seen later in the report in the chapter about Attributes. Using the importance rankings given by the developers for the attributes, the attributes were then weighted according to their relative importance. This allowed scores for each attribute to be tallied to give a total relative score, and that is shown below.

IBM's Rational Application Developer clearly got the best scores from their users. It's safe to say that RAD's users love their IDE and tools as well as the support and care they get from IBM. All of the IDEs in this survey are popular and very good tools, but RAD is just head and shoulders above them when it comes to user satisfaction.

"...RAD's users love their IDE and tools as well as the support and care they get from IBM"



The next three – JDeveloper, Visual Studio, and Sun Studio had very close relative scores, even though they have very different primary target markets and different strengths. JDeveloper is a sophisticated toolset optimized for Oracle environments, though capable of creating generic Java code. Sun Studio is Sun's high-end IDE and toolset designed for creating high performance applications in C/C++ or Fortran on Linux or Solaris/Open Solaris platforms, while Visual Studio is Microsoft's IDE for .Net and Windows and supports all of the Microsoft languages.

Adobe Creative Suite 3 is a comprehensive toolset aimed at web designers and developers and includes extensive video and graphics editing tools, just released last year and the successor to Macromedia Studio. Delphi, on the other hand is an IDE that goes back to the early 90's and was the first rapid application development tool. It has a solid and loyal base of developers. NetBeans and MyEclipse are the two open source IDEs in this report. Both ship pre-configured with a full set of tools.

All of these IDEs have their uses and their advantages. The next pages go into more detail on them and on what their users said about them.

**“All of these IDEs have
their uses and their
advantages.”**

Eclipse

“There are over 180 members in the Eclipse Foundation and there is a catalog of over 1000 plug-ins for the Eclipse platform.”

Eclipse is one of the most important technologies of our time, not just technically, but because of how it has reshaped the development tools landscape. Yet it is not the same type of product that the others in this study are, and to make direct comparisons would be comparing apples and oranges. The Eclipse technology is used in many major IDEs, including most of those in this survey. Thus to show Eclipse as a competitor would be misleading. For most IDEs, the very foundation that they're built on is Eclipse itself, and almost all of the companies that create IDEs or other tools have some presence in the Eclipse organization. Of course there are a few competitors such as NetBeans and Visual Studio, but even comparing these to Eclipse is not equivalent, as these are finished products that ship already configured, whereas Eclipse itself is basically a core platform on which extensions and plug-ins from any vendor can be added.

The Eclipse Foundation is a not for profit organization with two missions – the first is to build a platform that IDE vendors can use in their commercial products, and the second is to create an environment in which the community can build a strong tools ecosystem around that platform. In both of these missions, Eclipse has done extremely well, and has altered the world of tools and IDEs.

Eclipse began in 2001 as a research project by IBM to create an Open Source platform for building IDEs and tools. The goal was to make a platform framework and exemplary applications. The first tool was the Java IDE that has taken the developer world by storm.

In 2004, the Eclipse Foundation was formed as a not-for-profit organization and the realm of Eclipse solidified around that organization. Today there are over 180 members in the Eclipse Foundation and there is a catalog of over 1000 plug-ins for the Eclipse platform, though many insiders in the Eclipse organization think that number represents only a small portion of the plug-ins that are actually available. Eclipse is a phenomenon. It is wildly popular among Java developers and Open Source developers alike, and now that it supports languages other than Java, including C++ and now open scripting languages like PHP, we expect its popularity to swell even faster. Eclipse has been called the “killer app” of OSS development tools and its market share is increasing constantly. No doubt one of the reasons for its huge success is its business model as a core for plug-ins. These plug-ins are what has taken the industry by storm as more and more ISVs and OSS developers create and extend Eclipse extensions and plug-ins.

"Eclipse tools are becoming ubiquitous."

The latest big thing in the world of Eclipse plug-ins is Mylyn, a task-oriented interface for Eclipse. Introduced in the Spring of 2007, it reduces information overload and makes multi-tasking easier. Of course the Eclipse community produces hundreds of new tools for Eclipse each year. Eclipse supports a wide variety of languages, including Ruby and PHP in addition to C++ and of course, Java. But the bigger focus during the last year has been on runtimes rather than tools and the RCP platform is being widely promoted and used.

Eclipse's strengths are related to the quality and size of the development community as well as to a great availability of third-party tools and an innate ability to integrate them. These are the facets of the original design as well as functions of the supporting community.

The target market is extremely broad – as broad as the whole Open Source movement since tools are requisite to any software project. The Eclipse Foundation is shepherding over 90 open source projects, and there are over 1000 plug-ins available from Plug-in Central, the main distribution site for Eclipse plug-ins.

Since there are no corporate constraints concerning what type of plug-ins or tools will be created, Eclipse capabilities are extensive. Not just relegated to web projects, the Eclipse community has even been active in creating tools for parallel programming. The Parallel tools project initiated by the Los Alamos national labs and with participation from IBM have produced tools for very large parallel programs. In addition g eclipse, with participation from IBM has shipped which supports tools for grid computing. Eclipse tools are becoming ubiquitous.

Eclipse is open source and so is available free of charge. Tech support is provided by the community and commercial vendors. It is also the foundation for most of the other modern IDEs including almost all of the commercial products in this report.

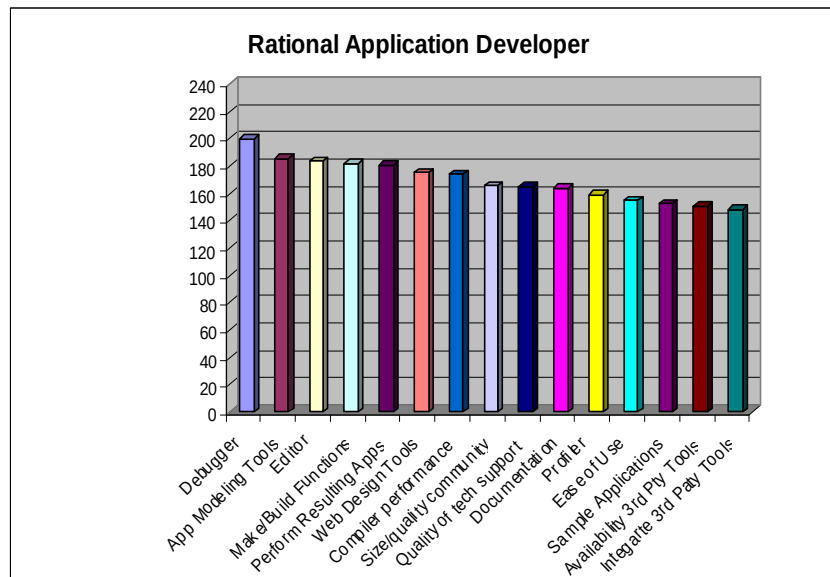
IBM Rational Application Developer

IBM's Rational Application Developer for Websphere is a complete IDE and toolset, based on the Eclipse open framework and optimized for the Websphere runtime. It helps Java developers create Java/Java Platform, Enterprise Edition (Java EE), portal, Web, Web services and service-oriented architecture (SOA) applications, and assists in rapidly designing, developing, assembling, testing and deploying these applications. It features visual tools that can be leveraged to reduce manual coding by abstracting the Java EE programming model.

"Rational Application Developer scored better than any of the other IDEs in user satisfaction for the second time..."

Rational Application Developer scored better than any of the other IDEs in user satisfaction for the second time in the three years that we have been conducting this study, and the reason is a strategic targeted market focus into the needs of the development community and a devotion to excellence in execution that has produced a superior IDE and toolset – as well as an entire family of integrated development products that runs from requirements management to project conception to development and implementation and beyond.

As a tool that is optimized for the Websphere runtime, Rational Application Developer is targeted at the Websphere user – usually found in large corporations, and this is a toolset that is powerful and complete enough to handle even the largest and most complex applications. It is tightly integrated with WebSphere Application Server and WebSphere Portal, and that integration works to help developers create powerful applications.



In June of 2007 Rational Application Developer v7.0.0.3 released tools in support of the WebSphere Feature Pack for Web Services. The Feature Pack for Web Services introduced a set of Web services standards that support interoperable and reliable Web services applications. This allows asynchronous messaging. In

“Having an integrated solution delivery capability is part of a larger and even more valuable solution for customers.”

addition, Policy sets, which are groupings of configurations of web service profiles, were introduced with the Feature Pack for Web Services. Policy Sets are used to associate different qualities of service with an application, and can be configured to allow only specific capabilities within a given WS-Interoperability (WS-I) profile.

More recently, Rational Application Developer released a v7.5 open beta. RAD v7.5 contains tools that support the new programming models standards support for Java EE5 (*EJB 3.0, JPA, JSF 1.2, JAX-WS 2.0, JSP 2.1, Servlet 2.5*) in line with WAS v7.0 (beta). These tools help support iterative development of Java EE5 applications, embracing an annotation based programming model style. Rational Application Developer's annotation support simplifies the code required to build applications and the knowledge required by the developer to understand the application.

Rational Application Developer 7.5 beta contains support for WebSphere test environments, WebSphere feature packs and WebSphere Portal Server. RAD v7.5 has enhanced its Web 2.0 support with Javascript, and DOJO editing to develop AJAX applications to extend Java EE 5 and SOA assets as REST style services. RAD is enhancing its support of J2C adapters and adding support for WebSphere Adapters (SAP, PeopleSoft, Siebel, Oracle E-Business suite, and JD Edwards).

RAD is an integral part of the overall Rational product portfolio which provides customers with an integrated solution for their end-to-end development lifecycle. This means being able to have traceability relationships of requirements to analysis and design elements, model Service Oriented Architectures with Rational Software Architect, share those assets with team members with Rational Asset Manager, and implement and test those services collaboratively with globally dispersed teams using RAD and Rational Team Concert. All of this while storing and managing work with ClearCase and ClearQuest.

IBM thinks of software development as a platform that should seamlessly integrate across all activities from conception through testing and beyond, and Rational Application Developer represents the core construction part of IBM's software delivery platform. When combined with robust modeling capabilities such as those found in Rational Software Architect, the right solution can be offered for the right role spanning across software analysts, architects, and developers. RAD has integrations with Rational ClearCase, ClearQuest, and RequisitePro as well as the ability to share Eclipse shell instances with all Eclipse based products that have a compatible Eclipse level

Having an integrated solution delivery capability is part of a larger and even more valuable solution for customers. When expanded to include business requirements capture, integrated business process analysis and modeling, and Enterprise Architecture, RAD becomes an important element of an overall Architecture Management discipline. Connecting EA and business processes with downstream technologies such as RAD that implement those models

“RAD also got the best score for quality of technical support...”

and processes is essential for making IT more responsive to the needs of the business.

