OpenJDK

Get Collaborative

Oracle's **Donald Smith** discusses the OpenJDK community, the place to collaborate on Java SE. **BY JANICE J. HEISS**

Donald Smith has spent more than a decade serving the Java community. After working as an engineer, consultant, and product manager for several information technology companies, he joined Oracle in 2002, where he served until 2005 as director of technology evangelism. He next moved to the nonprofit Eclipse Foundation, where he was director of ecosystem development from 2006 to 2011. Smith currently works for Oracle Canada ULC on the product management team for Java SE, where he is primarily responsible for the OpenJDK community.

When he returned to Oracle in May of 2011, Smith remarked in his blog about his new job, "The team I'm on has one simple mandate—keep Java the number one computing platform in the world."

Java Magazine: What is OpenJDK?

Smith: Basically, OpenJDK is the place to collaborate on an open source implementation of Java SE and related projects. It's distinct from some of the other open source Java communities, some of which are hosted at Oracle and some of which are hosted elsewhere. So, for example, the GlassFish community is a place to collaborate on Java EE. Eclipse is a place to collaborate on Java tools and runtimes at the Eclipse Foundation. The NetBeans community is a place to collaborate on Java tools at Oracle, and so on. There are lots of different communities in the Java ecosystem, and OpenJDK is the one for doing Java SE work.

Java Magazine: Give us some perspective on the history of OpenJDK, which goes back to 2006 when Sun initially open sourced Java.

Smith: It was first announced in November 2006, and in 2007 the actual project was created, so people could access the code. From 2007 to 2009, a lot of formation took place as more committers joined. And then, in 2009, Oracle began the process of acquiring Sun, which officially closed in January 2010, and that gave Oracle a chance to reveal its commitment to OpenJDK and ongoing plans.

PHOTOGRAPHY BY BOB ADLER

The difference between OpenJDK and Oracle JDK

Want to collaborate on an open source version of Java SE? Head to OpenJDK. Both individuals and companies (Apple, IBM, Oracle, Red Hat, and more) contribute code to OpenJDK. Oracle uses OpenJDK as the code base for Oracle JDK, and other organizations—particularly Linux distributions—produce builds of the Java platform based on OpenJDK sources for their distributions. Oracle JDK is a specific implementation of the JDK that is produced by Oracle. Although the code bases are very close, Oracle JDK has some minor differences. For example, Oracle JDK includes closed-source third-party components like a graphics rasterizer, third-party fonts, and additional documentation. Over time, Oracle intends to open source as much of Oracle JDK to OpenJDK as it can, except commercial features (such as Oracle JRockit Mission Control). Most of this open sourcing work involves replacing closed-source third-party components with open source alternatives or negotiating licensing agreements with third parties.

Why use one over the other?

From a feature perspective, there is not much difference between Oracle JDK and an OpenJDK build. Oracle publishes up-to-date and fully tested JDK binaries for popular development and server platforms. For most end users and developers, this is the most convenient way to get Java. If you are an enterprise user and need long-term support, or you want the most-predictable binary distribution for an ISV application, Oracle JDK is the better choice of the two. If you want to build, create, and tweak your own JDK, perhaps to address a niche performance variable or to support a specialized platform, you will want to use OpenJDK as your code base.

—Tori Wieldt

Unfortunately, I think some people assumed the worst during the quiet period of 2009 during the acquisition, which as a bystander at the time seemed to take forever. But as soon as they could, Oracle made it clear that we were serious about OpenJDK. We wanted it to be the place to do Java SE development, first and foremost, and we wanted a lot more diversity, which means having a lot of the key Java stakeholders participate.

So Red Hat has been participating for several years, and companies like IBM and SAP have joined as well. It was also great to have Apple announce with us that they were going to contribute some code with the Mac OS X port. And it's not all about corporate participation—a lot of individuals participate in OpenJDK as well and they are very important to the project. So I think a lot of people have been happy to see that Oracle is serious about doing Java SE development in the open. SUSE, a business unit of the Attachmate Group, has recently joined the OpenJDK project.

In 2011, we started to get some momentum going once the organizational structure was more established. We got the governance board up and running, and adopted a new governance model, which describes the development process. And we also launched Java SE 7 in 2011. The reference implementation for Java SE 7 was based on OpenJDK sources exclusively, so OpenJDK 7 became the reference implementation for Java.

When OpenJDK was first started (in JDK 6), much of the development was done internally inside Sun and pushed out somewhat haphazardly. The good news is that both the JDK 8 project and the JDK 7 updates project started in the open, which is an important step as we find our stride.

