Member-only story

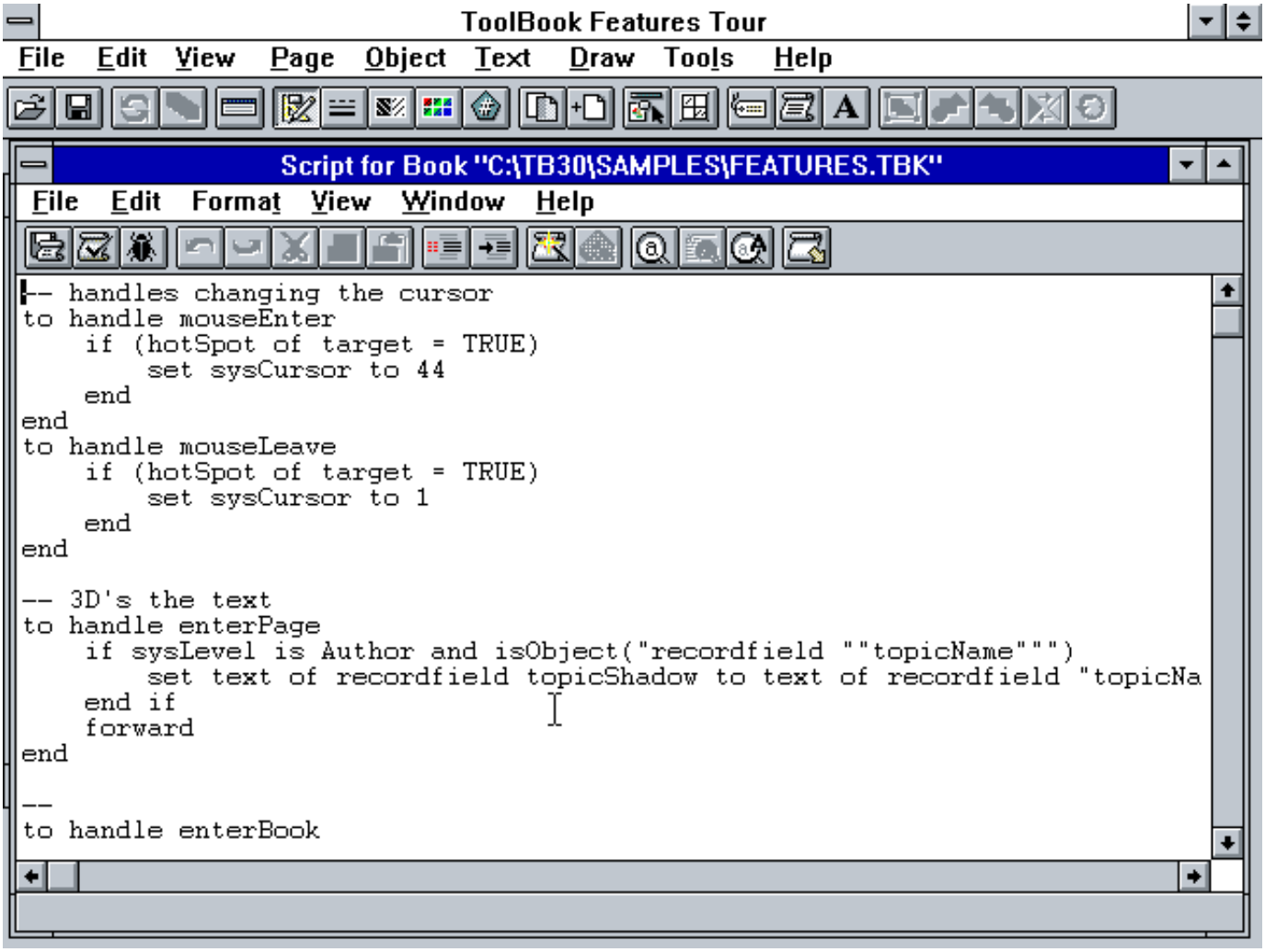
Revolutionary IDEs That Shaped Software Development

Jan Kammerath · Follow

14 min read · Jul 16

236 18

Back in ye olde days, writing code was merely black text on white background. Writing code required a hefty amount of due care and attention. Typos weren't immediately highlighted and syntax errors only popped up during compile time. It was an entirely different experience to programming today.



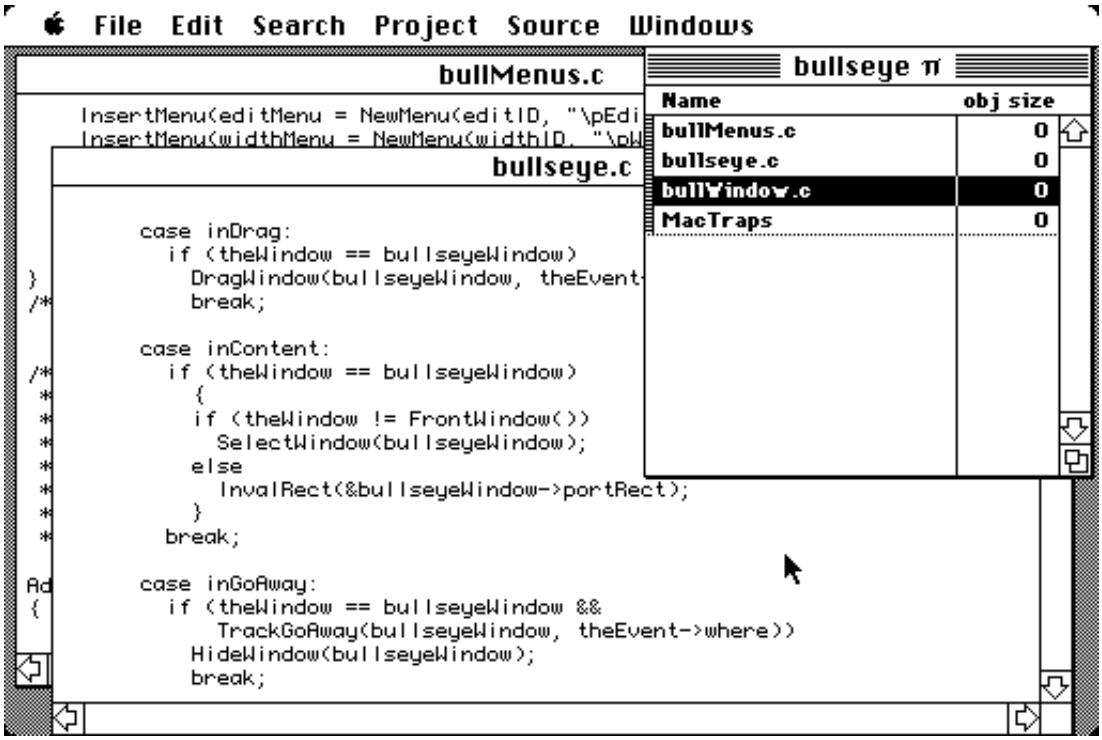
Asymetrix ToolBook — once popular for multimedia, now long forgotten

In this article, we'll be looking at the top 10 most influential development environments of the last 30 years of computer programming until today. We'll rank the IDEs from Top 10 to the absolute number one most influential of all time.

10. THINK C

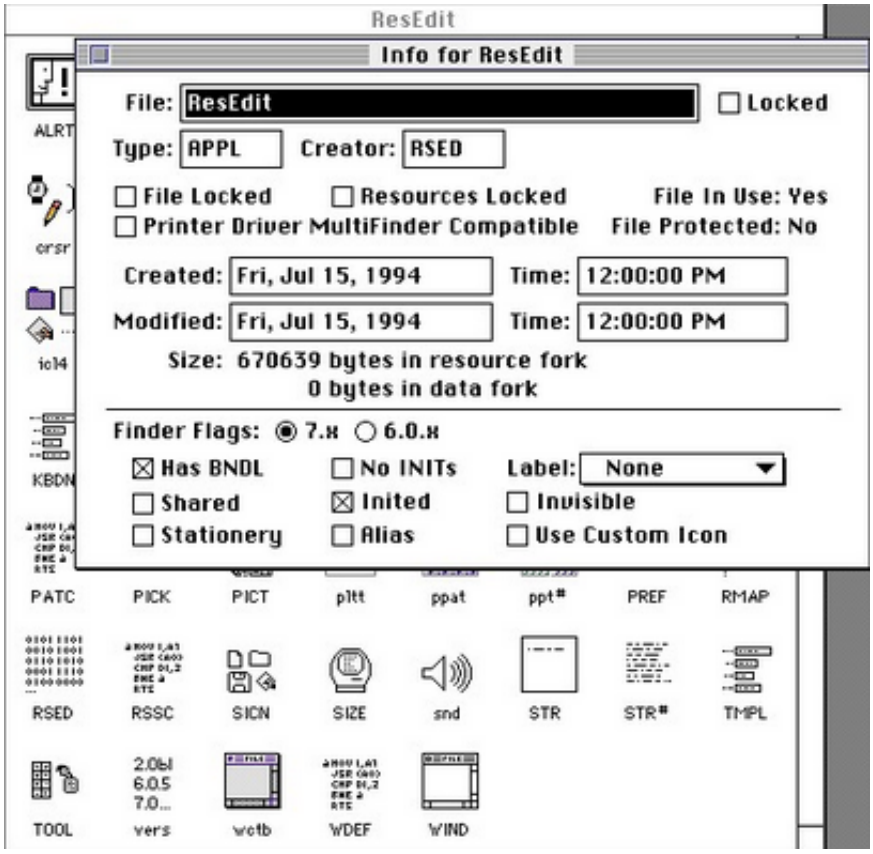
Building applications for the original Macintosh required using the Macintosh Programmer's Workshop or MPW in short. Since the Macintosh was so well ahead of its time with the graphical user interface, programming it was a totally different experiences as compared to other systems like DOS. Apple's MPW was tailored towards experienced

programmers and came at hefty prices. When Think Technologies released Lightspeed C in mid 1986, it drastically simplified the approach to programming on the Macintosh.



THINK C Version 3, formerly Lightspeed C

Think C, or Lightspeed C in the early days, made such an impressive start that the famous BYTE magazine named it the product of the month of September 1986. It combined powerful features, a very easy to navigate GUI and an extremely competitive price retailing at around $200 ($500 in today's money) as compared to the $600+ (over $1,500 in today's money) that the MPW C asked for. Think C also came with a source code level debugger which we totally take for granted today.