Just like last time, Rational's users were impressed by the core tools which provide the foundation for any IDE and toolset – the debug capabilities, as well as by the integrated tools, such as modeling tools, editor, and build functions. Rational's modeling tools include the inimitable Rational Rose UML tools, and now also the next generation UML 2 based Rational Software Architect. These have been allowed to blossom under IBM's stewardship and are powerful dynamic modeling tools. User's also liked the compiler performance as well as the performance of the resulting applications, but a new twist this time is the users' appreciation for the new web design tools IBM ships with RAD. Right behind Adobe in this category, RAD's web tools include a Web Diagram Editor with visual application flows that supports Struts and JSF applications, a page designer to visually layout pages, a site designer, and AJAX support with a function rich JSF library. . RAD's WYSIWYG Page Designer incorporates a palette of web components (JSF, HTML tags, Form tags, and JSP tags). Portlet and portal pages can be quickly built using the portal page designer. A developer can quickly build the web UI visually and then associate it to a back end java bean, EJB, database table or Web service. RAD supports the creation of asynchronous, reliable, secure web services from Java beans or WSDL/WSIL files, as you type validation of annotations and the ability to define Policy sets for Quality of Service. RAD helps developers build applications or fully develop the application right down to generating client code for the application and deploying it to the target runtime.

RAD also got the best score for quality of technical support – not exactly a toolset feature, but certainly an important consideration in the purchase of any piece of software, and especially important for development tools, which require a higher level of expertise than ordinary software. IBM provides excellent 24 x 7 phone support for RAD through a front line support team who are experts in their fields. In addition, support teams work together across disciplines to solve problems, and IBM is known for providing support for multiple versions for years.

The list price for Rational Application Developer is \$4,240 USD per Authorized User license and it includes 1 year of support. Additionally, earlier this year, IBM announced a software support policy change. IBM now offers a minimum of 5 years of purchasable support from the release date of a version/release with an optional purchase of an additional 3 years of support. This allows customers to maximize their software support investment and control their upgrade strategy.

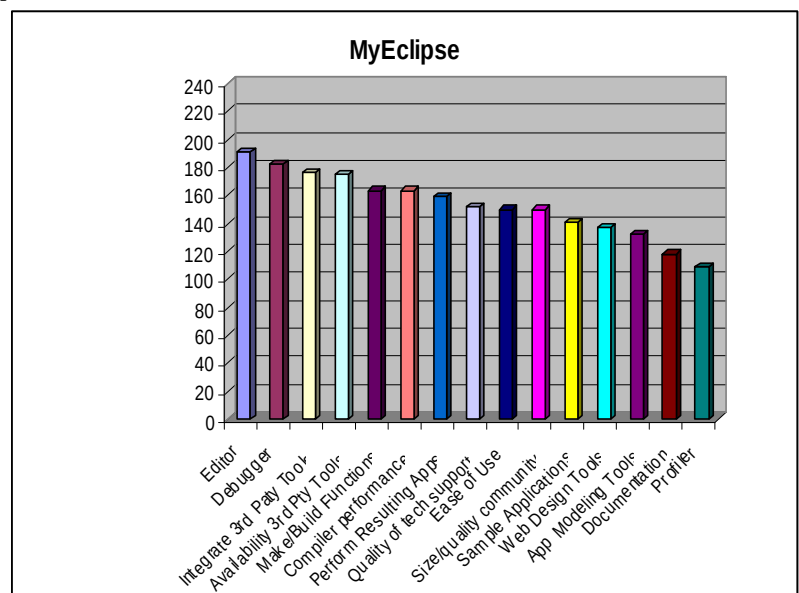
MyEclipse

MyEclipse is the Eclipse plugin-based IDE created and sold by Genuitec, a company that was started in 1997 with a focus on Java consulting. Genuitec's business model was to offer development services, and soon they had established themselves as a consultant for companies looking for help in designing, coding and implementing custom solutions based on Java.

"...they have continually added tools to create a very large toolset that includes Java EE/J2EE, database, RAD, UML, Reporting, JSF, Ajax and other tools."

MyEclipse was inspired by the company's experiences with clients who wanted to standardize on toolsets for Eclipse and who wanted customized Java implementations focused on their particular business needs. These implementations and the tools and/or components that Genuitec created for specific clients' needs soon began to form a large body of plug-ins for the Eclipse platform. Genuitec created the MyEclipse product based on those plug-ins. MyEclipse's presence in the market accelerated when Genuitec introduced the first JSP debugger for Eclipse. Since then they have continually added tools to create a very large toolset that includes Java EE/J2EE, database, RAD, UML, Reporting, JSF, Ajax and other tools.

The target market for MyEclipse is the corporate enterprise developer, the J2EE developer, and Ajax developers with their Ajax toolkit. In addition, Genuitec has recently released MyEclipse Blue Edition, which is optimized for WebSphere and is positioned as a replacement for Websphere Studio Application Developer, which IBM discontinued, and a competitor to Rational Application Developer for Websphere Studio. Genuitec also identifies IntelliJ as a competing product, as well as NetBeans.



Genuitec has also released updates in their reporting functions (a professional integration of BIRT), ICEfaces, Matisse Swing GUI designer from Net-Beans, and integration with their new Pulse delivery

Genuitec provides a strong support capability, with online rapid response to queries and usually gets problems addressed within a couple of hours. Indeed, users gave MyEclipse better marks for tech support than any of the other open source products. They were also very strong in the ability to integrate tools and the availability of third party tools. Genuitec has no salespeople on the payroll and instead has cultivated sales through the community in a viral, grass roots way, through word of mouth, and through cooperation with other Eclipse plug-in makers.

"Genuitec provides a strong support capability with online rapid response to queries and usually gets problems addressed within a couple of hours."

Though most sales are direct downloads, Genuitec also has some international distributors and now is also distributed with the new Pulse product. Pulse, which was released in November 2007, is a free service for the Eclipse ecosystem and is a key vendor distribution channel. It helps developers maintain the correct versions of their plug-in toolsets. Once installed, Pulse regularly checks your system to be sure you have the most up-to-date, configured software suite, and if there are updates available it notifies you and finds the connections to download updated software as quickly as possible. It also allows easy sharing of Eclipse tool stacks, eliminating version incompatibilities.

The standard edition of MyEclipse, which includes all the basic core Java enterprise development functions is available for \$29.95 for a one-year subscription for one developer. The subscription includes all updates released during the year, as well as email and forum support. The Professional version, which includes all of the standard tools plus the database tool suite, UML tools, Matisse, Ajax tools and others, sells for \$59.95. Blue Edition, with its advanced WebSphere-specific tooling is available at \$149.95 per year

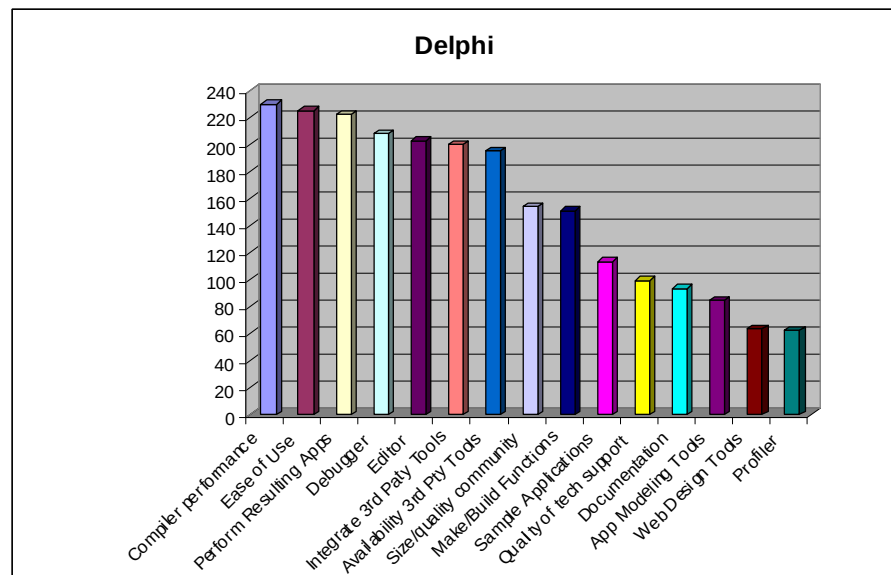
Delphi

CodeGear's Delphi has a long and illustrious past, starting in the mid-1990's when Anders Hejlsberg and his team at Borland created the product as an evolution of the Turbo Pascal compiler that was Borland's first product. This was during the time when Visual Basic and PowerBuilder were first introducing visual tools and easing development and the barriers to programming.

"Delphi now sports an enterprise framework, a reusable visual component library and a fully integrated suite of modeling tools."

Delphi was created to marry the high performance and object-oriented Turbo Pascal compiler with visual tools that let developers drag and drop components, "draw" windows, and so on. It was one of the first RAD (Rapid Application development) tools and ships with the Visual Component Library (VCL), which is a collection of re-usable component objects that can be incorporated into applications. VCL components are not limited to CodeGear's offerings, however and there is quite a thriving community of developers who create components for Delphi.

Delphi now sports an enterprise framework, a reusable visual component library, and a fully integrated suite of modeling tools. Its multi-language developer environment for Microsoft Windows supports Delphi (both Windows and .NET), as well as C and C++ applications within a single environment.



The current version, Delphi 2007 was released in September 2007, and includes several updates including improvements to the documentation, Windows Vista support with new Vista dialogs, VCL for the Web for building AJAX enabled Web 2.0 applications, new database development infrastructure and enhancements to the IDE itself. Delphi is targeted at Windows developers, and focuses on RAD development as well as database application development and integration. The VCL library is designed for rich GUI applications. CodeGear supports multiple languages and platforms with multiple products. Delphi 2007 for

Win32 is Windows only, and the CodeGear RAD Studio bundle also includes a .NET personality which can use the same code as native apps.

As with all CodeGear products, Delphi's users are fiercely loyal in most instances and rate their IDE and toolset highly. And as with other CodeGear products, developers praise the compiler itself – both for performance in compiling and for the performance of the resulting applications. Along those lines, Delphi has been making strides towards enhancing the ease of parallel programming for multi-core processors through threading tools and optimizations.

