



Coding in the Cloud

Using online development tools

A number of online integrated development environments (IDEs) have recently emerged. What benefits do these tools offer software vendors? Can these cloud-based development platforms be used to reduce costs, and improve a developer's productivity? Parity has investigated the use of various online IDEs, and identified their advantages and disadvantages for organisation developing software. We have considered the scenarios in which it makes sense to use cloud-based development tools, and the future long term impact on the existing application development software market.

Contents

Coding in the Cloud	1
Contents	2
Management Overview	3
Advantages and Disadvantages of Online IDEs	4
Advantages – Your Code Anywhere, Anytime	4
Disadvantages – Cloud Based Calamities	5
Online IDEs Compared	6
Trends in Online IDEs	8
The New Development Environment?	9
Future Trends in IDEs?	10
Conclusion	12

Management Overview

Over the past years, a number of companies have developed online alternatives to traditional desktop applications, allowing users to create documents and spreadsheets in the web browser. An online code editor is where a browser hosts a text editor, allowing a developer to create and edit source code for an application, without installing any client side software. The more advanced online tools aim to replicate the functionality of desktop IDE which compile and deploy the source code as an executable application.

These online IDEs are hosted by a provider using cloud- based storage, and are accessed by developers over a network, typically by using a web browser over the Internet. The cloud-based development platform encompasses more than simple online code editors, as other parts of the developer tool chain are also hosted in the cloud. These include source code hosting, build servers, task lists, bug tracking, and project planning, as well as continuous integration capability (i.e. automatically rebuilding the solution and running unit tests whenever code is checked in by a developer). The term Development as a Service (DaaS) is used to differentiate this emerging sector from the rest of the Software as a Service (SaaS) market.

Parity has reviewed a number of online IDEs to determine what value they can bring, and how they can be used for best advantage in developing software. These cloud based tools are marketed as replacements to the traditional IDEs like Microsoft's Visual Studio. Are they mature enough to do so, and what are the benefits of using them? What trends can we expect in development software as a result of online IDEs? This report outlines the findings from Parity's research, and what they mean for software vendors.

The Software Development Market

There are currently over 15 million professional developers (Evans Data, 2009), with over 1.3 million in the USA alone (Bureau of Labor Statistics, 2010), and a further 330,000 in the UK (Office for National Statistics, 2010). The application development software market is estimated as being worth over \$7 billion annually (Gartner, 2009).

The majority of software developers currently use either Microsoft's Visual Studio or a variant of the open source Eclipse (Forrester, 2008), both of which are traditional desktop applications. Licensing costs for a commercial desktop IDE can be a substantial overhead for organisations, with the current cost for a single Visual Studio 2010 Professional license starting at \$549. Eclipse is open source software, but many developers make use of commercial versions of the Eclipse IDE, such as the IBM Websphere IDE

Advantages and Disadvantages of Online IDEs

Before comparing the online IDEs, it is worthwhile considering their advantages and disadvantages, compared to existing desktop IDEs.

Advantages – Your Code Anywhere, Anytime

This is the tag line for the Cloud9 development platform (Figure 1 below), and it reflects the fact that an online IDE solves a number of problems that the traditional desktop IDE cannot:

- 1. Platform Independence.** An online IDE requires only a browser and an Internet connection. The developer can code from a PC, a net book or a smartphone – they are no longer tied to a specific device.
- 2. Accessible from Anywhere.** The code is available from any location with a broadband connection, allowing the developer to access their project with only a few browser clicks.
- 3. Collaboration.** An online IDE offers additional opportunities for team work, i.e. allowing pair programming between developers on different continents. This is of particular importance when using outsourced development teams.
- 4. Improved Productivity.** By having a consistent developer environment that requires only a browser to access it, a developer can focus on developing and not on configuring a traditional desktop development environment.

- 5. Reduced Licensed Costs.** Using a cloud based development environment allows organisations to use a „pay as you go“ model for their development environments, and only pay for the development environments they need. This is particularly useful for organisations that make use of contract developers, where bringing on a contractor will no longer require additional costs for new licenses for their development environment.

Integrated development environments

An integrated development environment (IDE) is a software application that provides comprehensive facilities to programmers for software development. An IDE normally consists of:

- A source code editor
- A compiler and/or interpreter
- Build automation tools
- Debugging tools

Typically, version control system(s) and graphical user interface (GUI) design tools are also integrated into the IDE. IDEs have developed into complex software applications that are designed to maximize programmer productivity by providing related components and tools in a single user interface.