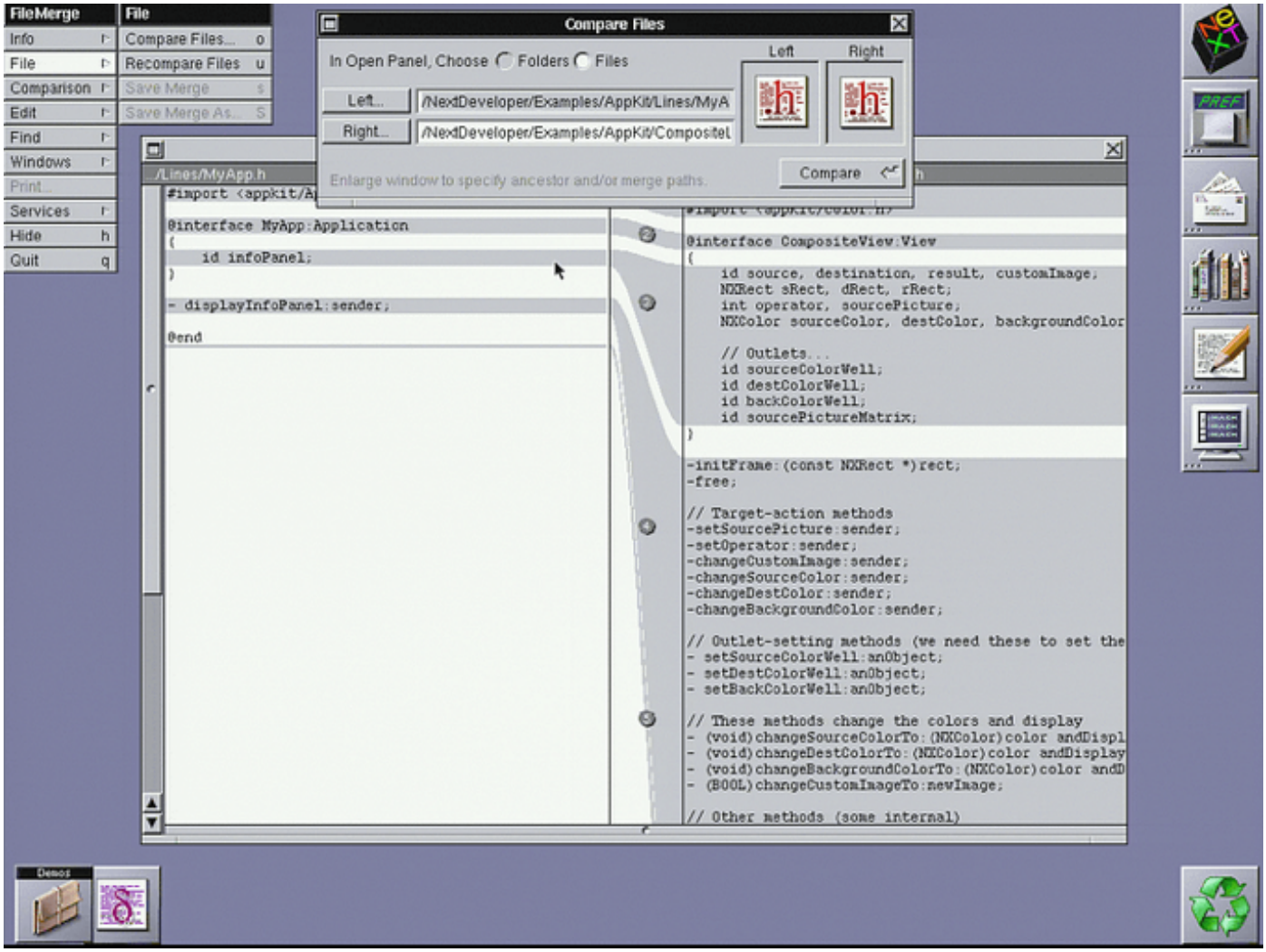
ResEdit application for the Macintosh

Since the Macintosh User Interface was all new and unheard of, developers needed to be able to construct user interfaces. Apple had offered a small tool called “ResEdit” that came either as standalone or part of the MPW package. In standalone it would cost around $100 to $200 which is around $500 in today’s money. Programming was an expensive excersive back in the old days. Although WYSIWYG GUI editors are free and standard in any modern IDE today, ResEdit mark the first native WYSIWYG editor for GUIs.

9. Apple Xcode

Long before the iPhone relased in 2007, Apple released their IDE named Xcode which is the primary and only official IDE from Apple to build apps for MacOS, iPhone, iPad, WatchOS, CarPlay and all other Apple

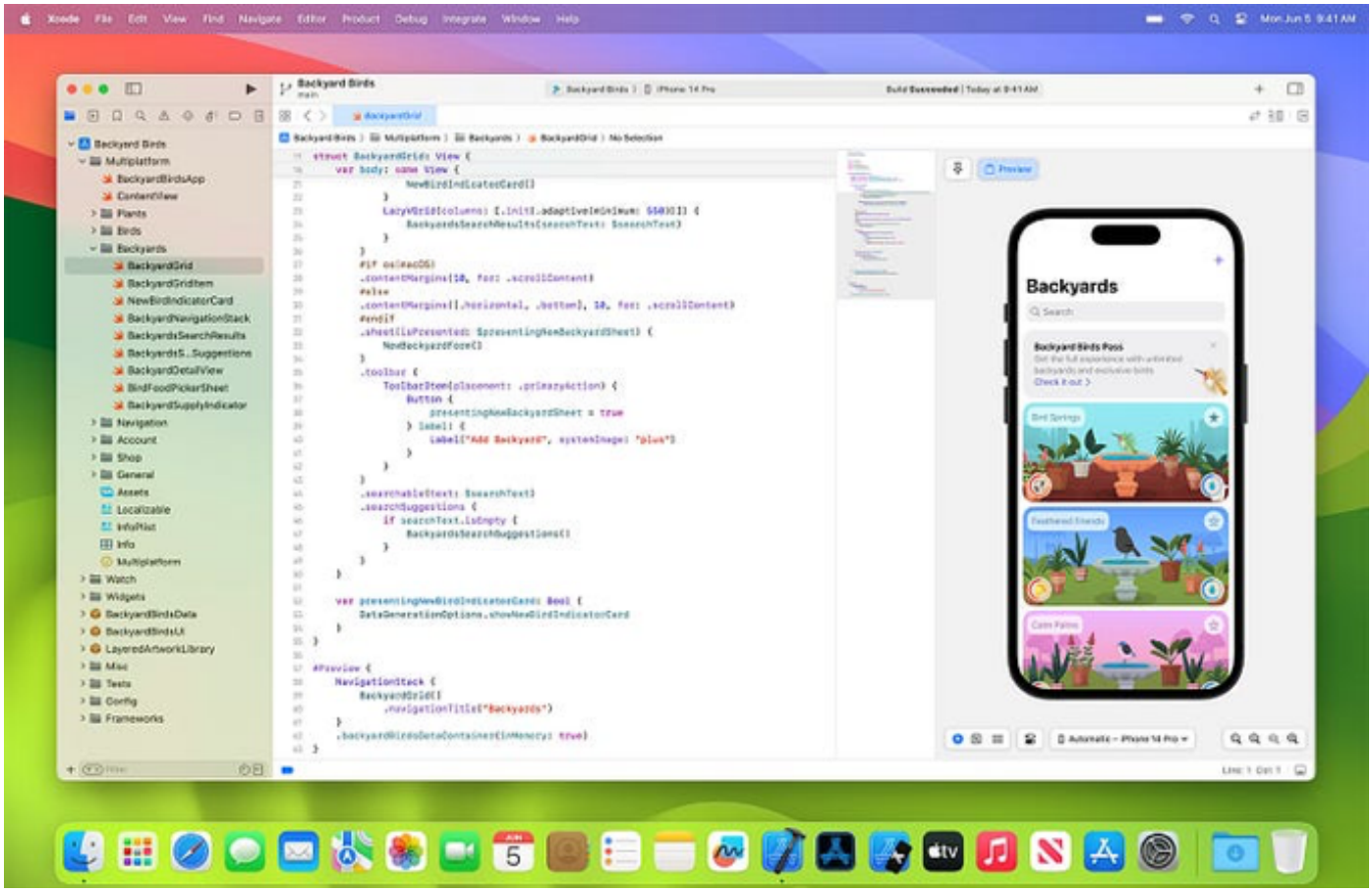
platforms in their ecosystem. When Apple transitioned from the classic Macintosh to OSX, developers had to change drastically. While THINK C, the Apple MPW and CodeWarrior on Macintosh System 8+ were all C and C++, the new OSX predominantly used Objective C with its Xcode IDE based on NextStep.



Programming on NextStep — notice something similar to Xcode?

The introduction of Xcode also marked the end of the many IDEs that were available for the Macintosh system as they would've to port their entire IDE and all the runtime libraries to the newly build OSX which is essentially NextStep. The libraries and objects on the new OSX all

identified with an “NS” prefix highlighting their NextStep origin. Besides being all new, Xcode also came with a new paradigm at Apple. Developers for the Apple ecosystem were and are still more or less forced to use Apple’s development environment. This was already somewhat the case the MPW, but the “walled garden” strategy of a closed ecosystem was strengthend with the introduction of Xcode.



Xcode 15 with hot reloading when building iOS apps

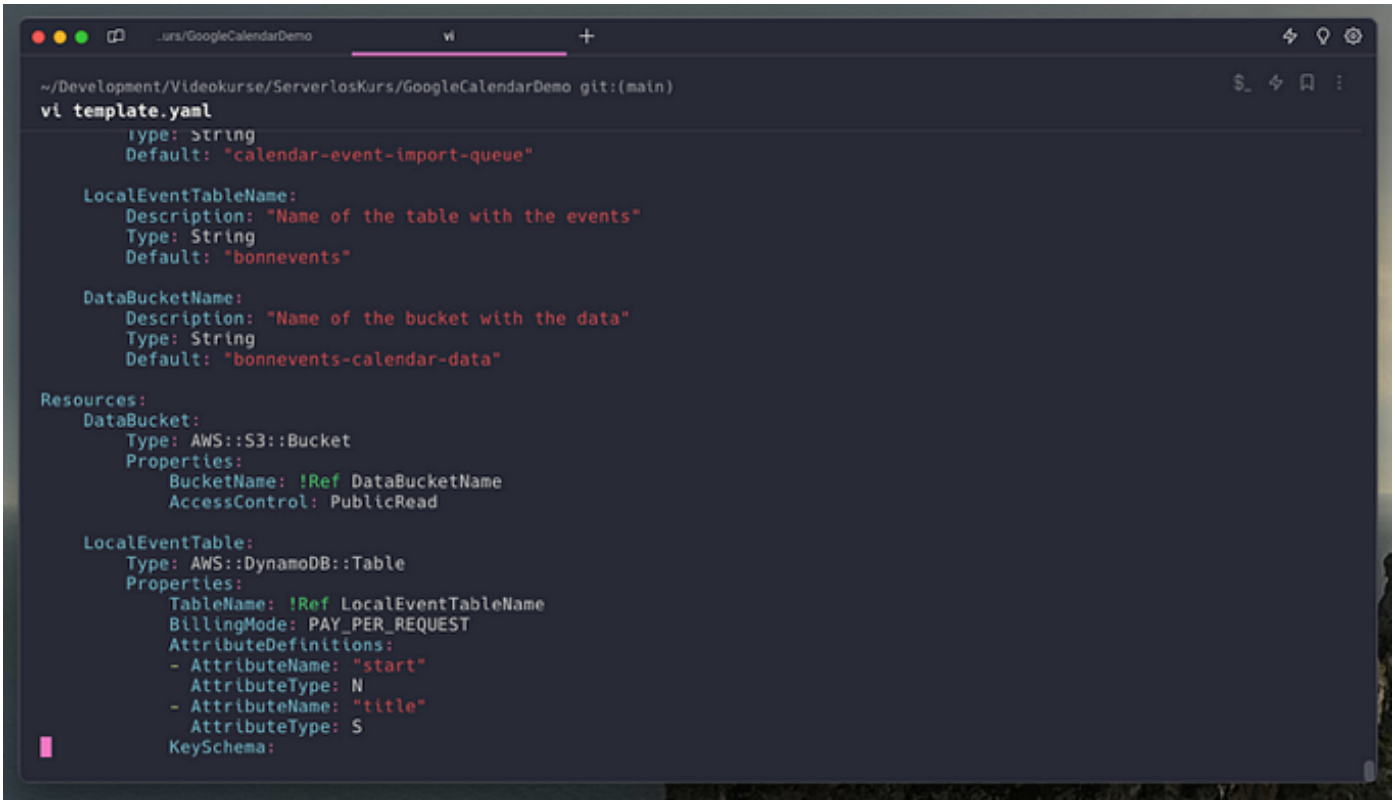
Although many other IDEs such as MonoDevelop allow programming for MacOS, the introduction of iOS for the iPhone, iPadOS for the iPad, watchOS for the Apple Watch and CarPlay for vehicle entertainment systems pretty much require developers to use Xcode if they want to have access to the entire Apple ecosystem and all features. The $99 annual

price tag for the Apple Developer programme on the other hand is a fair price considering that it includes all the development tools and the distribution through the Apple App Store.

Xcode is probably the first IDE for a large closed ecosystem. It's an amazing IDE with great advantages that come with a closed ecosystem, yet is has challenges keeping up with the latest developments sometimes. The introduction of the Swift programming language as a replacement for Objective C increased the attractiveness of building for Apple platforms.

8. vim

Released as a text editor in 1976, vi *(stands for "Visual")* and vim *(means "Vi Improved")* quickly become somewhat of a default editor for Unix and Linux. Yes, there's emacs and it's a nice editor as well. However, in 2015 there was a survey by Stack Overflow which highlighted that vim is more widely used than emacs. Since so many advanced IDEs arrived over the years, how come vim survived? *Very simple:* it kept up with the times. It has syntax highlighting, debugging works just fine and vim is fast. Really fast. Once you're up to speed with the keyboard shortcurts, you'll never want to use anything else for simple editing. Especially if you're working through the Terminal all day anyway.