**“The Delphi
community is over 15
years old and is
extensive with over 500
tech partners.”**

The Delphi community is over 15 years old and is extensive with over 500 tech partners. Delphi supports its community through forums, newsgroups and peer to peer support.

Delphi's newest version, targeted for the last half of this year will feature full Unicode enablement additions to the Delphi language, and improved database tools among other features.

Pricing is set at \$899 per seat for the new user and \$399 for an upgrade to the latest Pro version from a prior version. The Enterprise version is priced at \$1,999 per seat with an upgrade price of \$1,299. An ongoing subscription is also available and includes both minor and major upgrades and three tech support incidents with support available via phone and online.

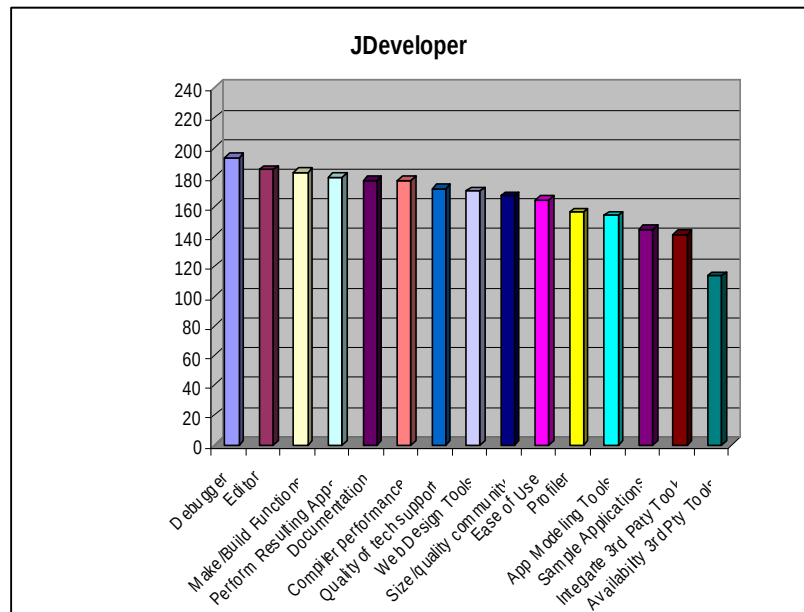
JDeveloper

“JDeveloper is targeted at enterprise developers building enterprise applications...”

JDeveloper began life as a licensed version of Borland's JBuilder ten years ago. However, in 2001 Oracle completely re-wrote the product from the ground up and today there is no JBuilder code left in it. In fact, Oracle spent lots of talent and effort on creating JDeveloper, not just as an IDE, but as an entire platform for the full Oracle family of applications; tooling, CRM, portal development, database, and application servers, and to very good result. Today it is a full featured IDE and includes database admin functions as well as SOA enabling tools, tools for BPEL, XML capabilities and a full spectrum of reusable components.

JDeveloper also ships with the Oracle Application Framework, which is designed to make development easier and more powerful through the use of visual objects and a very complete framework.

JDeveloper is targeted at enterprise developers building enterprise applications, including both the traditional Java/Java EE developer as well as SOA developers and particularly those from a 4GL background (using tools such as Oracle Forms, PowerBuilder and Visual Basic), as well as developers who are doing custom development to front the set of Oracle enterprise applications (E-



Business Suite, Siebel, Peoplesoft).

Included tools cover the full development life cycle from UML modeling through coding, testing, tuning, analyzing and deployment. In addition, they offer a highly visual approach to development with such features as WYSIWYG design of UIs (web and desktop), visual page flows diagrams for JSF and Struts, Modeling of Java Classes and EJBs, Visual editors for XML Schemas and WSDL. And their visual editors, property inspectors

“JDeveloper’s visual editors and page flows along with declarative wizards for creating business services make the IDE and toolset much more declarative in nature...”

and structure window are synchronized with the code editor offering a choice of development approach. JDeveloper’s visual editors and page flows along with the declarative wizards for creating business services make the IDE and toolset much more declarative in nature compared with traditional Java tools - and as such offers an easier transition and better productivity to enterprise developers. This is confirmed by the fact that its users ranked it so highly on a number of core development tools and functions. And while the profiler and application modeling tools were not the highest rated of JDeveloper’s features, they achieved the second highest scores of any of the IDEs in the study relative to the competition, as did the quality of tech support provided. JDeveloper’s users also gave the product the best ranks of any IDE for the quality of the documentation – an area that is often overlooked, but which is critical to the full use and understanding of a product.

The weakest area is in regards to the community. While JDeveloper is not limited to Oracle implementations and can create generic Java applications, the community of developers that creates third party plug-ins and tools for JDeveloper is not large. However, as noted elsewhere, this is only a disadvantage if the supplied toolset is lacking and that is obviously not the case with JDeveloper.

JDeveloper is a free development tool with no licensing fee.

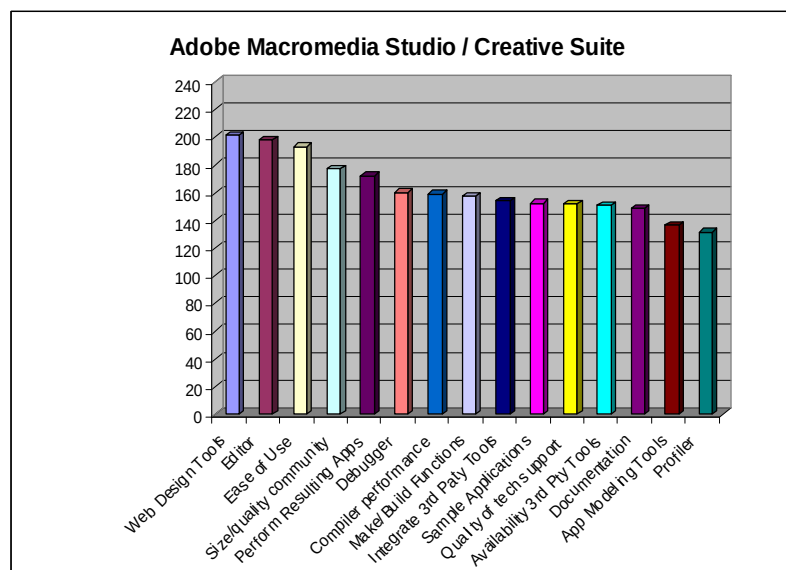
Adobe Macromedia Studio / Creative Suite

“Creative Suite 3 goes beyond any of the other products here in the breadth of its offering for web development...”

In April of 2007, Adobe transformed Macromedia Studio into Creative Suite 3. Before that, Macromedia Studio 8 was comprised of Dreamweaver 8, Flash 8, Flash 8 Video Converter, Fireworks 8, Contribute 3 and FlashPaper. Since this was a major name change, both Macromedia Studio and Creative Suite 3 were included in our survey questionnaire in order to capture data from those using either version. Next year, we will replace this combined entry with Creative Suite3, but at this point we didn't want to exclude users of Macromedia Studio 8 who may not have upgraded.

True to its nature and installed base, Creative Suite 3 is designed to help developers and even hobbyists create attractive and dynamic web pages and content. It offers advanced graphics, text, animation, video and audio tools, as well as the popular Flash technology for creating animation and interactive web content. It gets the best marks of any of the IDEs for web design tools, with close seconds for ease of use and size/quality of developer community, thus reinforcing its strong hold in that market.

Creative Suite 3 goes beyond any of the other products here in the breadth of its offering for web development including a host of other Adobe standalone products for video editing and processing, special effects, animation, and other graphics tools. Depending on which edition you have, you might find Dreamweaver, Fireworks, Flash, Photoshop, Illustrator, Encore, or several other well known Adobe products included. Thus the product provides a very robust toolset with a specific focus on graphics, video, and web design.



The developers' rankings of the CS3 features are telling of not just the product, but the entire programming ecosystem that has grown up around it and around the Macromedia products

before it. Designed for web designers and developers, those same developers appreciate the efficiency and effectiveness of the web design tools that are included. Similarly, the product gets good marks for ease of use and for the editor (which is almost always judged on ease of use). These all fit in with the design philosophy of the IDE itself, its target market, and its heritage and firm focus on web development. Adobe's product got the best scores of any IDE for web design tools.

**“Adobe’s product got
the best scores of any
IDE for web design
tools.”**

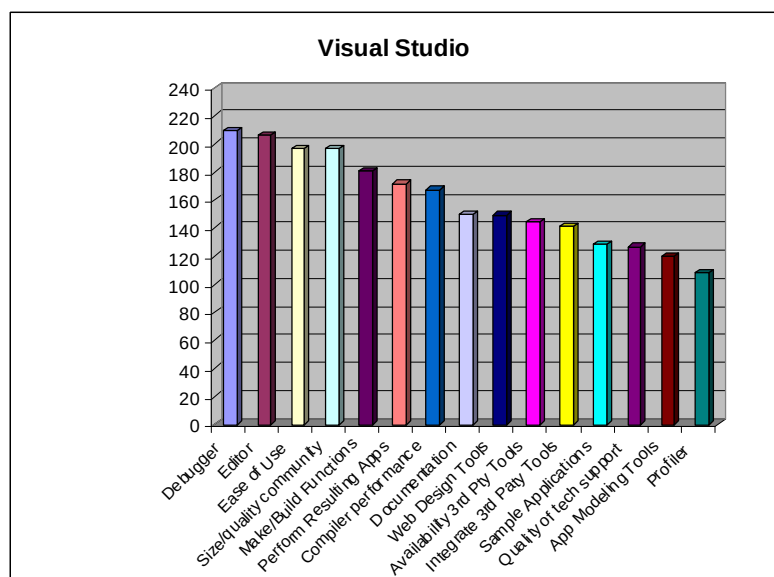
Microsoft Visual Studio

Visual Studio is the combination of Visual C++, Visual Basic, Visual C# and Visual J# – all in one package. Ten years ago these were all sold separately but the idea of an IDE that could be used across languages was a good one, as most developers use multiple languages. Visual Studio is a complete traditional IDE with a full toolset that can be used to create programs, websites, web applications, and Web Services that run on Microsoft Windows, PocketPC, Smartphones, and the World Wide Web.

“Microsoft Visual Studio is by far the most used IDE anywhere in the world...”

Microsoft Visual Studio is by far the most used IDE anywhere in the world with about twice as many users as any of its competitors. It is supported by the world's largest and best implemented developer relations program and it serves the world's largest installed base of platform users.