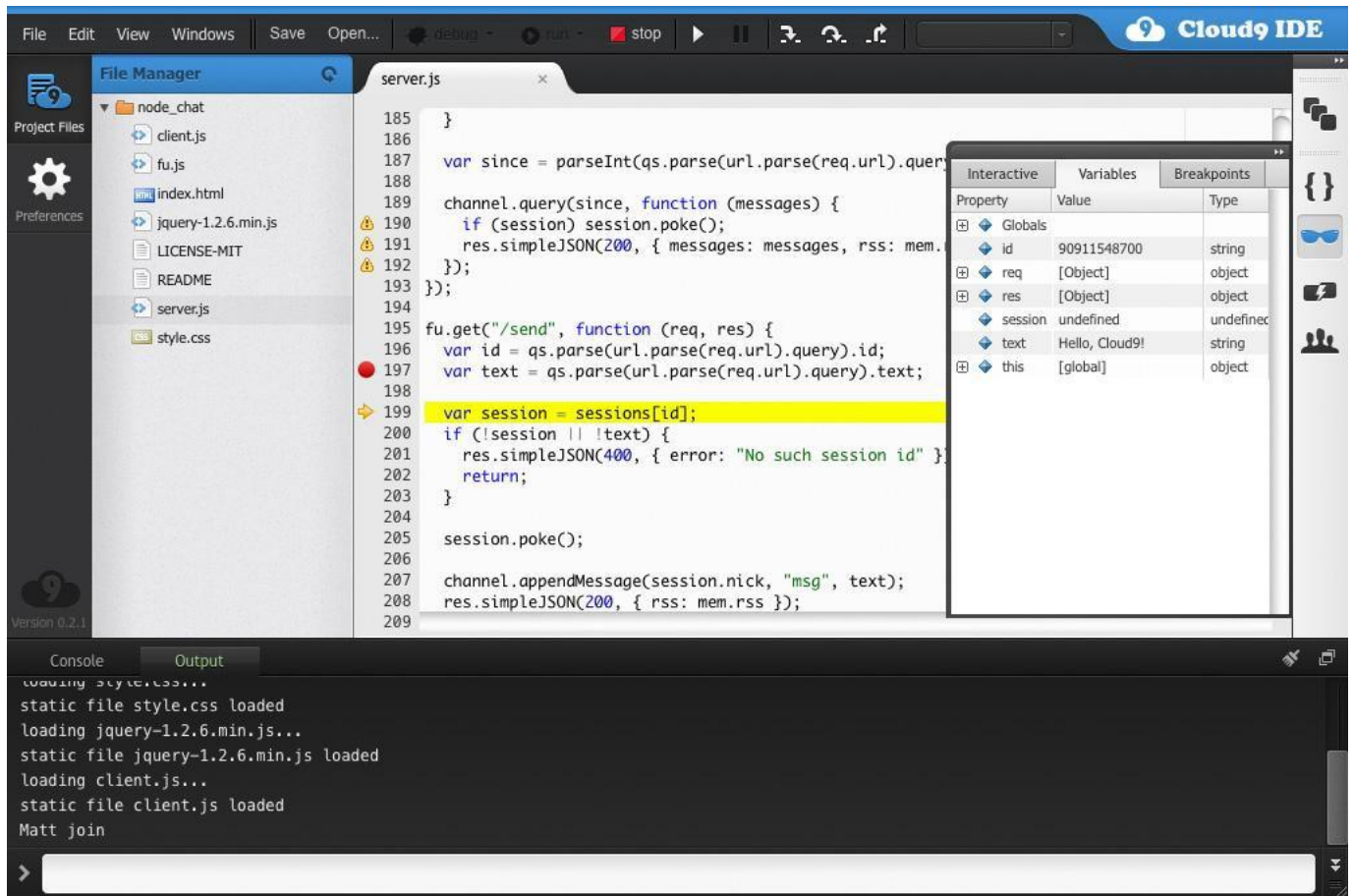


Figure 1 – The Cloud9 IDE

Disadvantages – Cloud Based Calamities

Online IDEs also have several shortcomings when compared to desktop IDEs:

1. Code Security. By default, your source code will be visible to whoever manages the IDE and this will prevent cloud based tools being used to develop sensitive software.
2. Data Loss. By using an online IDE, you explicitly trust a third party to manage your source code and development environments. How do you cope if the third party exposes your source code accidentally?
3. Service Continuity. Similarly, how does your organisation cope if the online IDE is unavailable for any length of time? An SLA is not insurance against this happening and damaging your company.
4. Broadband Connectivity. An online IDE requires a fast internet connection. How does a developer remain productive in the absence of a broadband connection? How useful are cloud based tools when the speed of the broadband service is reduced?

