



IronRuby Shines in New .NET Development Tool

Overview

Country or Region: Australia

Industry: Software development

Developer Profile

Software developer James Crisp is a Senior Consultant and the .NET Lead at the Sydney, Australia office of global IT consultancy ThoughtWorks.

Business Situation

Crisp wanted to automate software builds while working in Microsoft Visual Studio and the .NET Framework, but found existing XML-based build tools inflexible and hard to scale.

Solution

Crisp developed a build tool called nRake, which uses IronRuby, a Microsoft implementation of the popular Ruby programming language.

Benefits

- nRake makes software builds easier to specify and more scalable.
- Developers have a more powerful .NET development tool.
- IronRuby helps bring Microsoft development technologies to the open source community.

“Ruby developers can now access the enterprise-level features of the .NET environment such as managed code and standard APIs, which allow them to develop software once and then run it anywhere.”

Nick Hodge, Professional Geek, Microsoft Australia

Software developer James Crisp was looking for a way to automate software builds while working in Microsoft Visual Studio and the .NET Framework, but he found existing XML-based build tools inflexible and hard to scale for some of his larger projects. He decided to create nRake – a .NET-compatible version of Rake which is a popular build tool based on the Ruby programming language. Using IronRuby – Microsoft’s implementation of Ruby for the .NET environment – helped Crisp take advantage of Ruby’s intelligence and versatility, allowing software builds to be specified in a way that was easier to understand and more scalable. IronRuby helps connect the open source and Microsoft development communities, making it easier for Ruby developers to use .NET’s enterprise-class capabilities, while giving .NET developers access to a popular and powerful programming language.

Microsoft*

“Ruby is very popular with developers, especially in the open source world; however, many large business and government customers only want their software developed in what they see as enterprise-class environments, which means either .NET or Java.”

Nick Hodge, Microsoft Australia.

Situation

Software developer James Crisp is a Senior Consultant and the .NET Lead at the Sydney, Australia office of ThoughtWorks, a global IT consultancy. ThoughtWorks develops custom applications and provides consulting and training services, with expertise in Agile and Lean development principles. Crisp has worked in software development, IT architecture and consulting for more than a decade.

A core part of the Agile methodology is iterative and incremental development. As a result, it is important to have a consistent and easily replicated process for building software from its various code components – a task that is performed many times daily by all developers and automated build servers.

Developers working in Microsoft’s Visual Studio environment typically use build tools such as Microsoft’s MSBuild – part of the .NET Framework – or open source tools such as NAnt. These tools use XML (extensible markup language) to define build components and processes. For Crisp, these tools lacked flexibility.

“Once you get up to a fairly high level of complexity, these XML structures become quite difficult to maintain and extend,” says Crisp.

In 2008, Crisp became interested in using Rake – an open source software build tool that uses the Ruby programming language rather than XML to define software builds. This enables developers to use the full power of an advanced object oriented scripting language to write your builds.

“Ruby has the ability to almost transform itself into another language,” explains

Nick Hodge, Professional Geek at Microsoft Australia. “You can generate a domain-specific language to describe something like a software build in a way that makes sense and is easy for both humans and computers to read.

“Ruby is very popular with developers, especially in the open source world. However, many large business and government customers only want their software developed in what they see as enterprise-class environments, which means either .NET or Java.”

Solution

In 2009, Crisp started work on nRake – a complete package for .NET builds using Rake. His objective was to make the capabilities of Rake readily available to .NET developers.

“I had some time between clients so I started putting it together,” says Crisp. “nRake is based on a ‘batteries included’ philosophy – you don’t need any additional libraries or downloads, and all the plumbing work has been done for you. It can help you get up and running very quickly on a new project and provides a strong and flexible build scripting environment, which you can extend to suit your needs.”

nRake includes a template C# project with unit tests, IronRuby, the Rake build tool, the Rake Albacore gem, a default Rakefile build, templated configurations for different environments, configurations for automated build servers and the open-source NUnit software testing tool.

“IronRuby is an implementation of Ruby for the .NET Framework, which Microsoft announced in 2007 and launched in

“Using IronRuby, your build script can talk directly to the .NET code. So, for example, you can directly call .NET code and libraries from build tasks.”

James Crisp, Senior Consultant and .NET Lead, ThoughtWorks

2008,” says Crisp. “The first version of nRake used Ruby 1.9. About six months later, the performance and start up time of IronRuby was comparable to Ruby 1.9, so I moved nRake over to IronRuby.”

Says Hodge: “A lot of the developers working in Ruby are in the Linux, UNIX and Mac OS worlds, but there are also plenty of advanced developers in the Microsoft environment who really like Ruby. IronRuby gives Microsoft developers the best of both worlds – the flexibility of Ruby and the power of the .NET environment.”

Benefits

A powerful automation tool for developers

IronRuby has helped Crisp make nRake a valuable automation tool for developers working on large-scale .NET development projects, especially when using the Agile methodology.

“Using IronRuby, your build script can talk directly to the .NET code,” says Crisp. “So, for example, you can directly call .NET code and libraries from build tasks, something which is not possible using Ruby 1.9.

“Using IronRuby means that the nRake download is smaller and faster than using Ruby 1.9. Using IronRuby also makes nRake an attractive option for companies which focus on Microsoft technologies.”

As a result, nRake has quickly gained international attention.

“I’ve received a lot of interest from the ALT.NET community – a worldwide group of developers who are interested in ways of combining open source tools with the

.NET environment,” says Crisp. “One Sydney digital advertising and product development house I have spoken to is already using nRake.”

Integration between Microsoft and open source technologies

nRake and a range of other projects using IronRuby have demonstrated how Microsoft and open source technologies can connect.

“IronRuby brings the power and flexibility of Ruby to developers who are more familiar with the .NET environment,” says Hodge. “At the same time, developers who are comfortable working with Ruby can now access the enterprise-level features of the .NET environment such as managed code and standard APIs, which allow them to develop software once and then run it anywhere – a desktop, mobile device, server or in the cloud.”

According to Robert Evans, Platform Strategy Lead, Microsoft Australia, IronRuby is an example of Microsoft’s commitment to supporting developers and working with a wider range of non-Microsoft technologies, including open source software.

“The open source community and the Microsoft developer community have traditionally been at odds but IronRuby and similar projects are helping to bring them together,” he says.

“In addition, Microsoft gets to give back to the community by releasing source code and contributing to open source projects.”

For More Information

For more information about IronRuby, please visit the website at:
<http://www.ironruby.net/>

To learn more about and download nRake, please visit the website at:
<http://github.com/jcrisp/nRake>

Microsoft .NET

Microsoft .NET is software that connects people, information, systems, and devices through the use of Web services. Web services are a combination of protocols that enable computers to work together by exchanging messages. Web services are based on the standard protocols of XML, SOAP, and WSDL, which allow them to interoperate across platforms and programming languages. .NET is integrated across Microsoft products and services, providing the ability to quickly build, deploy, manage, and use connected, secure solutions with Web services. These solutions provide agile business integration and the promise of information anytime, anywhere, on any device.

For more information about Microsoft .NET and Web services, please visit these websites:

www.microsoft.com/net
msdn.microsoft.com/webservices

Software and Services

- Microsoft Visual Studio .NET 2003-2010
- nRake
- Iron Ruby

This case study is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Document published October 2010

Microsoft[®]