The latest version of Visual Studio (Visual Studio 2008) was released in November of 2007 along with the new .Net 3.5. While that is prior to when our survey was conducted, we cannot assume that all the users were reporting on that latest version. Most were likely using a previous version or a beta of the new Visual Studio 2008. The new version features; the ability to target multiple versions of .Net, built-in ASP.NET and Ajax support, a significantly improved HTML web designer (the same one that ships with Expression Web), with support for split-view editing, nested master pages, and improved CSS integration. It also contains LINQ (language integrated query) a feature that makes querying and working with data easier.



While the developers in this survey liked the editor and the debugger – both solid technologies with a long history

“It is impressive that such a powerful toolset with so many features and functions has managed to keep the essential ease of use it originally conceived.”

at Microsoft, one of the most highly rated aspects of Visual Studio was the size and quality of its developer community. These days Open Source development is famous for its communities and it's easy to forget that Microsoft commands the largest and best established developer community anywhere. That developer community means it's simple for Visual Studio developers to find helpful tips and tricks, online support groups, and even jobs. Another perhaps surprising thing about the results was that Visual Studio actually received very good marks for ease of use. Of course the entire “Visual” line of development tools, which stretches back to the advent of Visual Studio were based on the concept of wizards and making development tasks visual instead of text based, so thus easier. It impressive that such a powerful toolset with so many features and functions has managed to keep the essential ease of use it originally conceived.

Microsoft may wish to work on its profiler and modeling tools – both of which they've struggled with for years. In the case of modeling tools, Microsoft bought Visio several years back for their modeling tools, but obviously hasn't been able to assimilate them to turn that acquisition into an advantage.

Aside from the fact that Visual Studio produces code that's optimized for Windows and Windows is the largest target market anywhere, its main strengths are the quality of the toolset, and the extent and quality of its developer relations program. Microsoft has long understood the need to cultivate developers to write for their Windows platform and has gone to great lengths to make sure those developers are happy with their tools and their programs.

Visual Studio 2008 Professional edition is available for \$799 per user and the Standard edition is available for \$299 per user. Microsoft also provides volume licensing.

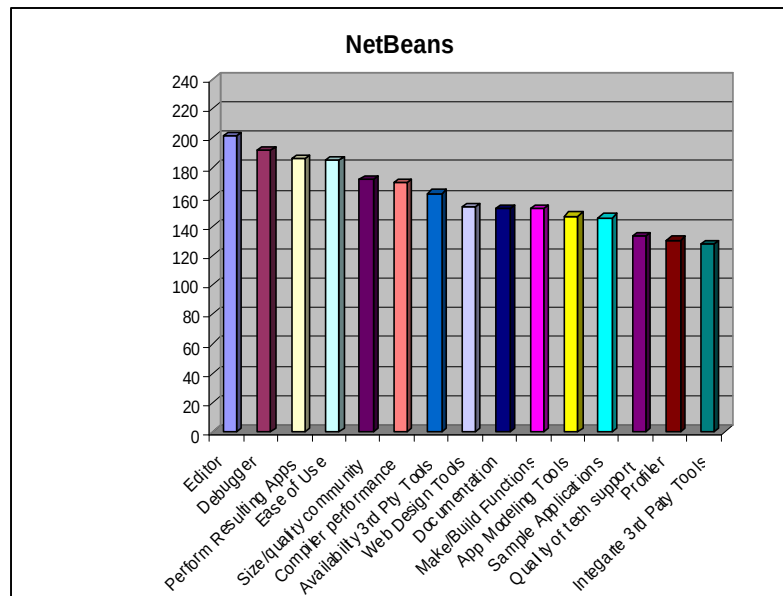
Net Beans

NetBeans began in 1997 as an IDE and platform for Java and it was bought by Sun Microsystems in 1999. Sun Open-Sourced the NetBeans IDE the following year. Since then, the NetBeans community has continued to grow. Applications are built on modules, and can be extended by adding new modules. Thus applications based on NetBeans can be easily extended by third-party developers. This is similar to the Eclipse model, and some say that Eclipse was IBM's answer to NetBeans.

However, NetBeans has had a life of its own and continues to thrive. After supporting NetBeans along with Sun Java Studio for the last few years, Sun has now decided to turn its attention squarely onto NetBeans as its primary Java tools focus and has discontinued Sun Java Studio (though Sun Studio for C still exists).

Today NetBeans IDE is an open source integrated development environment written in Java using the NetBeans Platform. NetBeans IDE supports development of all Java application types (J2SE, web, EJB and mobile applications). Among other features are an Ant-based make build system, version control, and refactoring.

“...version 6.0 of NetBeans was released which featured several improvements, including a crisp, responsive new editor and support for more dynamic languages.”



Last December version 6.0 of NetBeans was released which featured several improvements, including a crisp, responsive new editor and support for more dynamic languages. While we typically assume that users in these surveys are not likely to be using the latest version, in the case of NetBeans the users were either using or anticipating the new version, as the tools cited as improvements in version 6.0 match up with our users' experiences. Last year users were most impressed with the performance of the apps that resulted

“NetBeans’ greatest strengths are in its support for multiple language technologies...”

when using NetBeans, whereas this year the new editor, the debugger and make/build functions were much more appreciated.

In addition to the new tools, NetBeans has broadened its Java-centric focus to support more of the popular dynamic languages. Not only did release 6.0 add support for Ruby and JRuby, but at the Spring 2008 CommunityOne event, NetBeans announced the release of version 6.1, which includes (completely re-written) support for JavaScript and thus AJAX, speed and memory enhancements, and an early access release of NetBeans and new support for PHP. The NetBeans team has focused too long on Java only and now wants to be the favored IDE for all of the most popular dynamic scripting languages.

NetBeans’ greatest strengths are in its support for multiple language technologies, the powerful nature of the supporting toolset, and through its single “straight out of the box” installation which needs very little configuration. In addition, NetBeans includes a profiler and a popular Swing GUI builder – Matisse, which simplifies GUI development and automates a lot of the tasks that developers must complete.

NetBeans’ most obvious direct competitor is Eclipse. Eclipse has by far the larger user audience, which helps it to provide a greater range of support and toolset plug-ins. NetBeans, on the other hand is pre-configured for an easier installation and configuration experience, and Sun has put in a lot of work providing robust support for new dynamic languages.

NetBeans is mainly distributed from the NetBeans website, although Sun does provide NetBeans on media by request at its Sun Tech Days training events, as well as at various conferences conducted worldwide. As an open source IDE, NetBeans is free, and is supported mainly by the NetBeans community. However, technical support for NetBeans implementations can be purchased through Sun and support offerings range from as little as \$49 for a single incident up to plans for whole enterprises.

According to a Sun Spokesman, NetBeans is picking up momentum, with 8 million of the IDE’s 18 million downloads occurring during the past two years.

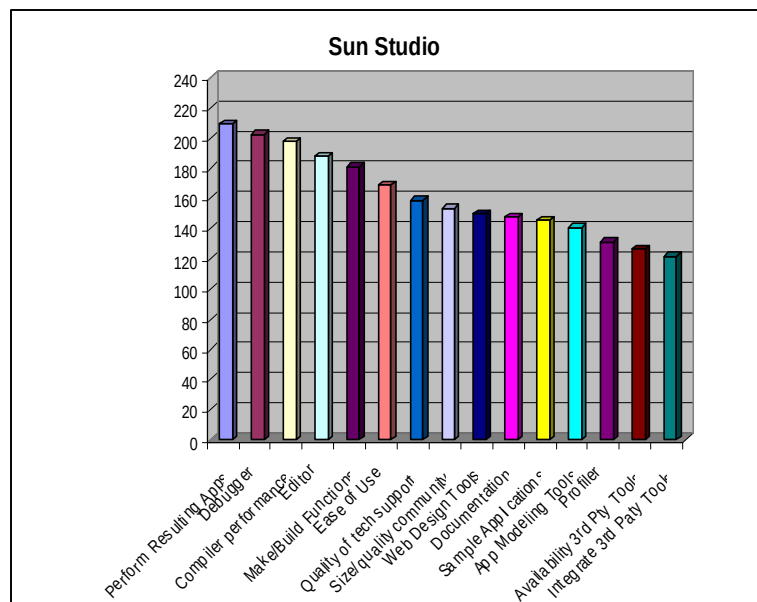
Sun Studio

“Sun Studio does an excellent job of producing the ultimate in applications optimized for performance.”

Sun Studio is Sun's high-end development system for C, C++ and Fortran. It supports Solaris, OpenSolaris, and the Linux platforms. Stressing performance, it is designed to optimize implementations on Sun's SPARC multi-core chips, as well as on dual and quad core processors from Intel and AMD. According to users, Sun Studio does an excellent job of producing the ultimate in applications optimized for performance.

Sun's traditional market for high-end servers includes the finance, telecom, corporate Fortune 500 enterprises, along with high performance computing, and Sun Studio, the toolchain of choice for Sun server installations, is equally focused on these areas.

The product provides a complete and polished user experience with a fully integrated toolchain: IDE, optimizing compilers, code/memory/thread debuggers, high-performance libraries, and more. In addition, it provides for a degree of cross-platform development across the three that it supports. But the capabilities that really set the product apart are those designed to increase performance and to enable parallel programming (auto-parallelizing compilers, thread debugging, thread profiling, thread analysis, OpenMP, etc.). These capabilities were enhanced further this year with additional performance improvements for Intel and AMD architectures. Sun Studio now claims to be competitive even with compilers from Intel and AMD and has recently set benchmark world records on all processors the software supports.



The preview version of the next release includes their Dynamic Tracing (Dtrace) tooling, an upgrade of the IDE to newest NetBeans version, on which the it is based, and other improvements.

Sun sees the Sun Studio competitors as mainly those open source tools, such as GCC and GDB, as well as the commercial Intel Compiler Collection (ICC). While, Sun Studio is a free license, it is not completely open source, and does not have the same type of community of developers that other IDEs in this survey do. The two entries relating to integrating third party tools are justifiably Sun Studio's weakest point according to users, though this is only a disadvantage if the toolset that accompanies the IDE is weak, and Sun's is definitely strong – especially when performance is a key issue.

“Sun Studio's focus on parallel programming and the complete experience of its IDE are surely targeted for in the future.”