Also in 2011, we showed some more momentum by having Twitter, Azul, and a few others announce their participation.

Java Magazine: Can you give us some concrete numbers that shed light on OpenJDK diversity?

Smith: First, let's agree that diversity is important for open source projects. One of the lessons I learned at Eclipse is that diversity is an important metric for the quality and longevity of projects. Optimally, you want to have a diverse mix of types and sizes of organizations and some key individuals participating in an open source project. It's also a plus to have diversity based on geography, experience, and other factors.

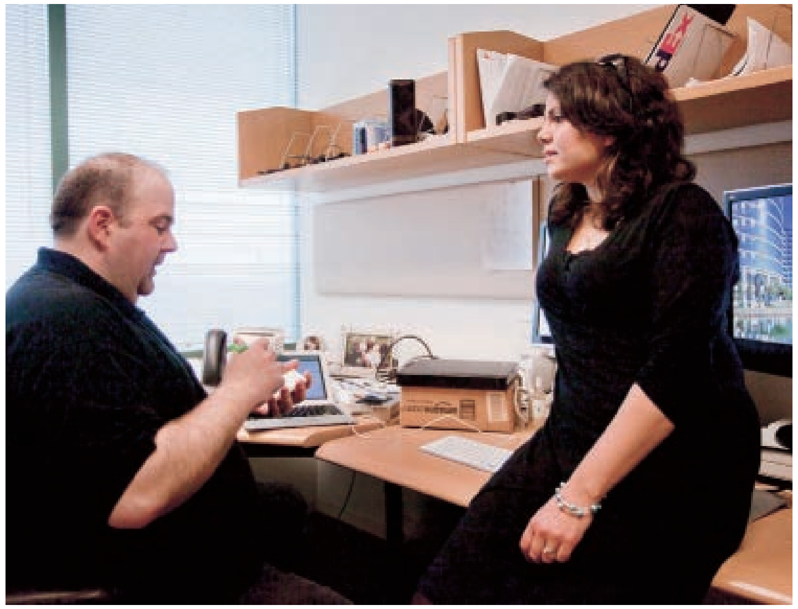
For OpenJDK, the initial code base and contribution came from Sun and then Oracle so we started from a position of low diversity, which is typical of many open source communities that start with a large initial contribution. But we want to keep pushing in the direction of more diversity. We can debate how fast and how far we move toward this, but pretty much everyone in the community agrees that we seek diversity.

THE GOOD NEWS

Diversity is important for open source projects. The good news is that we have about 350 developers in the OpenJDK community, and about 80 of those are not employed by Oracle. This is a good start.

The good news, based upon the census taken when the bylaws were approved in mid-2011, is that we have about 350 developers in the OpenJDK community, and about 80 of those are not employed by Oracle. This is a good start. We can also look at diversity in terms of e-mail traffic—about 40 percent of the e-mail traffic, according to MarkMail, is from non-Oracle employees.

Java Magazine: Can you talk about the OpenJDK community Technology Compatibility Kit [TCK] for Java SE 7?



Left to right: Oracle's Donald Smith, director of product management, and Dalibor Topic, principal product manager, OpenJDK, make plans for an OpenJDK community event. Smith and Cecilia Borg, OpenJDK onboarding program manager, talk about recognizing achievements in the OpenJDK community.

Smith: The OpenJDK community has access—through a TCK license agreement specifically for the OpenJDK community—to a compatibility test kit to validate that their builds of OpenJDK are compatible with the Java SE specification. This license is free of charge, but you have to sign the agreement, and we have to process it, which can take a few weeks. You can find the agreement on the OpenJDK Website. We're trying to keep Java compatible.

Java Magazine: What's happening now in terms of industry participation in the OpenJFX community?

Smith: We announced at 2011 JavaOne that we were going to begin the process of open sourcing JavaFX, and we've made strides with initial code contributions, and now we're looking to ramp up community participation. Several organizations are making strategic use of JavaFX, and we're hoping to see more participation in the project. We anticipate having some exciting announcements related to JavaFX at JavaOne this year.

Java Magazine: Tell us about the various groups who are a part of OpenJDK.

Smith: The notion of a *group* is codified in the OpenJDK bylaws. A group is a collection of participants interested in engaging in an open conversation about a common interest. There are, for example, groups that focus on the core libraries, the compiler, and security. One of the more topical areas right now that my team is focusing on is the quality group, where we want to make it easier for people to submit and run their own test cases and test harnesses. There is a group related to build. The challenge with both of these groups, and a number of our activities, is that we are trying to push the infrastructure along to keep up with the demands of the community. We're working on a better bug reporting system, but we also need a better build and test story so that people can more easily get their own builds up and running.