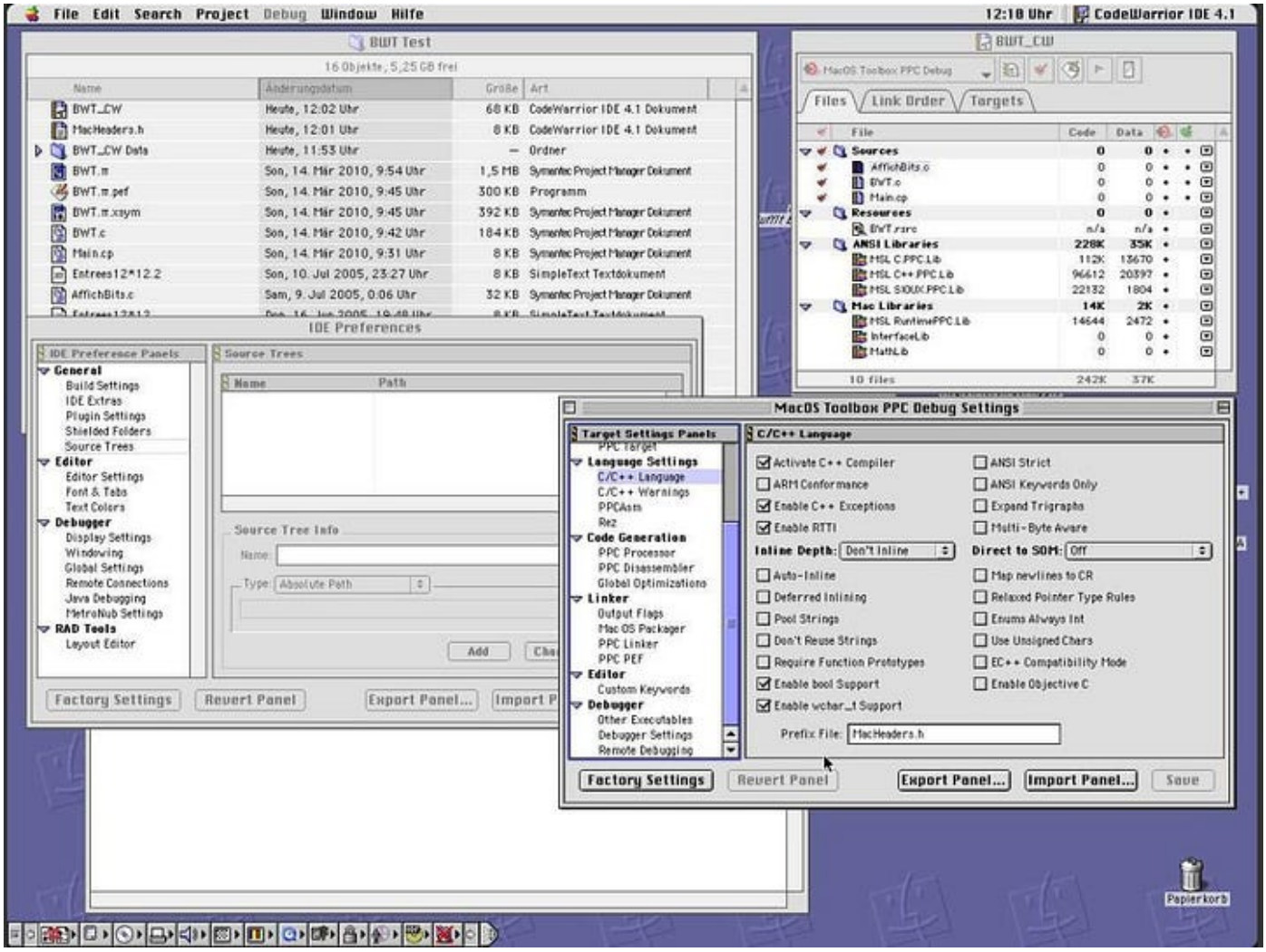
VIM, Vi IMproved 9.0 in the Warp Terminal on macOS

Vim also has the advantage of being highly portable across all operating systems. It runs on Windows, Linux, Mac, large Mainframes, your MacBook and your network router. This portability means that vim is accessible to developers on any platform they quickly need a helping hand editing config files, source files and much more. Newer versions such as Neovim bring even further comfort to development while sticking to the way vi and vim work. It is very likely that vim will survive many more decades to come. It was one of the first and most popular editors for Unix and is still the dominant choice on Linux and Unix. Especially experienced developers love vim for its speed and keyboard friendly approach.

7. CodeWarrior

When Apple moved from the 68k CPU to the PowerPC platform,

companies like Symantec, the then owner of THINK C, had to migrate their IDE to the new processor architecture. A company called MetroWerks worked on the CodeWarrior IDE with many members of the former THINK C development team. CodeWarrior was known for its ease of use, faster compile times than Apple's own MPW and attractive pricing. It's first release came out in 1993.



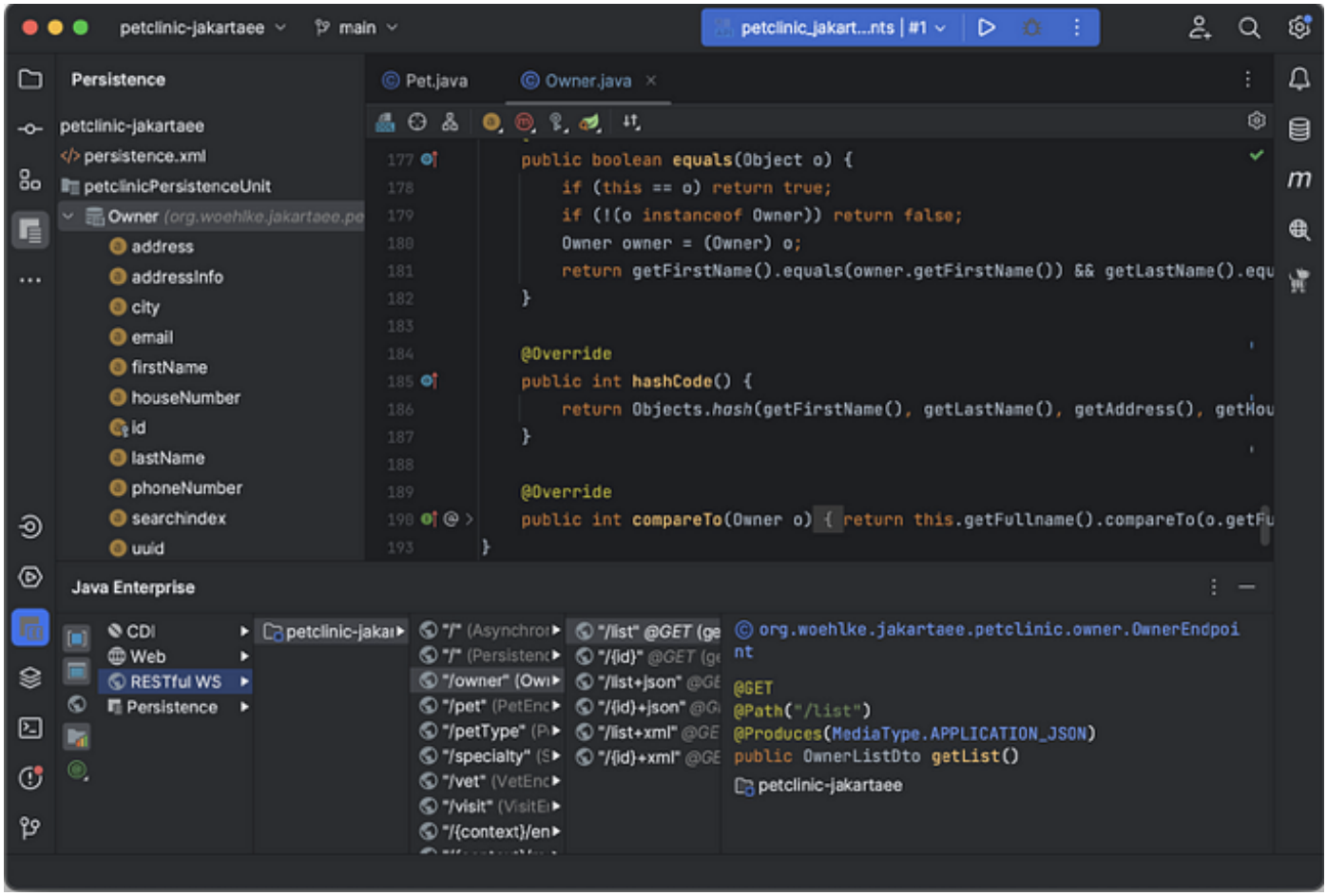
CodeWarrior 4.1 on Mac OS 9 with PowerPC architecture

From 1994 up until 2002, CodeWarrior became the dominant IDE for development of apps for Mac OS 8 and 9. In its history, Apple had moved from 68k CPUs to PowerPCs, to Intel and then to the new arm64 Apple

Silicon CPUs. Every change of the hardware platform is an absolute nightmare for IDE vendors. THINK C and CodeWarrior were the only IDEs to ever get a serious marketshare for professional software development on the Macintosh from the late 80s to the early 2000s. Only to be superseded by Apple's own IDE, Xcode. And not because Xcode was significantly better, but because Apple could force Xcode onto developers and give 3rd party IDEs a hard time to survive on the platform.

6. IntelliJ IDEA

When Java popped up in the mid 1990s, IDEs for it were few. Writing Java often involved just using editors and then compile on the command line. Netbeans and Microsoft's Visual J++ were probably the first prominent IDEs for Java. IntelliJ IDEA was released in January of 2001 and came with advanced code navigation and refactoring capabilities.



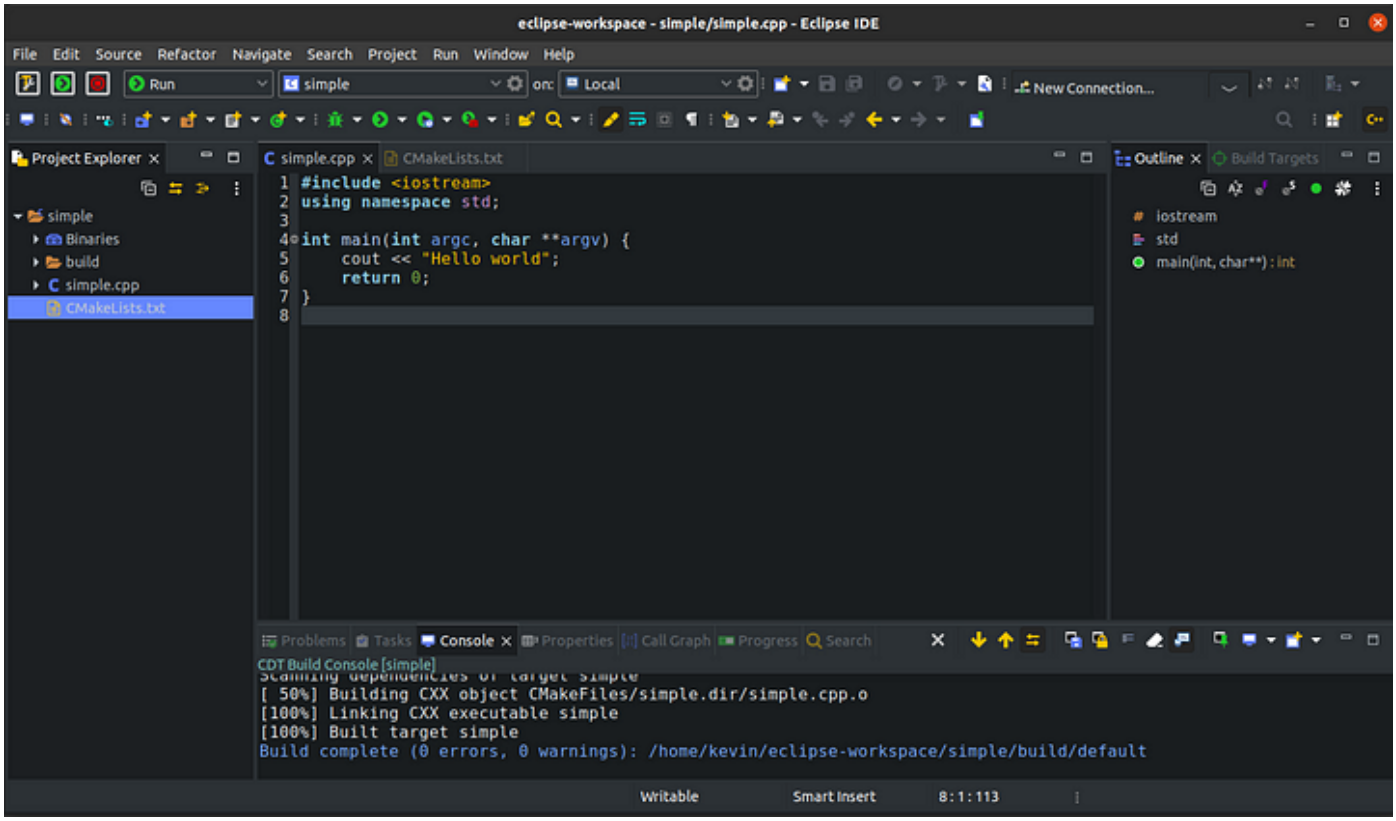
IntelliJ IDEA comes with every feature imagineable

IntelliJ to this date is considered on of the most advanced IDE ever to have been built. Probably only challenged by the fully fledged Microsoft Visual Studio. Besides every features imagineable, IntelliJ also highlighted how IDEs have come from being lightweight code editors to, sometimes monstrous, development environments with endless customization options. It often takes developers days to become familiar with new modern IDEs. IntelliJ started the era of the fully fledged integrated development environment that packs all tools ever needed by developers into a single application.