As multi-core processing becomes more mainstream developers are going to have to adapt their new and existing applications for multi-threading to take advantage of the multiple cores. Sun Studio currently leads the way amongst these IDEs in a focus on that space. Sun Studio's auto-parallelizing compilers can take existing single thread apps and recompile them into multi-threaded applications-multi-thread code written as a single thread. It also includes thread debugging; the ability to follow execution and debug state across multiple threads; thread profiling, performance analysis of threads: thread analysis tools to identify race and deadlock conditions, OpenMP, visual, clock-profiling, hardware counters, and more.

While Sun Studio's main competition today may be implementations of the non-integrated tools made popular and used by programmers twenty years ago, Sun Studio's focus on parallel programming and the complete experience of its IDE are surely targeted for in the future.

Attributes

“Obviously, some capabilities are more important to the job of development than to others...”

The goal of this study was to examine how developers rank the IDEs that they personally use for a variety of features or capabilities. Only users of the IDE were asked to rank the functionality of the IDE, as they were the only ones qualified to do so. The attributes that they were asked to rank are shown below in the table arranged in order of their relative importance, as assigned by the developers.

Obviously, some capabilities are more important to the job of development than others, so the developers were asked to rank not only the features of the IDEs they use, but also the importance of the attributes, themselves. In this table, these attributes have been assigned a relative ranking based on how important the developers felt each one was.

Importance	Relative ranking
Debugger	256.8
Ease of Use	237.94
Performance of the resulting Apps	232.16
Editor	217.06
Make / Build functions	200.7
Documentation	188.81
Ability to Integrate 3rd party tools	180.64
Compiler performance	174.75
Size/Quality of the Community	141.52
Quality of Tech support	127.29
Web design tools	124.65
App modeling tools	102.69
Profiler	99.8
Sample apps	90.18
Availability of 3rd party tools	89.2

Debuggers are critical to development as all applications have to be debugged, and consequently vendors have spent a lot of energy trying to make the debug process easier and more effective. The other attributes from ease of use to editor also appear to be intuitively critical. The editor is the tool wherein the developer spends most of his time. Ease of use and performance of the resulting apps described the development process as well as its result, and thus are integral to developing apps in an IDE.

Very little has changed in the relative importance assigned to these attributes from last year. None of the most important categories have changed places relative to each other. At the lower end of the scale, where less importance has been assigned, the availability of third party tools has decreased in value rather significantly – perhaps because it's become more popular to buy complete “out of the box” solutions during the last

year. What used to be the standard 15 years ago appears to be making a return.

Other than that, sample apps have slightly decreased in importance and profilers have similarly increased. With parallel programming looming upon many developers, profiler and other code analysis tools are going to become more important.

**“Very little has
changed in the
relative importance
assigned to these
attributes from last
year.”**

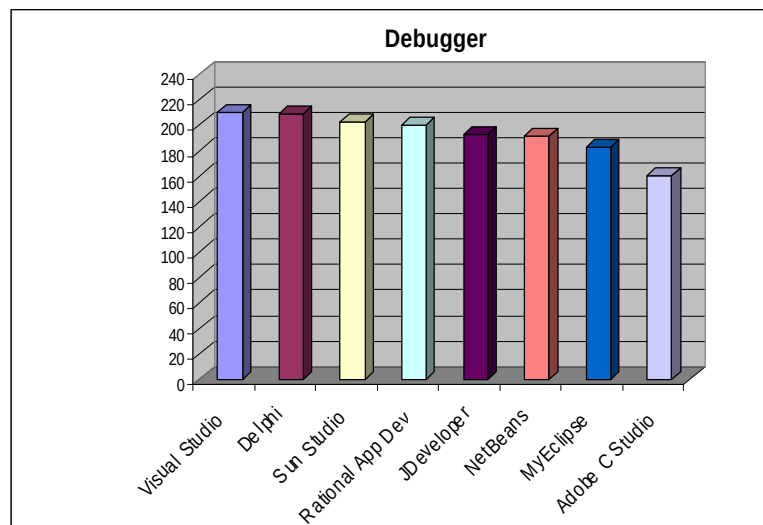
Debugger

This chart shows the relative rankings that users of each IDE gave to the debugger that ships with it or those that are available for it. IBM's Rational Application Developer scored better than the others followed by Visual Studio and Delphi which were very close together – in fact, Visual Studio, was virtually tied with Codegear's Delphi. However, all of the scores are relatively high. What this shows is that debuggers as a technology are fairly well advanced and most developers are relatively satisfied.

“Debuggers are a critical tool in an IDE's toolset.”

Debuggers are a critical tool in any IDE's toolset, and rightly, developers think of them as one of the most important. All programs must be debugged, and this is usually an iterative process, meaning that developers spend quite a lot of time in this activity.

The debuggers that ship with most IDEs are source-level or symbolic debuggers. At its most basic function, the debugger runs with the applications and stops on the problem line when the program crashes, thus making it easy for the developer to identify and fix the problem. Most debuggers that ship with these IDEs, however, are much more sophisticated than that and include features such as the ability to single step through each line of code, set breakpoints in the code, set conditional breakpoints and other capabilities.



As noted, developers gave the highest ratings to Microsoft's Visual Studio debugger as they did last year. Microsoft's powerful debugging tools have a heritage that dates back to the CodeView debugger of the early nineties. One of the most popular debuggers of all time, CodeView's descendants show the excellence of its line with full featured debugging with full symbol and source integration, remote machine debugging, multiple watch windows, threads, call stack, and modules. Microsoft's Visual Studio debugger supports both as well as Windows.

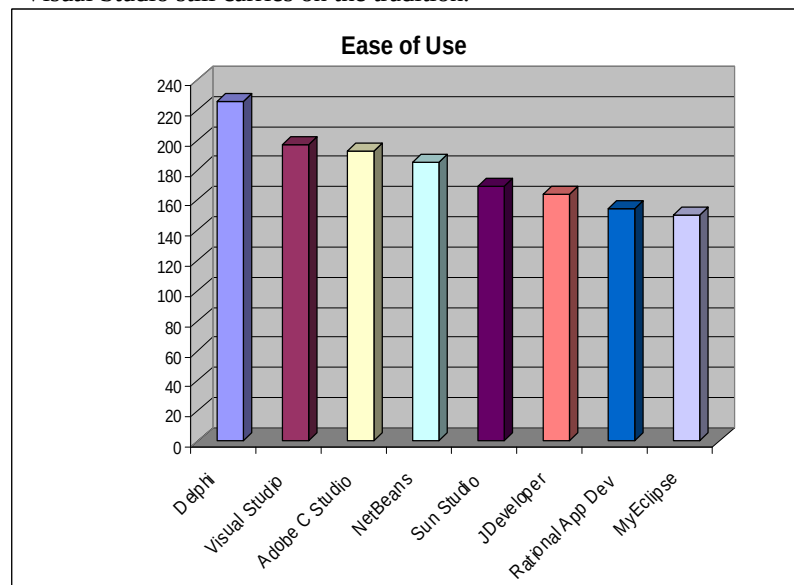
Ease of Use

“Visual Studio’s roots go back to the ground-breaking Visual Basic...”

Although ease of use is a highly subjective quality and can vary from one individual to another, not to mention from country to country or culture to culture, it is nonetheless universally understood and universally important. That said, one of the major reasons for the existence of IDEs is ease of use. It’s quite possible to write programs using unrelated toolsets, it’s just a lot more time consuming, tedious, and error prone. Integrated Development Environments provide the platform for tool interaction with the goal of orchestrating tool use in a harmonious and efficient way.

As one of the first visual development tools and the first Rapid Application Development (RAD) tool, Delphi charted the course and pioneered ease of use in development tools. This year, just like last year, the loyal Delphi users felt the original RAD tool is still the easiest to use. Delphi users are loyal verging on fanatical and their IDE always does well in user satisfaction surveys.

Visual Studio’s roots go back to the ground-breaking Visual Basic, an entirely new concept in development introduced in the early nineties with a goal of making programming accessible even to non-programmers. Although secretaries and accountants didn’t actually take up coding with it, Visual Basic did usher in the new era of visual tools and redefined ease of use in development. Visual Studio still carries on the tradition.



Adobe’s Macromedia Studio / Creative Studio has always been focused on web developers, designers, and others who may not have as much experience with coding. Creative Suite, released a year ago has raised the bar on web development both in ease of use and in the powerful features it delivers to users.

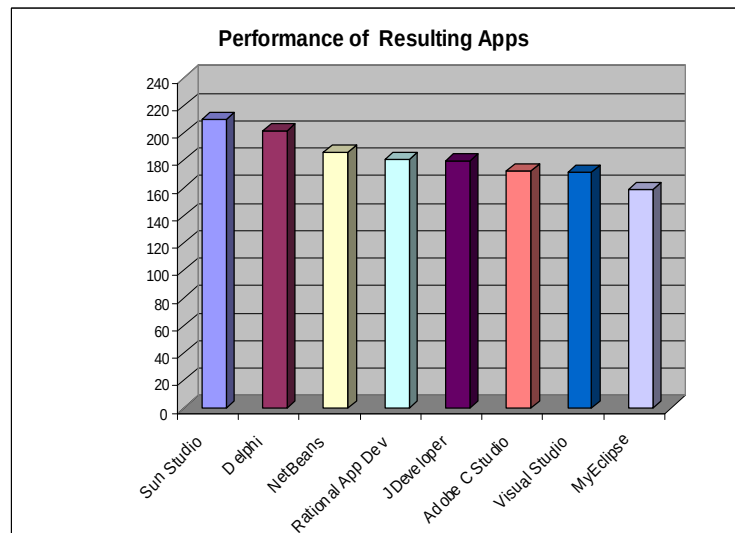
Performance of Resulting Applications

When asked which is more important, compiler performance or the performance of the resulting compiled applications, developers invariably reply that they care more about how fast the applications they create are than how much time they spend waiting for the compile. There's good reason for that too, as the performance of the resulting application will determine to some degree how successful that application is in the market.

"Sun Studio, coming from a high performance computing heritage, offers extensive optimization options"

The speed of the resulting apps is facilitated either by associated tools, like profilers, or by the compiler or interpreter. In the case of interpreted apps, the speed of the runtime is directly related to the speed of the resulting apps.

Generally, compiler optimizations are focused on improving performance by avoiding redundancy, inlining code and eliminating jumps where possible, trimming unnecessary code, and by locating those pieces of code which are executed close together in time close together in memory. However, today developers are under a lot of pressure to write multi-threaded apps that can take advantage of the new generation of multi-core processors on the market.