SOLID FUTURE

I believe we've established Oracle's fundamental commitment to Java through OpenJDK. We have a roadmap for releases, a Java Virtual Machine strategy for HotSpot and Oracle JRockit, and many key industry participants.

Luckily, we are getting help. The London Java Community has done a great job of pushing along information about how to do builds and providing very constructive and timely feedback on these topics. And there's a developer from SAP named Volker Simonis who has done a fantastic job at documenting his own experiences building OpenJDK on a number of platforms. There are lots of stars in the community, but we're always looking for more!

Java Magazine: OpenJDK contributors now include such companies as Apple, and as you mentioned, IBM and Twitter. It's noteworthy that Twitter joined, even though they are not a company that gets their revenue from software licensing—they're an end user of Java.

Smith: This is a very important point. When I was at Eclipse, we noticed something that academics and others have written about. There are different waves of participation in open source communities. The first wave is usually one or two organizations that have a significant code base that, for whatever reason, they're interested in making open source. They are usually ISVs [independent software vendors], or other enterprise software companies, that make their revenues directly from software licensing.

Then, after the project has proven itself, a second wave of participation occurs, consisting of companies that are software companies but don't generate their revenue directly from software licenses. So software to them is not their

Smith maps out an upcoming presentation.

product. Twitter is a great example. They build software—their biggest expense is probably software development—but they don't sell software per se. They sell a service and generate revenue from advertising and data intelligence. They're savvy with software but don't sell software licenses.

And then the third wave of participation is another degree removed from that—banks, insurance companies, maybe shipping companies, retailers, and so on. These organizations generate none of their revenue from software but still use software internally for strategic reasons.

Java Magazine: So what is the business community model that makes OpenJDK, or any open source projects for that matter, work?

Smith: The fundamentals of business suggest that there are three ways to increase the value of your company. You can increase revenue; you can lower costs; or you can increase the multiple that somebody who would want to buy your company would assign you based on your agility, your ability to grow, or whatever. Open source and participating in open source can help in all three ways.

People sometimes say that open source lowers costs because you share the development cost with other organizations. But it doesn't really work that way. Instead, different organizations focus on what they're good at. It's not so much that you lower your costs. But you end up with a richer product with greater functionality than you could do alone. You get more for your dollar because new opportunities arise. In that sense, it lowers costs because open source can often cost less to create higher-quality products. You can drive products in new directions, which helps with revenue.

Java Magazine: Describe the virtuous cycle that drives open source communities.

Smith: As with any cycle, you need a starting point. In most cases, I've noticed that you initially start out with a code base, which is the project. And that project gets adopted by end users. In the case of OpenJDK, there are two kinds of users: there are Java developers, but also the millions of end users who have Java on their desktops and servers.

SPECIALISTS NEEDED

Many contributors may assume that they don't have enough expertise across the Java platform. What is really needed is deep specialization, and there are many technologies to specialize in.

With a large pool of users, whether it's a large pool of developers or end users with Java on their desktops and servers, there exists a business or personal opportunity that attracts vendors and individuals. The vendors will usually be looking for profit in the form of cash. Individuals may be looking for profit, but they may also be simply looking for the satisfaction of having helped and participated in an important software project. Or they may be looking to hone their skills or get the personal satisfaction of fixing a problem.

So those benefits tend to attract more committers who either work for companies or for themselves, who improve the quality and robustness of the software, which in turn strengthens projects, which increases adoption and attracts more vendors, which leads to more committers, which improves the quality, and so on. That's the cycle.

Java Magazine: How can people get more involved in OpenJDK?

Smith: There are many ways to get involved. Many contributors may assume that they don't have enough expertise across the Java platform to be able to contribute, but that's often not the case. What is really needed is deep specialization, and there are many technologies to specialize in. We need help with Java SE 7 updates, new language features, Lambda, Jigsaw, the Mac OS X port, JavaFX, and so on.

If you observe any open source community for a while, you'll know that you can't just show up and expect to know all the norms and development processes and have the credibility to make immediate significant contributions. At Eclipse, we used to call this *meritocracy*, which means that those who demonstrate the ability to effect change will get that right if they want it. And it takes time and results to prove your worth. </article>

Janice J. Heiss is the Java acquisitions editor at Oracle and a technology editor at *Java Magazine*.

LEARN MORE

- OpenJDK
- Donald Smith's blog