Online IDEs Compared

Parity reviewed five of the most popular online IDEs currently available (June – July 2011). The popularity of the IDEs was determined by the number of references made to the tools on programmer websites such as StackOverflow, InfoQ and DZone.

The IDEs were compared on the following points:

- What programming languages are supported?
- What support is given to developers to write and debug code?
- Can an existing code project be modified using the IDE?
- Can third party code libraries be referenced?
- Offline capability – can the developer continue to be productive even in when there is no internet connection?
- Can the IDE be customized or is there a plug-in framework that allows the IDE to be extended?
- Is there integration with database and source control systems?
- Is the created source publicly searchable by default?

The results of the IDE comparison are given below in Table 1:

Feature	CodeRun	Kodingen	Complir	Cloud9	Orion
Languages Support	C#, PHP, JavaScript, CSS	PHP, Ruby, Python, ColdFusion, Classic ASP	C#, VB.NET, Java, Ruby, C++	HTML5, JavaScript, CSS, Ruby, PHP, Python, OCaml, Clojure	HTML5, JavaScript, CSS
Syntax Highlighting	✓	✓	✓	✓	✓
Code Completion	✓	✗	✗	✗	✗
Debugger	✓	✗	✗	✓	✗
Upload Existing Projects	✓	✗	✗	✓	✓
Reference Third Party	✗	✗	✓	✗	✗
Database Support	✓	✓	✗	✗	✗
Version Control Integration	✗	✗	✗	✓	✓
Code Private?	✓	✓	✗	✗	✓
Cloud Hosting	✓	✗	N/A	✓	✓
Offline Capability	✗	✗	✗	✓ Full	✓ Local Install
Customization or Plug-in	✗	✗	✗	✗	✓

Table 1 – Comparison of Online IDEs

Trends in Online IDEs

From the results of Parity's comparison of online IDEs, the following trends can be seen:

- Most of the IDEs are aimed at web development only, with only one IDE (Compilr) focused on desktop development.
- The main language support is for dynamic web languages (JavaScript, PHP, Ruby and Python).
- There is limited support for the .NET and Java languages, which are predominantly used in enterprise organisations.
- Several IDEs allow developers to upload existing code projects, but there is again limited support for referencing third party code libraries.
- There is limited support for database and version control integration.
- Most IDEs offer the option of development and/or production hosting of websites.
- Only one IDE (Cloud9) offered a full offline capability, while a second (Orion) offered a locally hosted option to avoid relying on an internet connection.
- Only one IDE allowed significant customization, and it also offered a plug-in framework to allow developers to extend its functionality. This was Orion, the only open source online IDE currently available. (Note, the Cloud9 source code editor is open source software, but the actual IDE is proprietary).

The New Development Environment?

From Parity's investigation of online IDEs, it is clear that they are not currently mature enough to replace existing desktop IDEs. The feature set offered by an online IDE isn't complete enough to allow developers to be as productive as they are with an existing desktop IDE such as Visual Studio or Eclipse. The absence of an extensible plug-in framework and lack of customization options means that developers are unable to customize the „vanilla“ offering from an online IDE for their specific roles. Additionally, most of the IDEs have not developed an offline mode, or addressed the important code security concerns that exist.

Our view is that online IDEs will be used as a supplementary IDE for developers to experiment with new languages and paradigms, and to develop proof of concept applications without investing in new desktop tooling. They will also be used when the developer is unable to access their current desktop IDE, when on the move, working on client site, etc.

In addition, an online IDE could be used in specific team working scenarios, i.e. where a contract developer is brought on to a team for a short period. Instead of having the contractor waste time configuring a desktop environment, an online IDE could be used to quickly allow him/her write productive code. Online IDEs could also allow developers to collaboratively write code in different locations, i.e. „pair programming“ at a distance. This could also be useful in telephone interview situations, i.e. present a candidate with a coding scenario delivered by passing a URL, and give them time to develop a solution, whilst monitoring the code being developed in real-time.

Even when online IDEs successfully resolve the shortcomings listed above (see Cloud Based Calamities?), it is unlikely that they will be able to replace desktop IDEs. The latency issue posed by relying on a network connection means that online IDEs will be unable to match desktop IDEs for performance. They also offer developers less opportunity to customise and extend functionality, which is contrary to the existing trends in IDE usage (Forrester, 2008).

Online IDEs will be taken up by the hobbyist programmer and programmers for specific niche areas, such as developing mobile applications.