Sun Studio, coming from a high performance computing heritage, offers extensive optimization options including optimizations for parallelization and vectorization, as well as more common optimizations like dead code elimination, interprocedural analysis, and loop optimizations. Last year NetBeans got the best score in this category, but Sun Studio wasn't included in the survey. However, Sun Studio is built on the NetBeans technology and so shares the same aptitude for creating applications that turn in superior performance.

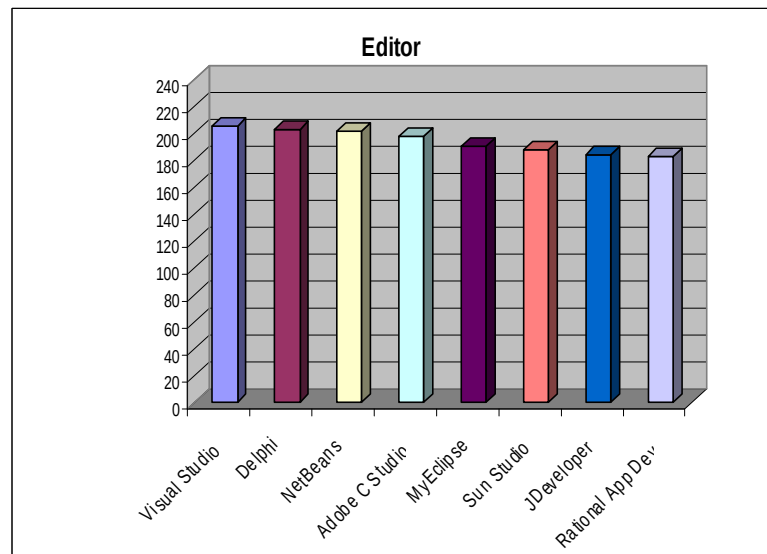
Editor

Of all the tools that ordinarily accompany an IDE, there is none that inspire as much loyalty or lack thereof than the editor. Why is this? The answer is obvious. The editor is the tool that developers spend the most time with – though a few might argue that they spend as much time with their debugger. Developers become very attached to their editor, so much so that they will often integrate their favorite editor into an IDE rather than use the one supplied.

“As with debuggers, the relative scores for all of the products were high, suggesting that developers mostly like the editors they use”

Any text editor can be used to create code. However, source code editors have special features built in to help with the editing of source code such as color syntax highlighting or auto complete mode. These features are often designed to be able to run a debugger or compiler and streamline switching between them.

Microsoft's Visual Studio editor is integrated in and supports the major text editing functions expected in a code editor. It also supports autocompletion for symbol names and variable names, and also supports code refactoring for “cleaning up” code. Delphi's editor is also popular with the Delphi users, and achieved very similar rankings as the Visual Studio editor.



In fact there is not much range in these scores, and this is especially true at the upper end of the scale. Also, as with debuggers, the relative scores for all of the products were high, suggesting that developers mostly like the editors they use – a fact that seems intuitive considering that the developer spends so much time with his editor, and it is quite easy to switch out editors if there is dissatisfaction.

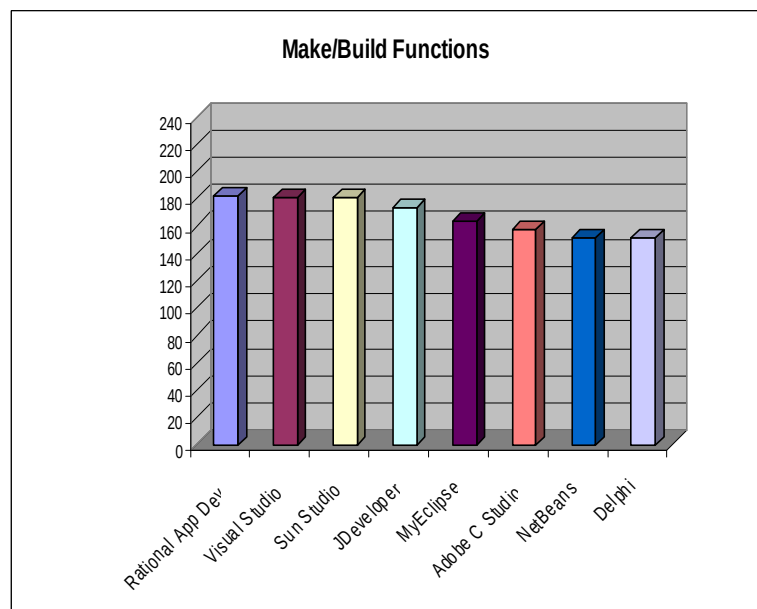
Make/Build Functions

The original make file is a utility that automates the process of converting files from one form to another, doing dependency tracking and invoking external programs to do additional work as needed. It was originally used with UNIX and it and its variations are very widespread. Developers have been using make for a long time in the process of building their applications. Make-like tools are inherently shell-based – they evaluate a set of dependencies, and then execute commands not unlike what you would issue in a shell. Thus vendors can easily extend these tools.

“Build automation tools are available commercially and as Open Source tools and most can be integrated into IDEs or certainly used alone.”

Build automation tools are available commercially and as Open Source tools and most can be integrated into IDEs or certainly used alone. ANT, the Open Source build tool from the Apache Project is especially popular, as is Maven.

Here we see very little difference in the scores of the leaders, but a fairly significant drop between the top three IDEs and NetBeans and Delphi.



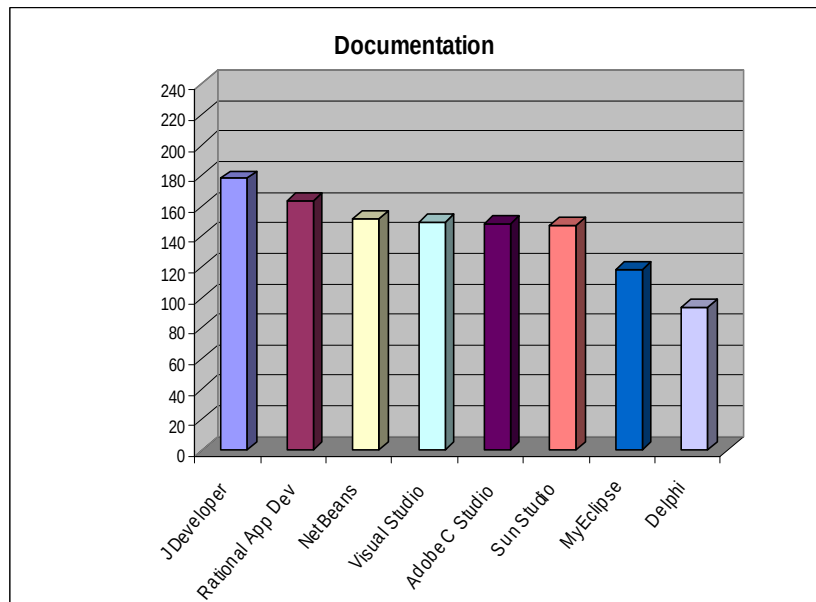
Documentation

Documentation is just one of the things that have been upended by the Open Source world. While there are plenty of developers who will spend long hours of their own time working on the code for an Open Source projects, there are far fewer who would like to spend any time creating documentation for a project or a plug-in.

“Documentation, or the lack of it, has always been a favorite topic of complaint for IDEs.”

Documentation, or the lack of it, has always been a favorite topic of complaint for IDEs and for any other software product. Logically there is no reason why the documentation should always be such a problem, but it nonetheless usually is. The reason for this may be more human than technical. Everyone is exposed to the help system and everyone has an opinion about how things should be said and how they should be presented. With an extensive help system there is plenty of room to present information in a way that not everyone will like.

Note that none of the IDEs in this study were rated very high on documentation, although many of them were rated better than last year. All of the graphs in this study have been drawn with the same scale so that the overall difference can be seen.



Comparatively, JDeveloper did very well, as did Rational Application Developer. The four IDEs in the middle were virtually identical in scores, whereas both MyEclipse and Delphi had significantly less user satisfaction – just as they did last year.

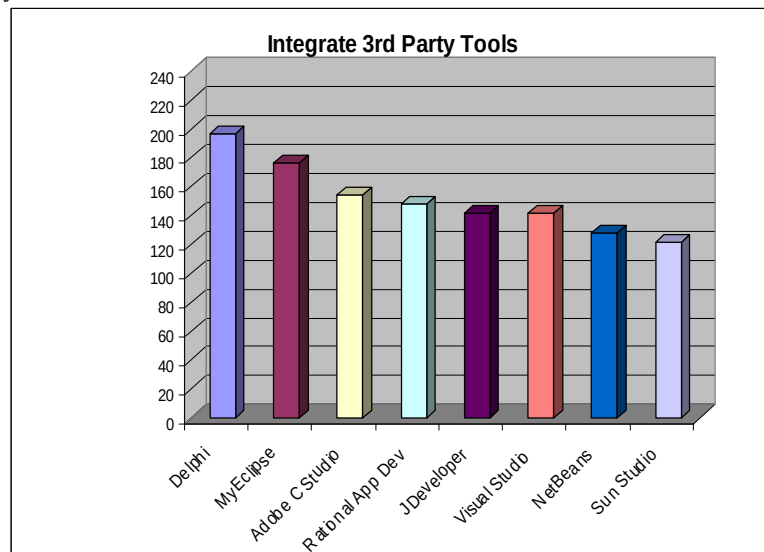
Ability to Integrate Third Party Tools

Before Eclipse, tool and IDE makers spent resources on developing partners who could and would create add-on tools which could be integrated into various IDEs. Eclipse turned that model on its head by providing the core IDE and encouraging a community of developers to form who would create plug-ins of every conceivable variety as extensions to the core Eclipse IDE.

Consequently, the importance of a developer community to create plug-ins and the ability of the IDE to integrate them has become much more important.

Delphi has been growing a loyal developer base for many years and has evolved with a clear focus on cultivating that community and encouraging add-ons. Consequently, it retains a reputation amongst its users for easily integrating third party tools, and like last year, this is reflected in the users' scores.

"Delphi has been growing a loyal developer base for many years and has evolved with a clear focus on cultivating that community and encouraging add-ons."



MyEclipse, based on the Eclipse core, is also viewed as very easy to integrate with, which is not surprising considering that it is actually multiple tools that are already integrated, and that its origin comes from integration with other tools. The surprising thing here is that the NetBeans users didn't rate it higher in this category. Just like Eclipse, NetBeans has fostered a large community and encourages add-ons and plug-ins. It does, however come fully configured for a straight "out of the box" user experience, but this shouldn't give users the feeling it is not easy to integrate with.