5. Eclipse

Around the same time as IntelliJ, Eclipse saw the light of day. While

Eclipse was originally intended for the Java programming language, it quickly expanded into any language known to men. Further, it's extensive plugin environment allowed to do anything with it and by anything I mean absolutely anything. While IntelliJ today is also available for a variety of different languages like PHPStorm for PHP and many others, Eclipse was pretty much the first IDE that aimed to be truly multilingual, multi-platform and multi-multi-anything.



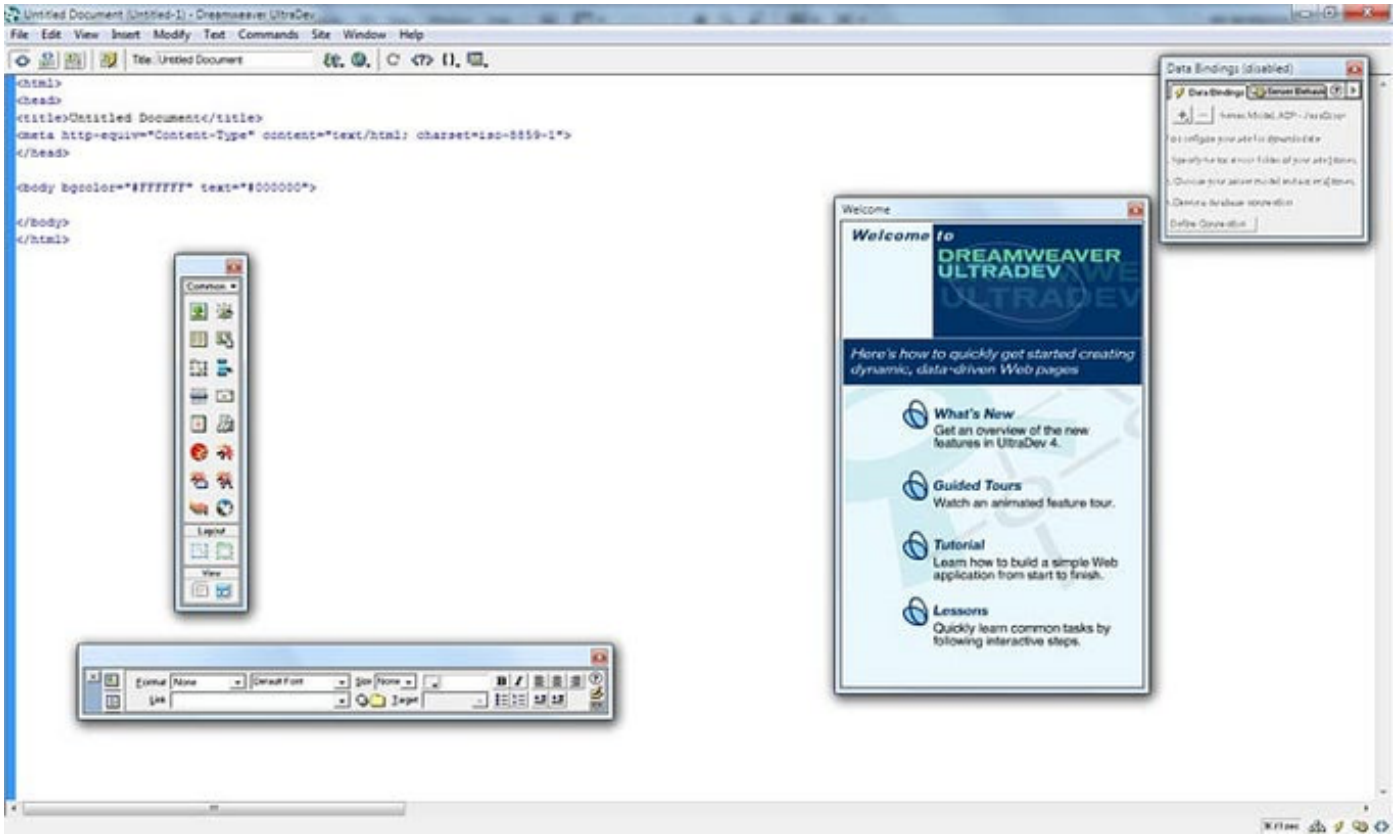
C++ development with the Eclipse IDE

There's almost nothing you cannot build with Eclipse. From C++ to PHP, Python and Go. Eclipse has got a plugin for everything. Eclipse opened the era of the open development environment that anyone could use to build his or her own IDE. Eclipse ended up feeling very messy at some points. Regardless of its future or its downsides, Eclipse deserves a special place in the hall of fame for liberating the market of development

environments. I wrote my first Google App Engine application in Eclipse back in 2008. When Eclipse peaked, Microsoft would still charge hefty amounts for its development environments. Eclipse was free, fully fledged and took developers hearts by storm.

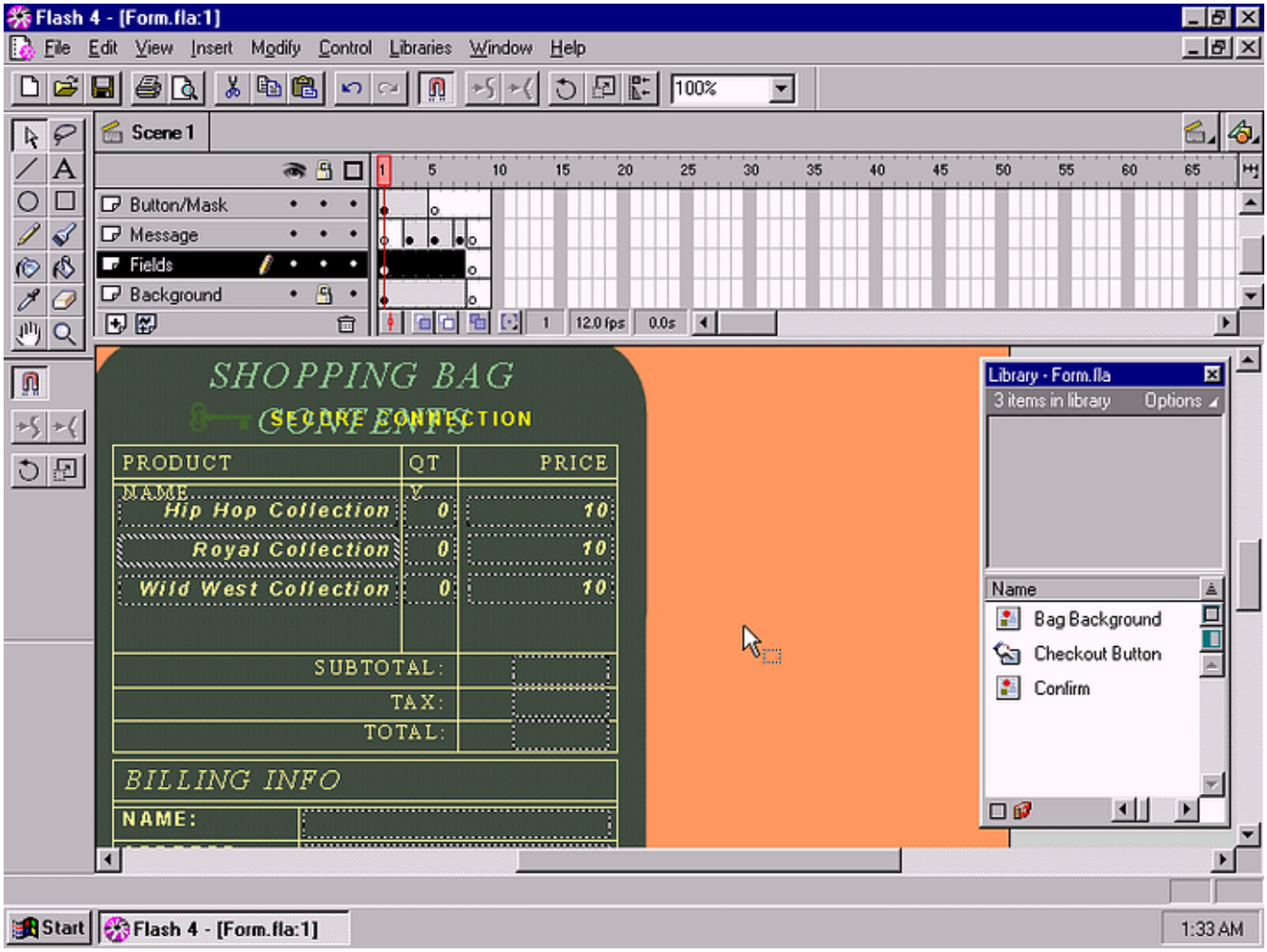
4. Dreamweaver, Flash & Fireworks

The years 1995, ’96 and beyond marked the breakthrough of the world wide web and the appearance of job titles such as webmaster and later web developer. Most IDEs at the time, like the prominent Visual C++ of 1995, were tailored for desktop and server application. The maximum they offered in terms of design was a WYSIWYG GUI builder for the target operating system allowing to design windows, lists and buttons. They had nothing to offer for the browser.



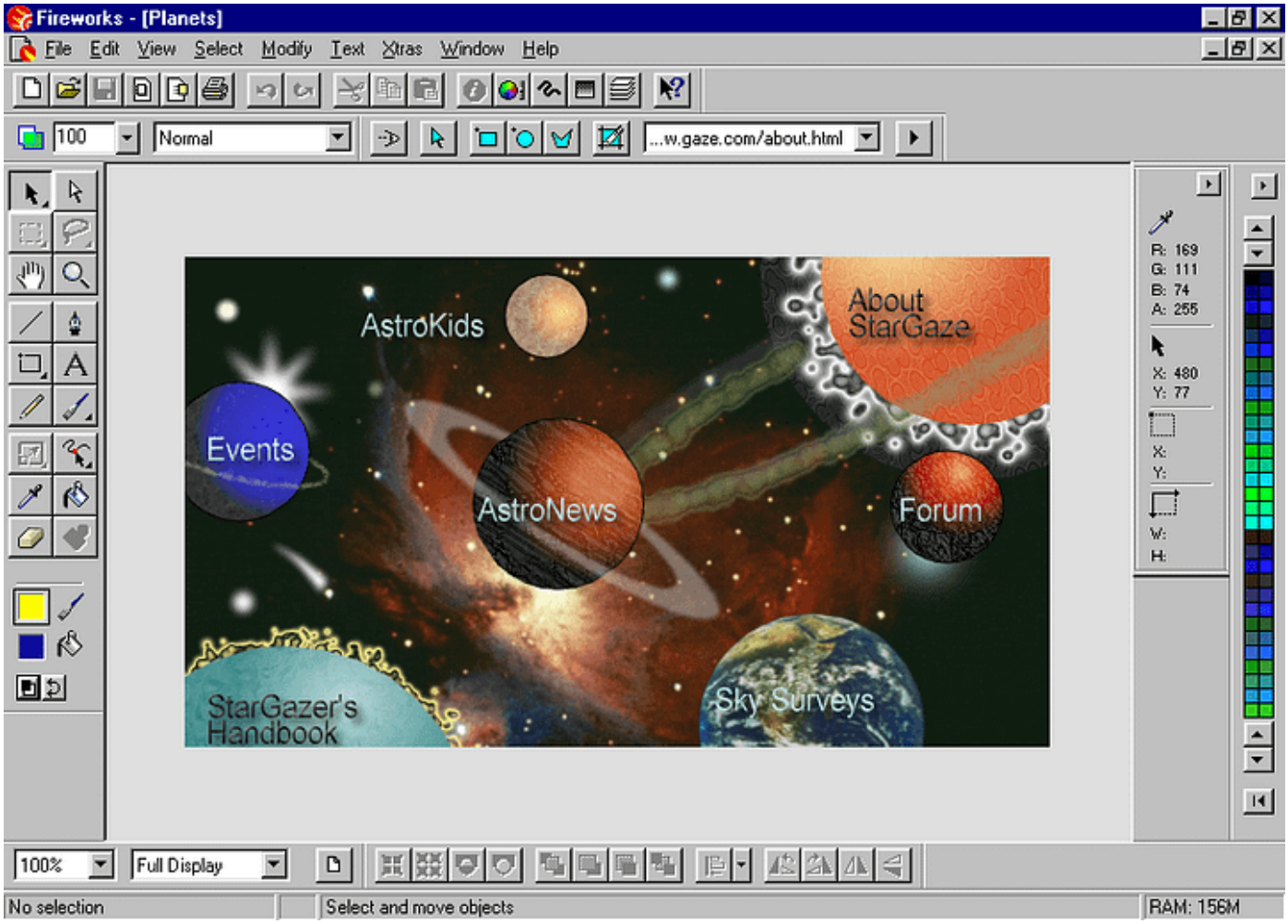
Dreamweaver UltraDev, a Web Development IDE released in December 2000