Future Trends in IDEs?

Whilst online IDEs are not able to replace existing IDEs at this point in time, they point towards the trends in development tools. In particular, online IDEs answer several major shortcomings of desktop IDEs:

1. Platform Independence and Accessibility. Developers require ways to access code that don't limit them to using a single desktop PC or laptop. Online IDEs free developers from this reliance.
2. Collaboration. With changes to the work environment (greater use of outsourcing, development teams spread across several locations, use of contract staff), there is a requirement for more collaborative working between developers.
3. Improved Productivity. Organisations want their developers to spend more time delivering business value by coding, and less on administrating their development environments.
4. Upgradeability. By focusing on specific technology stacks (web development, mobile phone applications), online IDEs are more focused than traditional IDEs, and can be more quickly updated to respond to changes in the development paradigms. For example, developers using a desktop IDE have to upgrade every instance of it that is installed. Online IDEs require a single update of the server installation to make new features available for all developers using it.

Future releases of existing desktop IDEs will have to address these issues, or face having their market share reduced by the new online IDEs. This would point to the following features being offered in new IDEs:

- Cloud Synchronized Configuration. Once a developer has setup their IDE, they will simply synchronize their custom configuration settings, including their various extensions and plug-ins, to a cloud host. Whenever they move to another device that has the IDE installed on it, they will simply login to the cloud hosted configuration, and the settings on the new machine will be

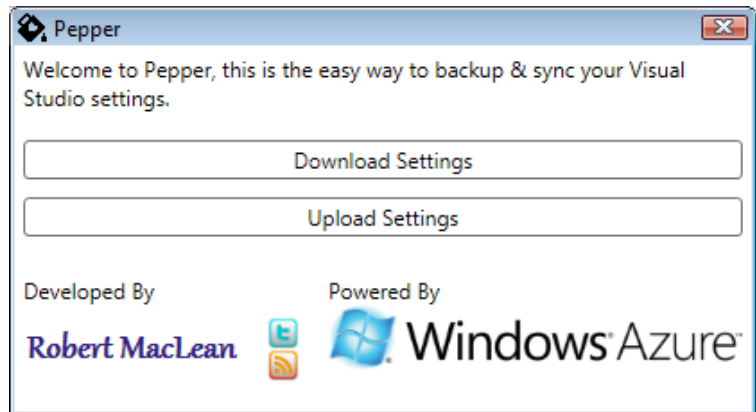


Figure 2 – The Pepper extension for Visual Studio

automatically synchronized. This is already possible on Visual Studio using an open source extension called Pepper, which allows developers to synchronize their Visual Studio settings across multiple PCs (see Figure 2).

- Configuration Comparison and Automatic Updates. A major issue with software development is the “works on my machine” issue. Because of differences in a developer’s environment setup, a software project that compiles and executes successfully on one machine may fail on another, due to a difference in the IDE configuration or a mismatch in code libraries, etc. Using the above cloud stored configuration settings, IDEs will in future allow developers to compare their IDE setups and identify issues that prevent software building successfully. Once the configuration differences are identified, the IDE will prompt the user to synchronize the setting to resolve the issue. Optionally, the dependency will be automatically recorded in the project, and when another developer attempts to open the project, they will be prompted to update their development environment to the required configuration.
- Online Versions of Desktop IDEs. Desktop IDEs will include the option to host online versions of the IDEs. Instead of the security threat of using a third party online IDE, the existing IDEs will allow developers to use locally hosted online IDEs. For an analogy, think of Microsoft Exchange – as well as powering the desktop Outlook client, there is also the option to use the Outlook Web App to access your emails. Similarly, the commercial desktop IDEs will have to provide online (and mobile) clients for their products. These online and mobile versions will not offer the full capability of the desktop client, but they offer enough features to allow a developer to remain productive whilst on the move.

- Improved Team Working and Collaboration Features. Greater team working features to be built-in to the existing IDEs. Instead of using a separate application or an online IDE to work together with a developer in a different location, the desktop IDE will mature to allow developers to work collaboratively on the same code file and to show the changes made by the other developer in real time.
- Support for Online Toolset. Additionally, desktop IDEs will allow you to swap elements of their toolset for online hosted versions, i.e. having built-in integration for online code hosting providers like BitBucket and GitHub, automatic deployment of websites to the main cloud hosting providers, etc.
- Faster Upgrades and Greater Extensibility. Software tools will mature beyond the traditional release model, where a new version of the IDE is released every few years, and a service pack appears every 6 months, bundling together all patches and updates. Instead, the IDE will include an automatic update facility, allowing new features to appear in the IDE every time it is started. This is already starting to happen, with use of the NuGet package manager in Visual Studio allowing libraries and tools in Visual Studio to be easily added and updated (see Figure 3).