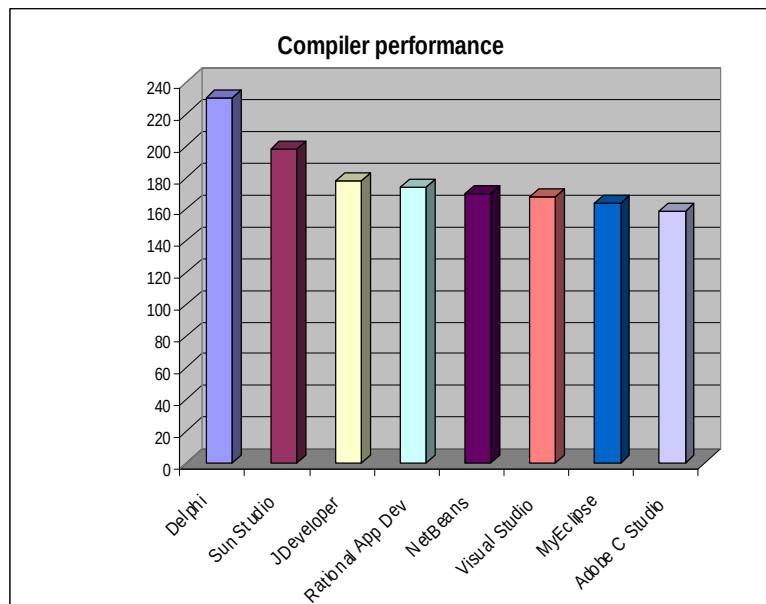
Compiler Performance

Speed of execution for most applications is a variable that effects user satisfaction more than anything else. Reliability and accuracy may be near perfect, but when the user has to wait, it becomes inconvenient. Software programs execute with varying degrees of performance based on several factors including the hardware and operating system they're running on, the optimizations that were used when they were created, and the basic way in which they were created and coded.

"Since the first turbo Pascal product, Borland released to begin its life as a software company back in the eighties, Borland has focused on providing great performance."

Compilers are software programs, and as such are subject to the same variability in performance as other programs. The difference between compilers and other programs is that compilers are often asked to compile extremely long applications and sometimes millions of lines of code, and so they may need to be able to execute for a relatively long time. Thus performance becomes much more important for compilers than it might for other types of programs. If the difference in compile time is a matter of hours, it may become more than an inconvenience.

Since the first Turbo Pascal product Borland released to begin its life as a software company back in the eighties, Borland has focused on providing great performance. Anders Hejlsberg, who wrote the original Turbo Pascal also wrote Delphi, and the same fast and efficient code is still providing its users with excellent performance, according to these Delphi users.



Sun Studio is aimed squarely at computing in the large enterprise where multiple processors are common and where high performance is required. Sun has created a fast compiler along with optimizations that make the resulting applications also run faster.

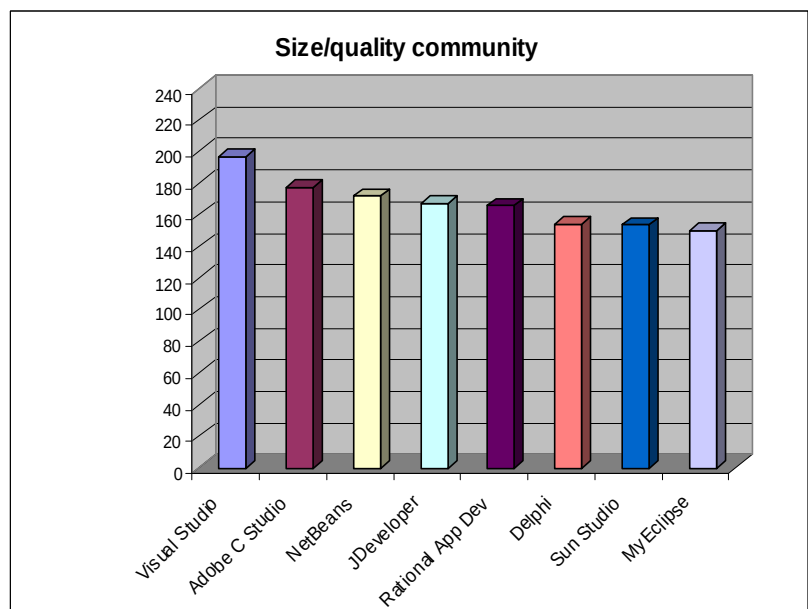
Size and Quality of Developer Community

“Adobe has been making great strides in the last few years to broaden their appeal and their reach...”

Visual Studio is the most widely used IDE and toolset in the world and it has been for many years. Because of the dominance of the Windows operating system and the technical optimizations for Windows and .Net in Visual Studio, this is the IDE of choice for more than half the world's developers. From the early days of DOS, Microsoft understood the strategic value of getting developers to write to their operating system. They developed MSDN, the world's largest and most respected developer relations program, and with Windows they have dominated development for the last fifteen years. More developers write for Microsoft platforms than for any other and this means there is a huge and active developer community.

Adobe has been making great strides in the last few years to broaden their appeal and their reach from web designers and developers into conventional development circles, and this is working. Adobe's Creative Studio builds on the past successes of Macromedia Studio and Dreamweaver – products that produced huge bases of users. In addition, Adobe provides a developer program, a Flash developer program and a partner site to share resources.

NetBeans is the other star performer in community development. According to a Sun spokesman, NetBeans has had over 18 million downloads with 8 million occurring in the last two years. In addition, Sun goes to considerable trouble to nurture the community by providing forums and training, conferences and Sun Tech days all over the world.



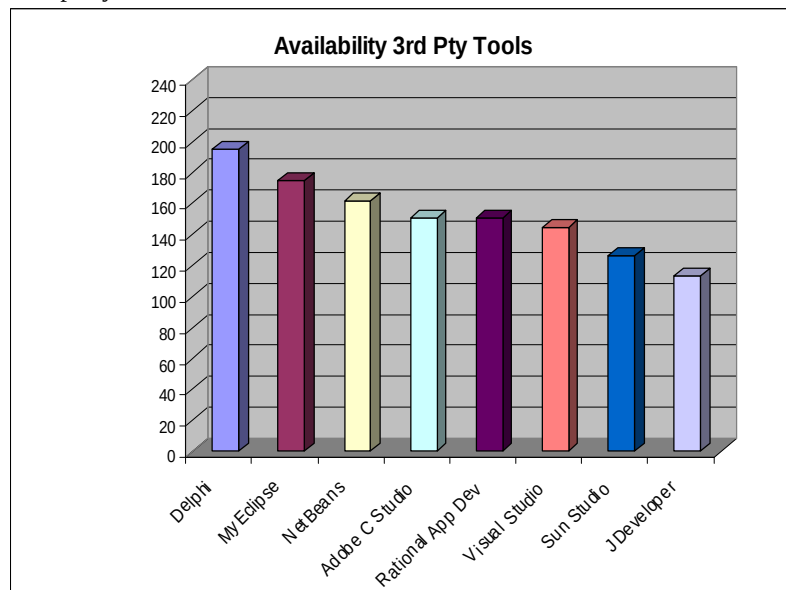
Availability of Third Party Tools

The availability of third party tools is largely related to the ability to integrate tools and the size and quality of the developer community. The larger the community and the more well fostered, the more third party tools will spring up.

As noted before, Delphi has a large and very dedicated user base that has been developing third party add-ins for more than ten years. With such a community in place, it's no wonder that Delphi is strongly supported with a lot of third party tools, and also no wonder that the Delphi users are so keenly aware of this.

On the other hand, Eclipse is a community as much as a technology, and MyEclipse is firmly in the middle of that community of companies and individuals who create third party tools for the Eclipse platform. Naturally, it is perceived of as having a strong availability of third-party tools.

“NetBeans rounds out the top three. This IDE has made the most positive gains in this area during the past year. “



NetBeans rounds out the top three. This IDE has made the most positive gains in this area during the past year. The NetBeans community and thus the availability of third party tools is active and growing. Sun has done much over the past year to develop that third-party community and is now seeing the results

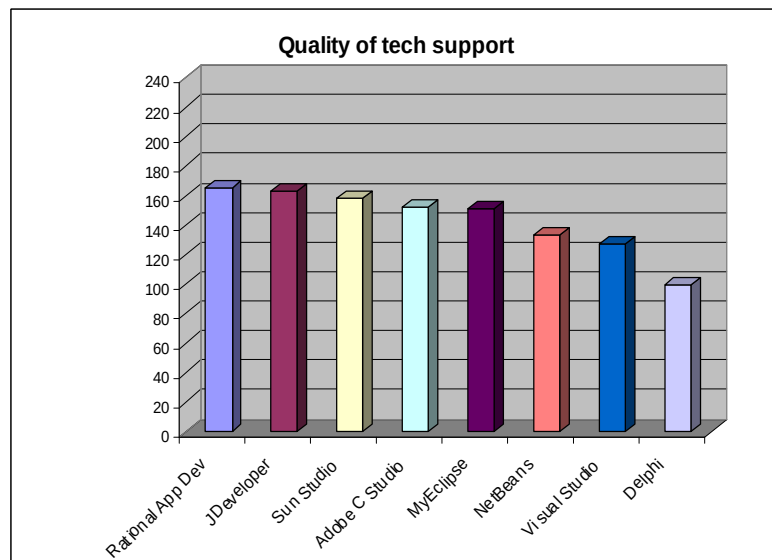
Quality of Text Support

The first thing to notice about this chart is that, while there are some differences in the perception of users between the various IDEs, all of them were ranked relatively poorly when compared to other attributes. This is most likely due to the inherently unhappy nature of technical support. It is only needed when there's a problem and getting answers is often a long and frustrating process.

"IBM provides excellent 24 x 7 phone support for RAD through a front line support team who are experts in their fields..."

IBM's users were the happiest of the respondents, followed shortly thereafter by Oracle. IBM provides excellent 24 x 7 phone support for RAD through a front line support team who are experts in their fields. In addition, support teams work together across disciplines to solve problems, and IBM is known for providing support for multiple versions for years. IBM now offers a minimum of 5 years of purchasable support from the release date of a version/release with an optional purchase of an additional 3 years of support. This allows customers to maximize their software support investment and control their upgrade strategy.

Oracle, which got the nod for best technical support last year, was also viewed by users as providing solid support options and execution.