That was when companies such as Macromedia launched their offering with software like Dreamweaver, a WYSIWYG editor for HTML, later CSS and JavaScript. The first version of Dreamweaver was Macintosh only and a Windows version followed later on. Dreamweaver had its peak in the New Economy bubble between ’99 and 2002. Dreamweaver UltraDev 4 was the most advanced web development IDE of its time. It did not just include development features for frontend JavaScript, HTML and CSS, but also supported Microsoft ASP with server-side JavaScript and Visual Basic, JSP, PHP and Macromedia’s own ColdFusion.



Macromedia Flash: gone for good, forever remembered

Macromedia Flash and its ActionScript scripting language provided endless possbilities for building any multimedia application to run in browser with the Flash plugin installed. It will forever be remembered as a nightmare by some and an incredible experience by others. People built the first games for the web with flash, the first popular video and audio streaming sites used flash and website owners loved animated introductions on their homepages. Flash and ActionScript programming perfectly represents the world-wild-web, as the era was often remembered as.



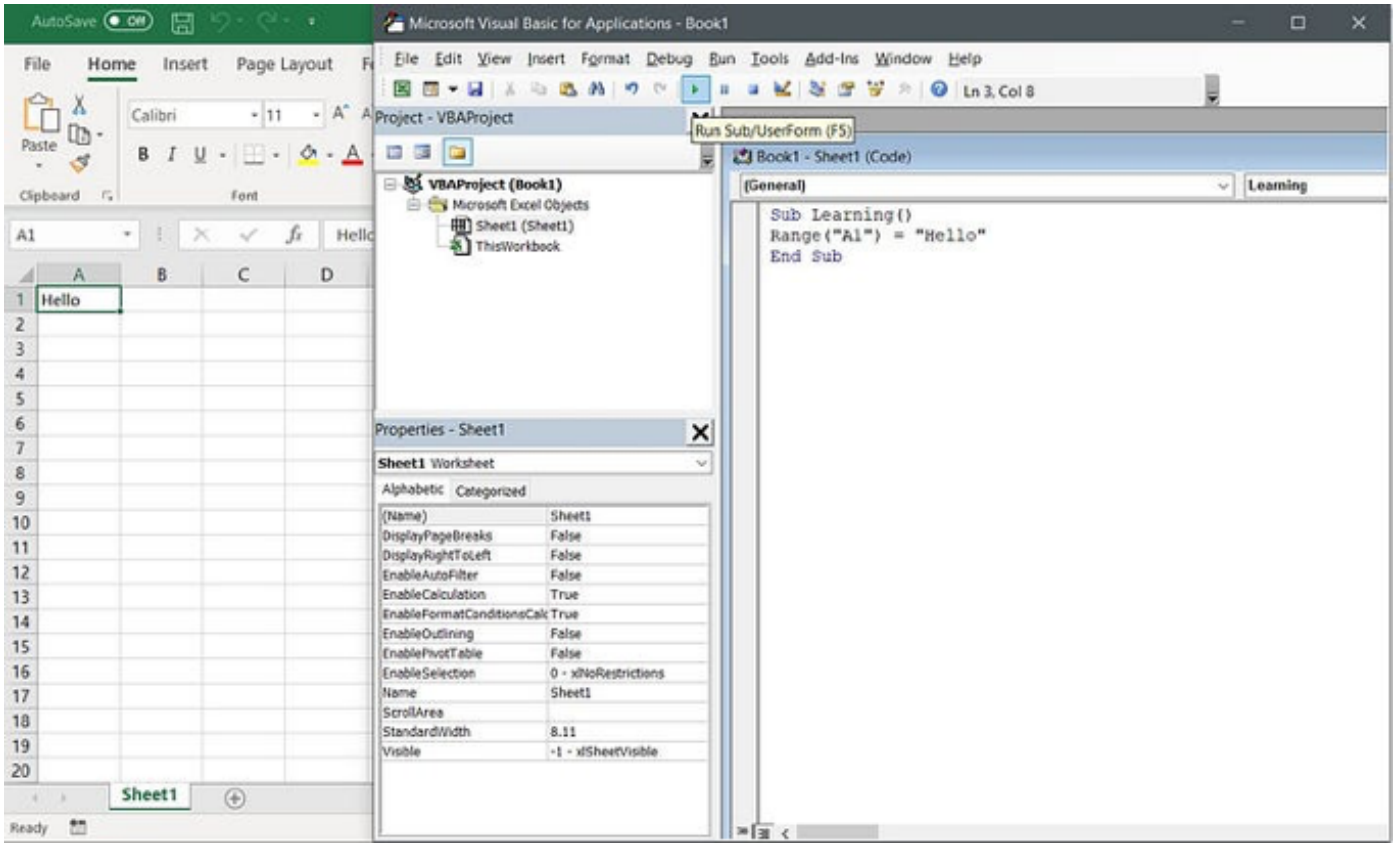
Macromedia Fireworks: Graphics and HTML WYSIWYG editing

The bunch of Macromedia web development tools wouldn't be complete

without Fireworks. Although Flash, given the existence of ActionScript, could arguably be a development environment, Fireworks comes across as a vector graphics editing software. However, Fireworks had an integrated code generator that allowed you to slice the graphics, embed animations and export the HTML contents including the necessary JavaScript snippets. Dreamweaver, Flash and Fireworks were unheard of WYSIWYG tools that took WYSIWYG to a whole new level. While Dreamweaver still exists, Flash and Fireworks have been abandoned after Macromedia was acquired by Adobe. Dreamweaver is part of the Adobe Creative Cloud today. If you're interested in the history of Web Development, have a look at my article A tale from 30 years of HTML.

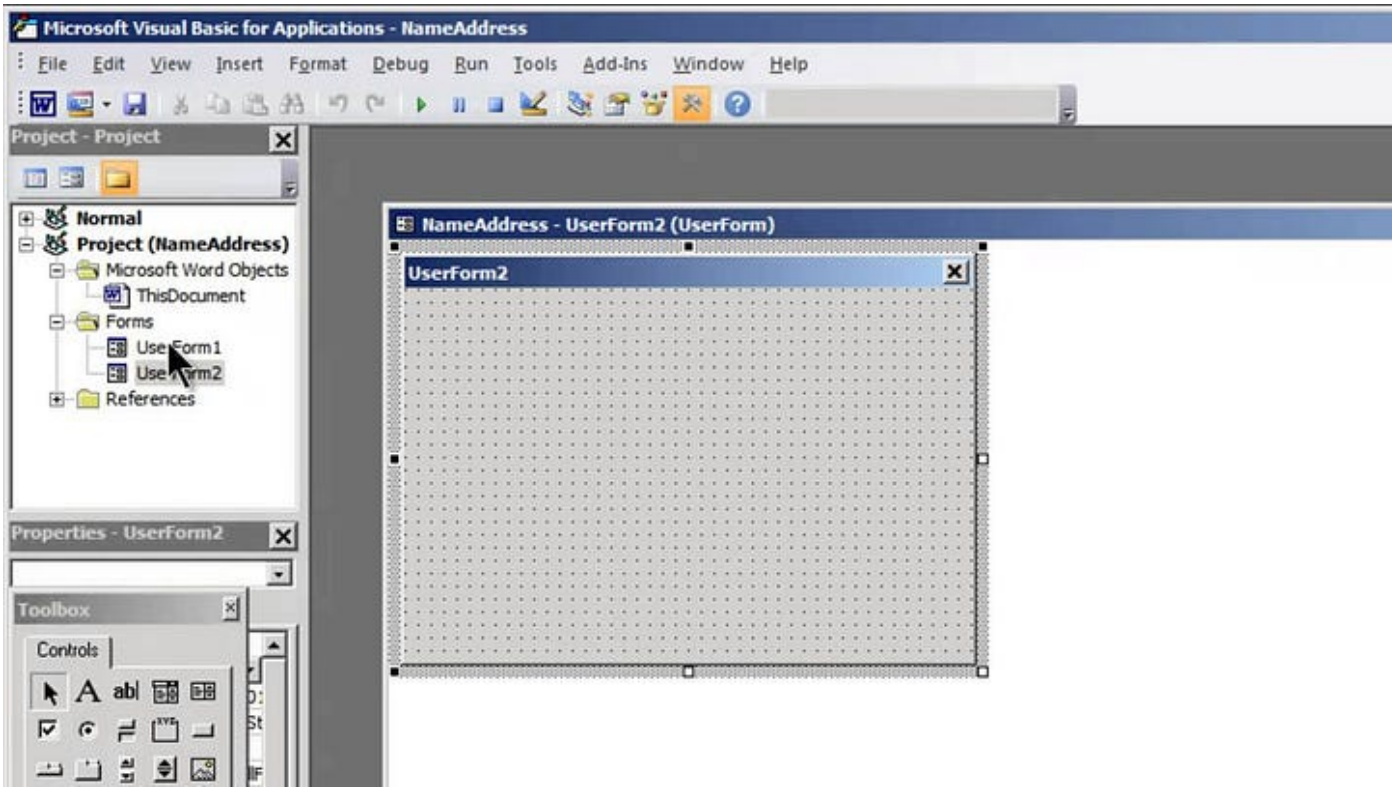
3. Microsoft Word and Excel

While there are memes about people writing their code in Microsoft Word, anyone who ever pressed ALT+F11 in Excel on Windows knows what happens. Both Word, PowerPoint and Excel will present you with a fully fledged Visual Basic For Applications Development Environment. The IDE itself is very similar to Visual Basic 6 from 1998 and later. The language is Visual Basic 6 (known as VB6) exclusively. VBA was first launched with Excel in 1993. VBA was imagined by Bill Gates as a universal macro language and became exactly that for decades.



Visual Basic for Applications (VBA) in Microsoft Excel on Windows

Spreadsheets and Excel specifically were the main business application on computers since the late 1980s. Only followed by word processing with applications like Microsoft Word and presentations with Microsoft PowerPoint. The ability to have a complete IDE, based on the latest Visual Basic 6, inside Microsoft Excel allows the spreadsheet user to connect the spreadsheet to any data source and have it execute any task. Automatcially collecting all network hosts in the local network in a spreadsheet? Excel and VBA can do that automatically.