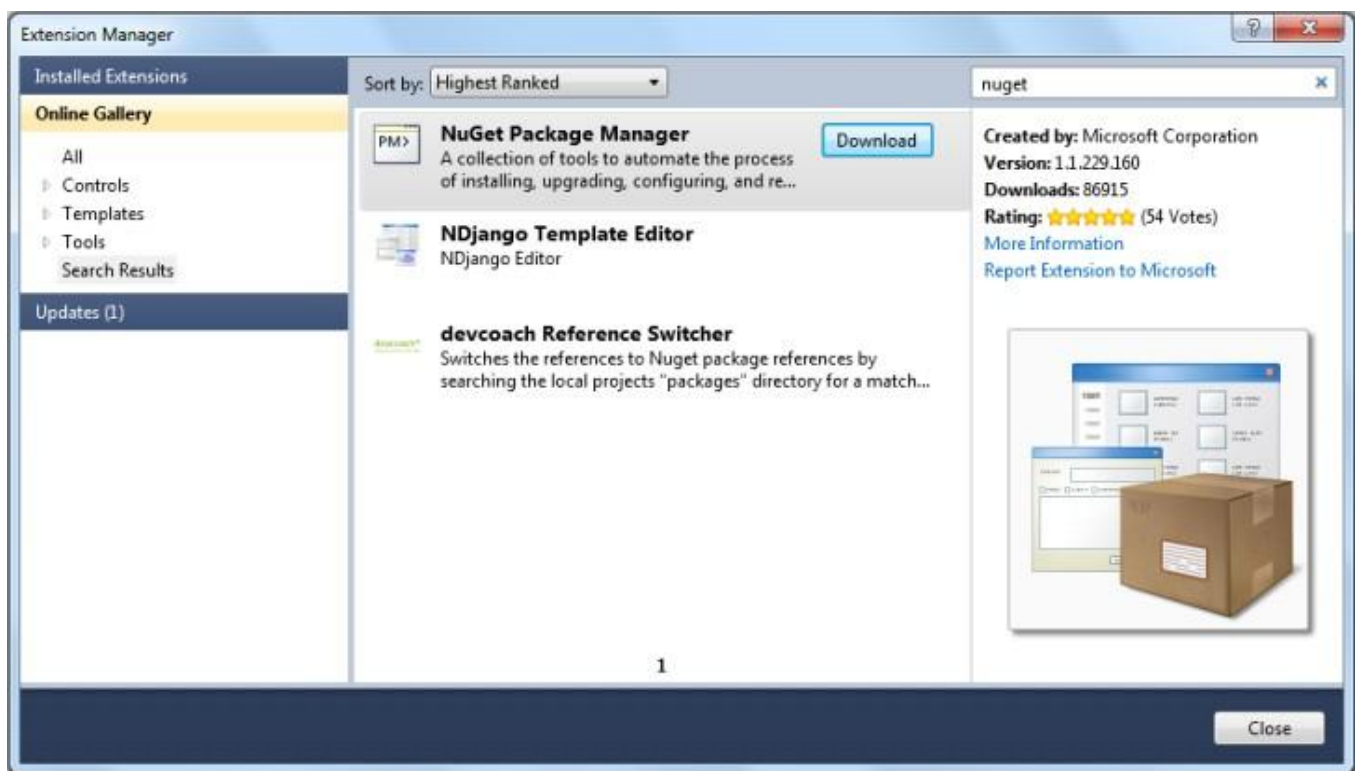


Figure 3 – The NuGet Package Manager for Visual Studio 2010

Conclusion

Online IDEs are not currently a viable alternative to existing desktop IDEs. But they will be used in specific scenarios, as a supplementary tool to a primary desktop IDE. The advent of online IDEs will also force existing IDE vendors to address the issues that the online IDEs have resolved:

1. Platform Independence and Accessibility
2. Collaboration
3. Improved Productivity
4. Upgradeability

In time, the features offered by online IDEs will be incorporated into existing IDEs, so that IDEs will be increasingly connected to cloud based services (such as source control, cloud based storage, build tools and environments), as well as offering more collaborative and extensible development environments.

Parity Systems

Wimbledon Bridge House

1 Hartfield Road

Wimbledon

London SW19 2RU

For further information

Call us on 0845 873 0790

Email marketing@parity.net

or visit us at www.parity.net