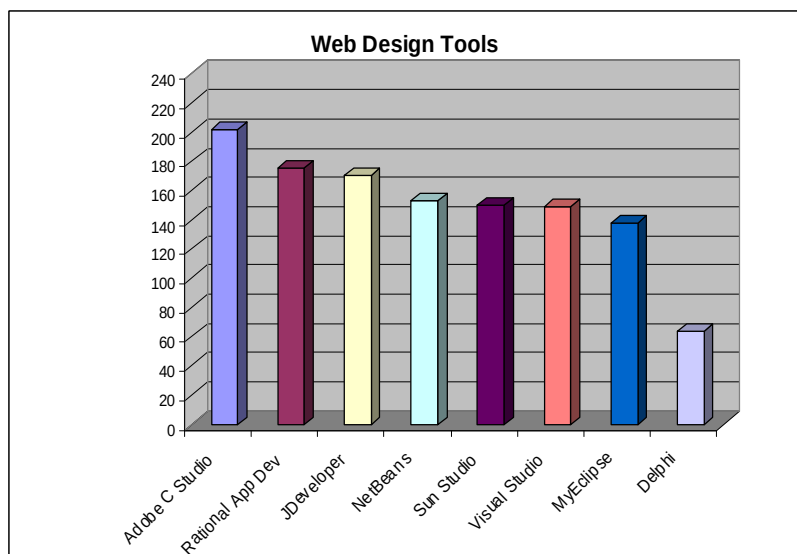
Web Design Tools

“It is no surprise that developers using Adobe’s products are extremely satisfied with the included web design tools”

Web Design tools are designed specifically for just that – designing websites, portals, and other applications that run over the Internet. Adobe’s Macromedia Studio 8 was built from the ground up with precisely these capabilities in mind, and it is no surprise that developers using Adobe’s products are extremely satisfied with the included web design tools. Last year Adobe released Creative Suite which carries on a long line of superior web design tools. These include Dreamweaver (one of the most popular web design and html tools), Flash (the number one animation technology), plus Fireworks 8 (a graphics editor), FlashPaper 2 (a PDF tool), and the website manager Contribute 3.

Adobe’s tools did significantly better than any of the other IDE toolsets, with Rational App Developer the closest behind. RAD web tools have been greatly enhanced and now offer excellent capabilities. This is reflected in the change in position relative to other IDEs in this year’s user satisfaction survey.

JDeveloper has also added to their web development capability with the release of JDeveloper 11G.



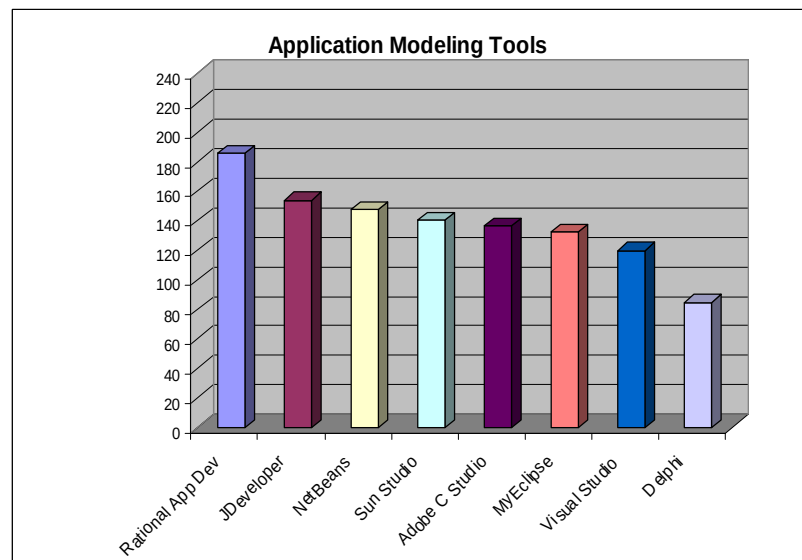
Application Modeling Tools

In 2006 when we asked developers to rate quality of application modeling tools, it was no surprise that the Rational Application Developer turned out on top with JDeveloper rated almost as high. Last year was a similar result except that JDeveloper was slightly on top but only by the smallest of margins. This year the excellent application modeling tools that were at one time the cornerstone of the Rational product line before Rational was acquired by IBM have again been ranked more highly by their users than any other IDE and toolset and by a wide margin.

Traditional modeling tools allow an architect to create a visual model of how a complex application will work, outlining components, calls, and other functions and thus providing means to optimize and provide stability to applications.

Thus, application modeling tools are most useful in environments where large and complex applications are common. These are precisely the environments that Websphere and Oracle's flagship database product occupy, and so it is no wonder that these two companies have devoted time to perfecting their modeling tool capabilities.

"...the excellent application modeling tools that were at one time the cornerstone of the Rational product line before Rational was acquired by IBM have again been ranked more highly by their users than any other IDE and toolset."



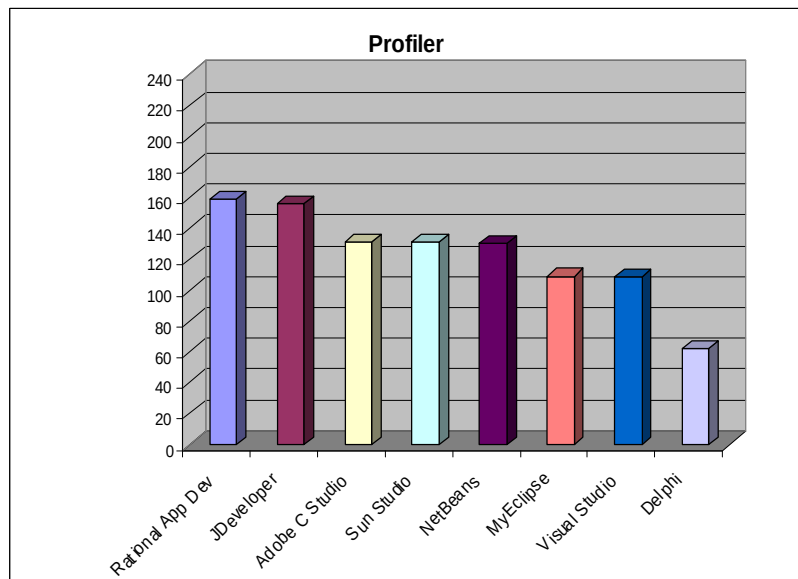
Profiler

A performance profiler is a tool that is used on prototype code in order to optimize it or make it more efficient. Most profilers today have distributed capabilities so developers can identify bottlenecks in their code throughout a network. Performance analysis is often used to determine how long certain parts of the program take to execute, how often they are executed, or to generate a call graph. This information is then used to identify those parts of the program that take the longest to complete. These time consuming parts can then be optimized for better performance.

“In the past, many IDEs didn’t have a profiler or performance analyzer included.”

In the past, many IDEs didn’t have a profiler or performance analyzer included. However, that is changing. As more multi-core processors come into the mainstream, developers will once again have to optimize their code to provide performance instead of relying on an ever faster processor clock speed.

Both Rational Application Developer and JDeveloper have well developed and efficient profilers.

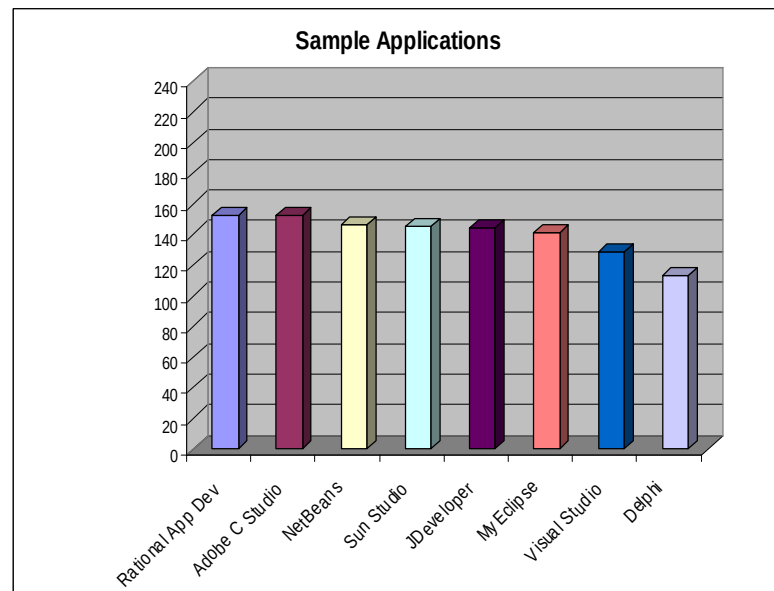


Sample Applications

Some might say that sample applications are not a tool and not necessary for the operation of any IDE and therefore, they are not a valid feature for comparison. However, sample applications not only illustrate to the developer how a tool works, but they're also mostly reusable. Developers can take snippets or whole programs of sample application code and build them right into their own applications. Thus sample applications provide both education and some shortcuts.

Sample applications do not ship only with IDEs. In the Open Source community there are plenty of applications with which source code can be found, thus explaining NetBeans relatively high score in this area.

"...sample applications not only illustrate to the developer how a tool works, but they're also mostly reusable."



Sample applications were considered among the least important of the attributes this year in this study, though in previous years developers have given them more weight. The decline of value perception for sample applications may be due to the Open Source community, at least in part. Today, the value of reusability of sample apps is very much offset by the availability of Open Source components that can be simply plugged in and used free of charge.

Evans Data Corp (EDC) has made every effort to produce the highest quality research product in this effort. The customer understands that EDC uses those statistical and data gathering techniques which, in its opinion, are the most accurate possible. However, inherent in any statistical study is a possibility of error, which must be taken into account in evaluating the results. Evaluations and interpretations of statistical research findings and decisions based on them are solely the responsibility of the customer and not EDC. The conclusions, summaries and interpretations provided by EDC are based strictly on the analysis of the data gathered, and are not to be construed as recommendations; therefore EDC neither warrants their viability or accuracy nor assumes responsibility for the success or failure of any customer actions subsequently taken.

This report is the exclusive and proprietary property of Evans Data Corp and is subject to limited distribution and restricted disclosure only. Any unauthorized use, reproduction, or transfer of this material is strictly prohibited. Clients of this study may freely distribute this report to full-time employees within their corporate entity, providing that the above requirements are adhered to by all recipients. Clients may not distribute any of this information outside of their organizations without written permission from EDC.