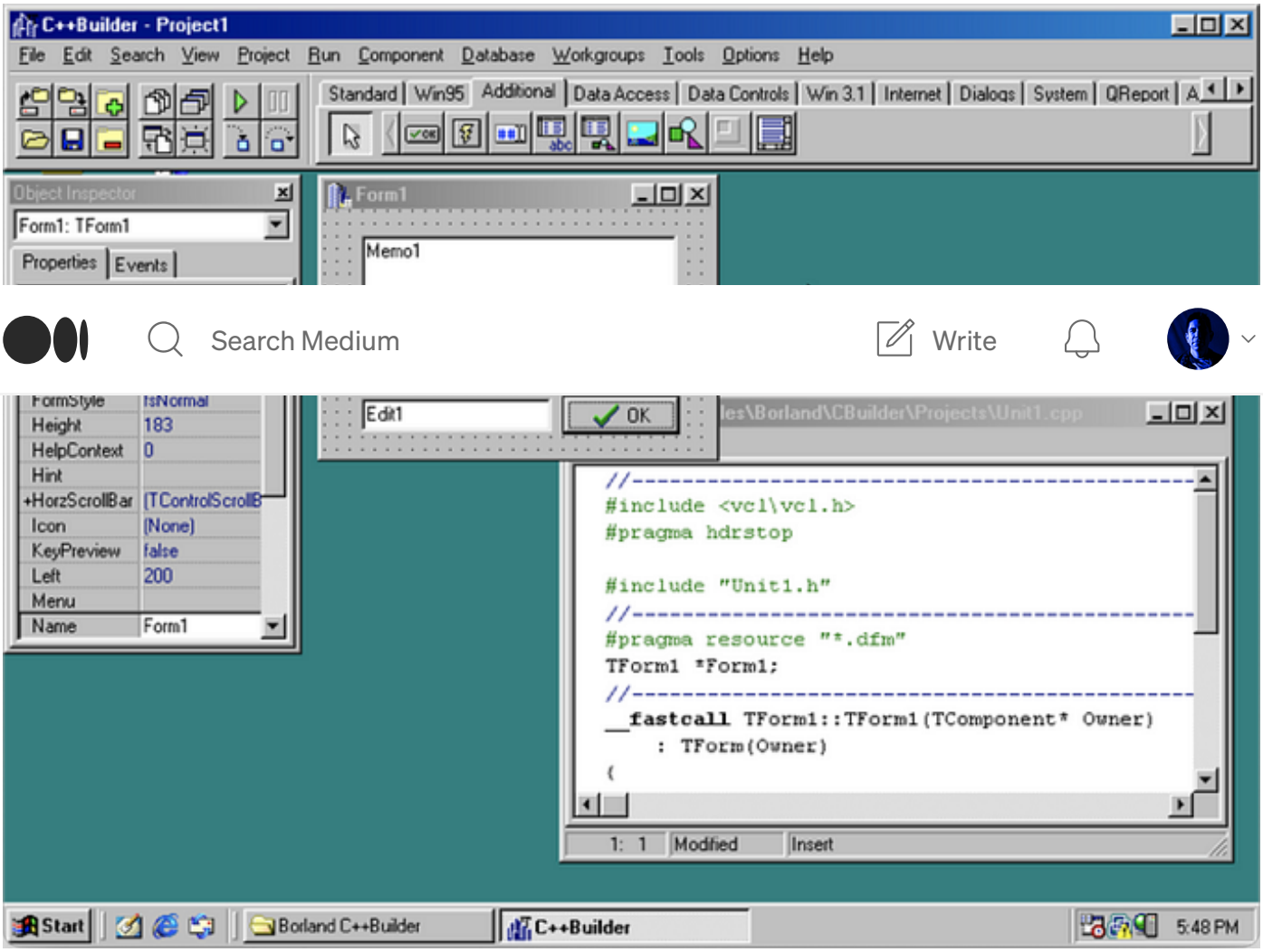
The Form designer and editor of VBA in Microsoft Word

VBA is not limited to simple algorithms and basic macros. It includes the complete GUI builder, classes, modules and practically anything that VB6 has. This turns innocent Excel sheets, Word documents and PowerPoint presentations into complete applications. VBA is the hidden secret that catapulted Microsoft Office into its market dominance. The IDE would no longer be a standalone application, but large applications would now bring their own IDE along. It marked the beginning of extending standard applications through an IDE integrated in the application itself.

2. Borland C++ Builder and Delphi

Borland was the household name for software development environments in the 1980s and 1990s up until the early 2000s. Borland offered the famous C++ Builder, Delphi as a successor to Turbo Pascal, acquired dBase and so much more. When you talked about IDEs in the

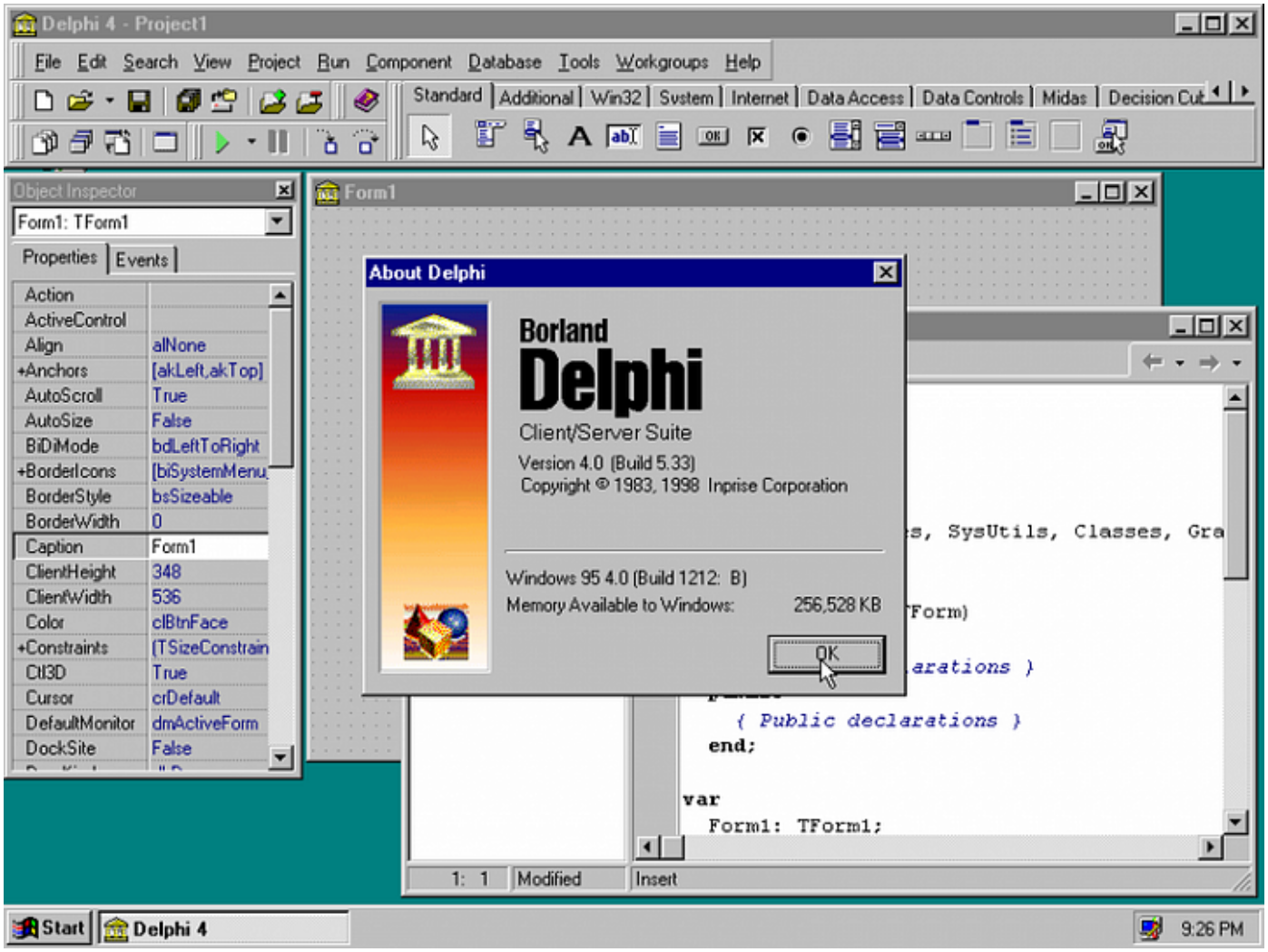
90s, Borland was the name that would come up. Many of the most prominent Windows applications of the 90s were written with Borland software. Schools taught young software engineers in C++ using the Borland Builder. I myself learned C and C++ with Borland software for both DOS and Windows.



A C++ Windows Desktop application with the Borland C++ Builder

What Visual Basic was for Microsoft, Delphi was for Borland. The late 90s and early 2000s had large Visual Basic and Delphi communities. Delphi, being an Object Pascal dialect, lives to this day and remained the biggest competitor to Microsoft's Visual Basic. Both Borland IDEs came with GUI

editors that were highly advanced and featured a number of small advantages over Microsoft's offerings. Borland also offered a wider variety of custom user interface controls as Microsoft would stick to the Windows standard components.



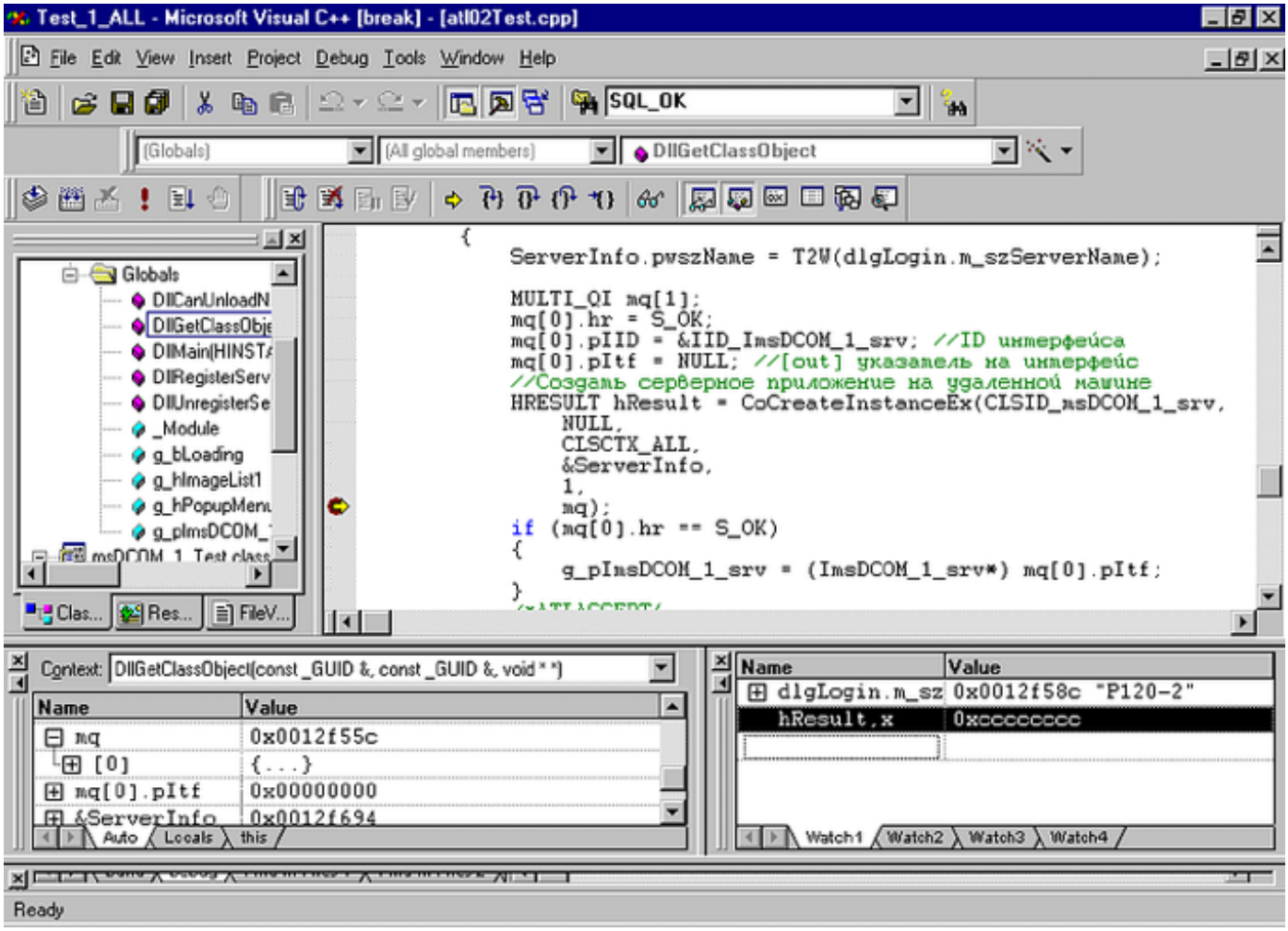
The Delphi 4 IDE from Borland

For years Borland C++ and Delphi competed with Microsoft Visual C++ and Visual Basic. Some of the most advanced features, like code completion, advanced syntax highlighting, easy to use components, simpler abstract system APIs and a lot more were invented during this phase of fierce competition between Borland and Microsoft. While many

remember the browser wars back in the days, developers still remember the IDE wars that happened during the same period between Microsoft and Borland.

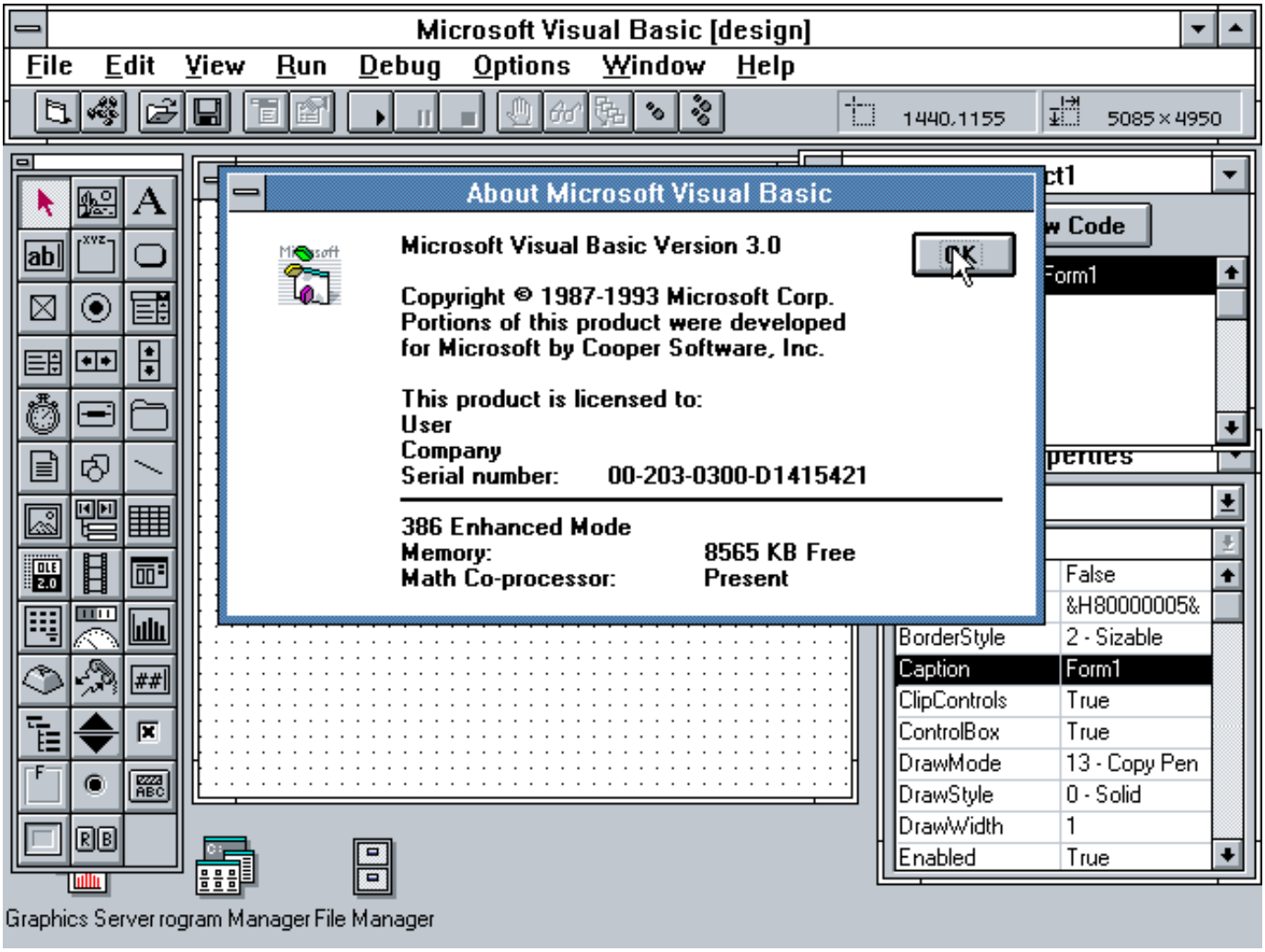
1. Microsoft Visual Studio

The number one, the uncontested heavy weight champion of Integrated Development Environments for over 30 years is Microsoft with its Visual Studio software, coming from Visual C++ and Visual Basic all the way to Visual Studio .NET with C# and Visual Studio Code. From the day Microsoft was founded, the company focused on software development and software development tools. Microsoft BASIC was the foundation product of the company. Microsoft is a software giant, but at its core it's a software development giant.



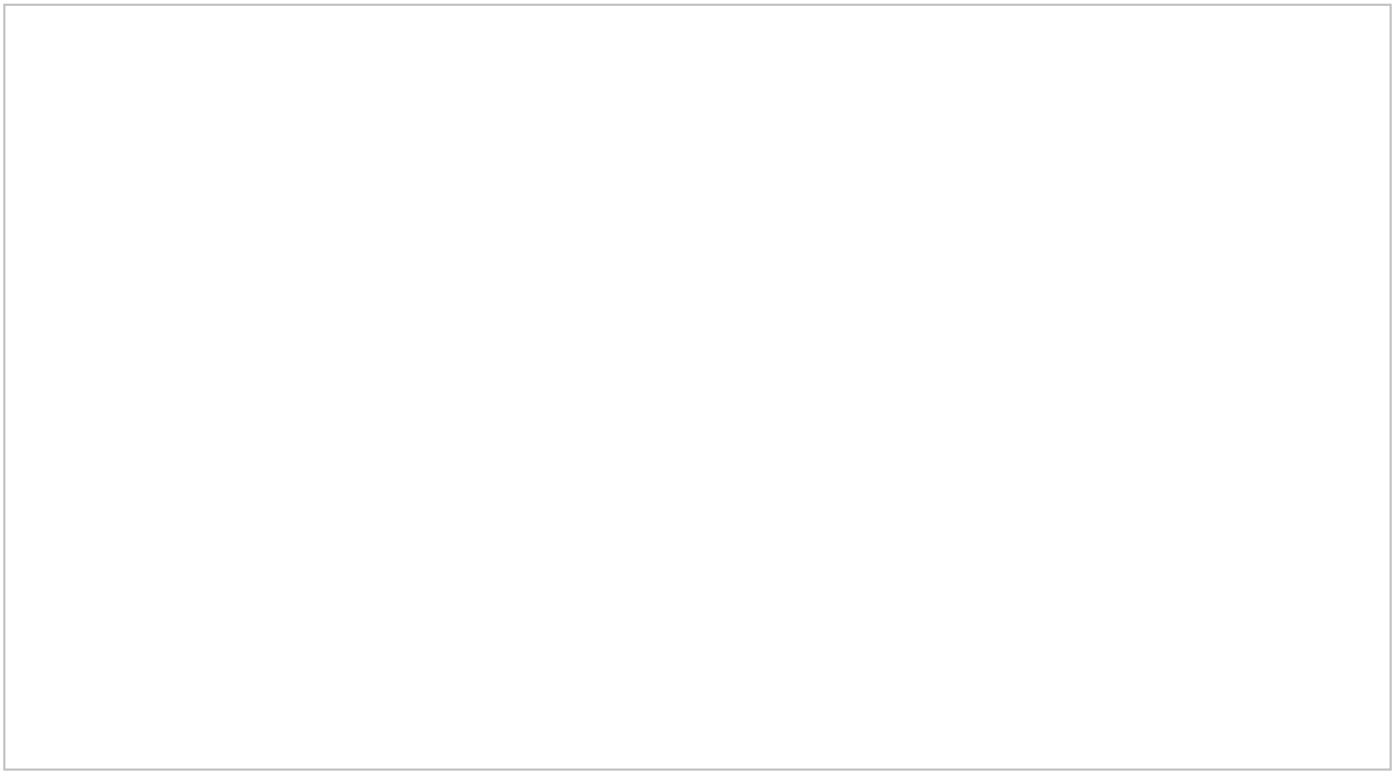
Microsoft Visual C++ 6.0 for Windows 95/98/NT4

Steve Ballmers famous "*developers, developers, developers*" speech, or rather dance, not just became an all-time internet meme. It also highlighted the focus Microsoft has on constantly attracting developers and having a strong marketshare in the software development community. While many wondered why Microsoft acquired Github, insiders very well knew that Microsoft would not want to allow anyone to rival them in the space of software development tools and services.



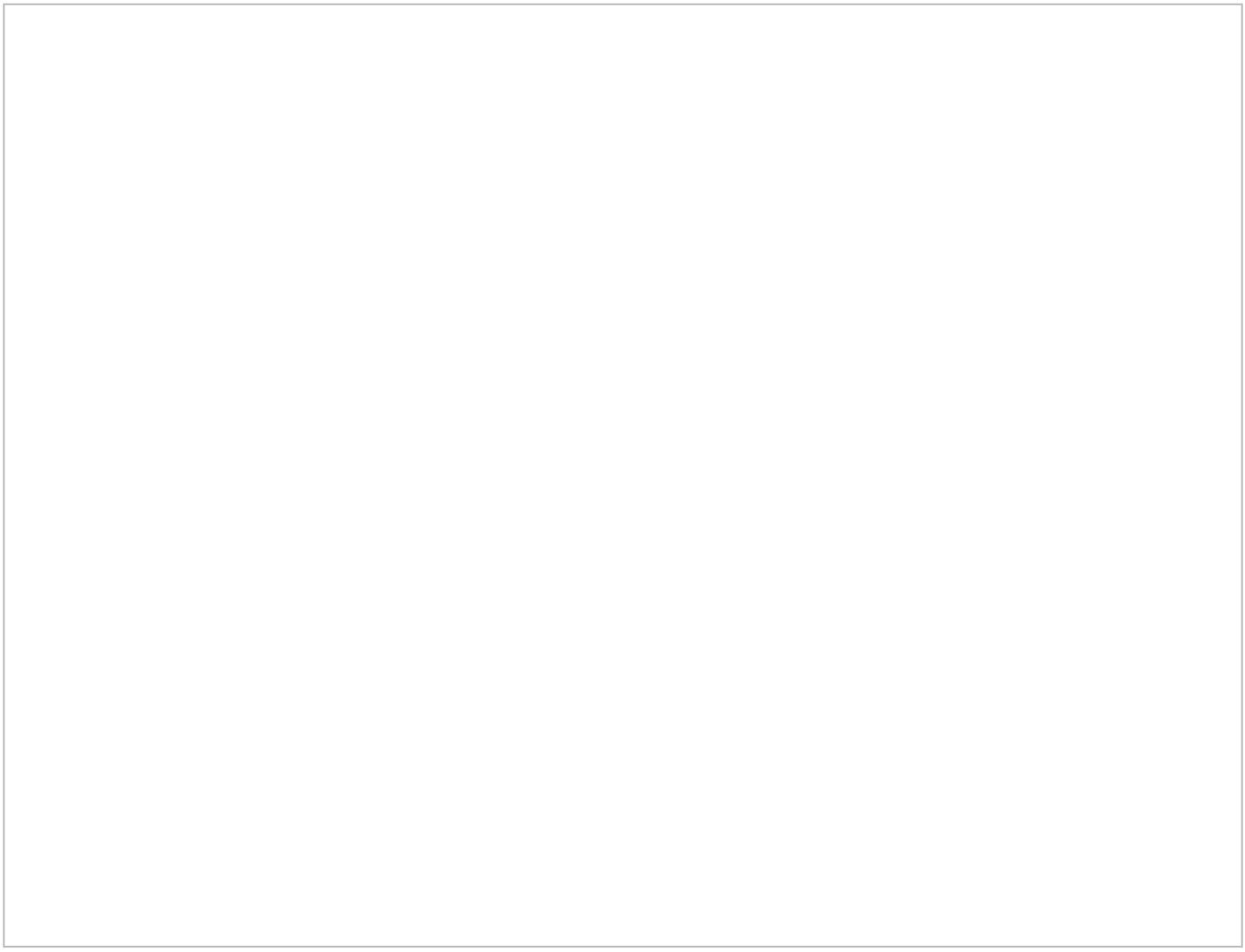
Microsoft Visual Basic 3.0 for Windows 3.1

Microsoft Windows 3 and its graphical user interface weren't really new, given that the Macintosh existed before. However, the Microsoft Visual Basic IDE was a complete game changer. Visual Basic introduced dragging and dropping user interface or control elements onto the canvas of a window. With this, Microsoft launched the era of visual programming in 1991. Developers could now visually design the application with a clear focus on the user interface and thus the user experience. It marked the beginning of user centered software design.



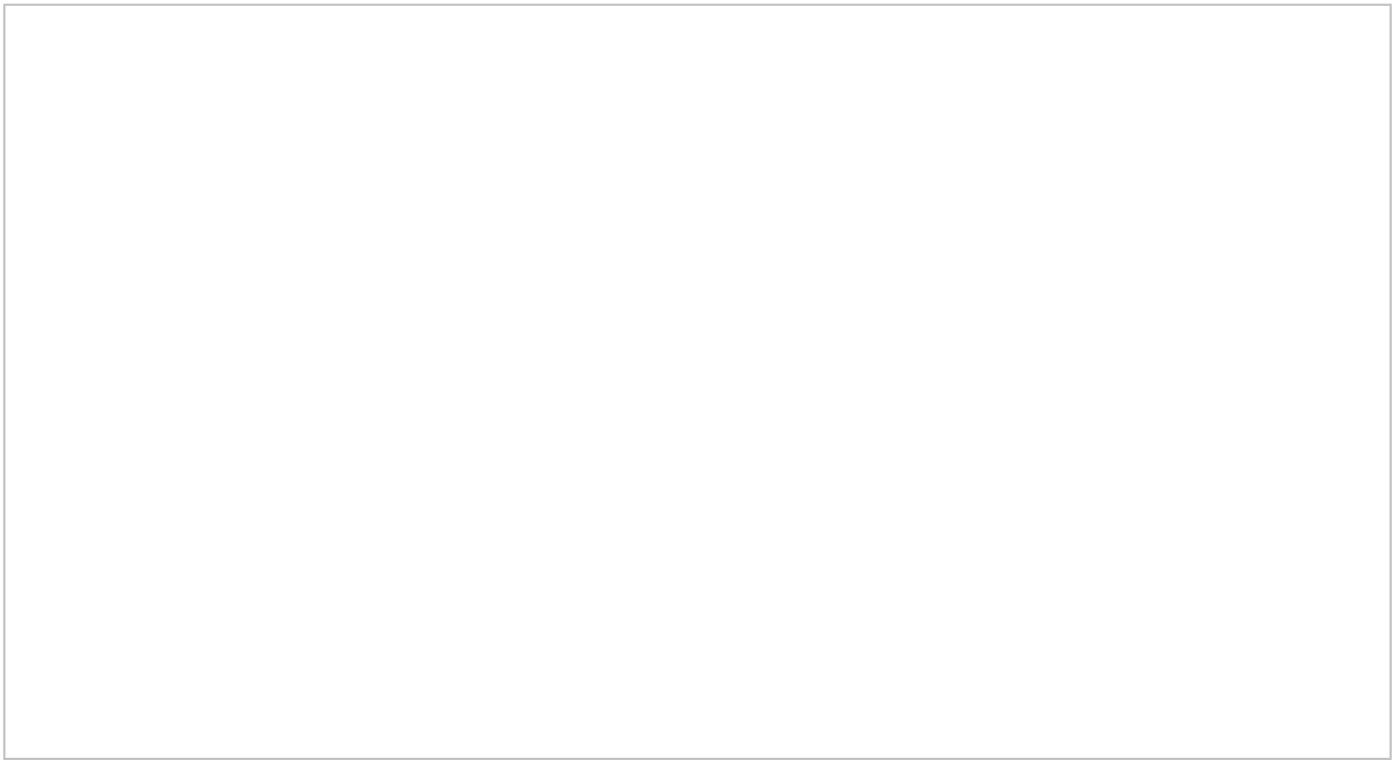
Microsoft Visual Basic 6.0 — Microsoft's breakthrough in the IDE market

Visual Studio 6 marked a major milestone for Microsoft and Visual Studio. It was the first Visual Studio with database connectivity, the ability to build applications and libraries for Windows, Pocket PC, server side applications for IIS with ASP, ActiveX controls and so much more. Visual Studio 6 was also extendable to build for other platforms like PalmOS. ActiveX and COM components allowed using 3rd party components on Windows as well.



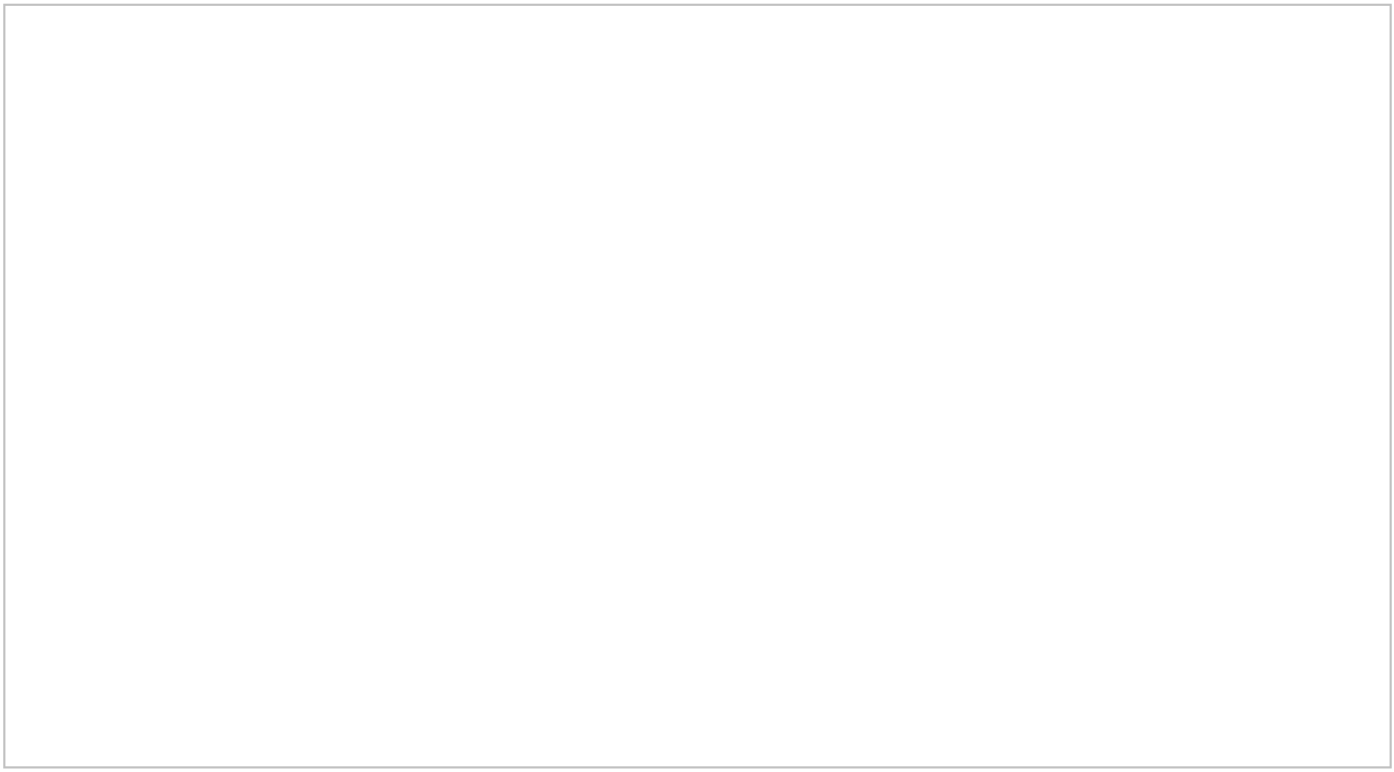
Microsoft Visual Studio .NET on Windows XP in 2002

In 2000 Microsoft released a new programming language by the name of C# that would rival the popular Java language and use the newly invented .NET framework. The .NET framework was a complete class library abstracting the Windows API. Visual Basic .NET was also released giving Visual Basic a more seamless access to Windows functionality through the .NET framework. Microsoft Visual Studio continued to evolve and became as feature rich of a monstrosity as IntelliJ IDEA was and is.



The Latest Visual Studio can easily compete with IntelliJ, both in advantages and disadvantages

Given that IDEs became ever bigger, resource hungry and heavy weight, with features and complexitity that many developers actually did not need or want, a trend to move to simpler editors slowly began in the 2010s. With editors like Sublime Text released in 2008, Github's Atom released in 2015 and Notepad++ already taking market share from 2003 onwards, there was a significant trend away from full blown IDEs back to editors. Microsoft was swift to respond to the trend and released the lightweight development environment by the name of "Visual Studio Code" in 2015.



Microsoft Visual Studio Code, “VSCode” or “Code” in short

According to the 2022 developer survey by StackOverflow, a total of 74.48% of all developers use Visual Studio Code. Microsoft, for 30 years now, continues to totally dominate the IDE market with market shares well beyond the 60% mark for over 30 years. VSCode today supports all programming languages imagineable, runs on Windows, macOS and Linux. With Github Copilot, it provides the most advanced AI-based code completion in the industry. It supports version control, debugging, extensions and has a sheer endless amount of capabilities.

No company shaped Integrated Development Environments and the way we build software more than Microsoft did in the past 35 years. That’s why Microsoft and the Visual Studio family deserve the number one spot on the list of revolutionary IDEs that shaped software development.

Thank you for reading. Jan

Did I miss any IDE? Leave a comment below.

Technology Software Development Programming Computer Science Software Engineering

Written by Jan Kammerath

Follow

2.6K Followers

I love technology, programming, computers, mobile devices and the world of tomorrow.

More from Jan Kammerath

Jan Kammerath

Technical Debt Has Taken Over The Economy

Technical debt is not just in your software, it is not even in systems and dominating...

· 9 min read · Jun 9

1.5K 35

Jan Kammerath

How Kubernetes And Kafka Will Get You Fired

Kubernetes and Kafka: dream team or horror show? Not every business can affor...

· 9 min read · Apr 19

3.1K 55

Jan Kammerath

Using AWS Like A Pro: Best Practices From Solutions...

All beginners start off with the same mistakes. Read the best practices from th...

· 11 min read · Jul 15

156 4

Jan Kammerath

How Go fixed everything that was wrong with programming

Why do you write software? I write software to build computer programs that solve rea...

8 min read · Apr 16

1.3K 29

See all from Jan Kammerath

Recommended from Medium

Rico Fritzsche in Level Up Coding

Microservices: The Million-Dollar Mistake Your Company is Making

In the middle of the last decade, as I was elbow-deep in code, tinkering with...

✦ · 5 min read · Jul 11

701 24

Ignacio de Gregorio

Microsoft Just Showed us the Future of ChatGPT with LongNet

Let's talk about Billions

✦ · 8 min read · Jul 20

2.4K 35

Lists

General Coding Knowledge
20 stories · 122 saves

It's never too late or early to start something
13 stories · 46 saves

Stories to Help You Grow as a Software Developer
19 stories · 208 saves

Coding & Development
11 stories · 76 saves

Aaron Dinin, PhD in Entrepreneur's Handbook

One Type of Founder Investors Hate Funding

You'd be surprised by what kinds of unexpected things can make investors sa...

✦ · 5 min read · Jul 17

2.9K 45

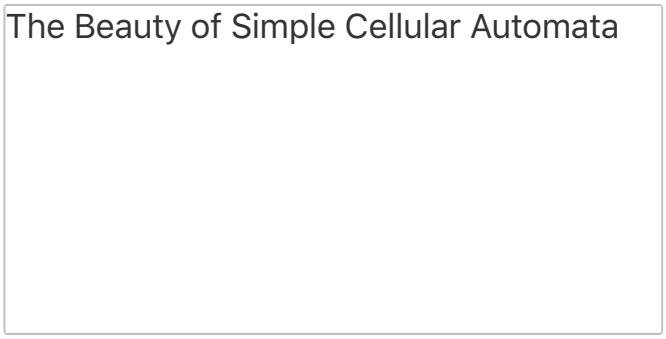
Barry Gander

Russian's Joke About Famine Is Not Appreciated Amid Huge...

Kremlin propaganda chief Margarita Simonyan's blatant admission that Russia ...

✦ · 10 min read · 4 days ago

822 17



Cole in Cantor's Paradise

Ethan Siegel in Starts With A Bang!

The Beauty of Simple Cellular Automata

You may have heard of Stephen Wolfram, founder of WolframAlpha and the math-...

· 9 min read · Jul 21

444 5

Is the Universe 13.8 or 26.7 billion years old?

For many years, cosmologists have claimed the Universe is 13.8 billion years old. A ne...

· 12 min read · 4 days ago

939 19

See more recommendations