

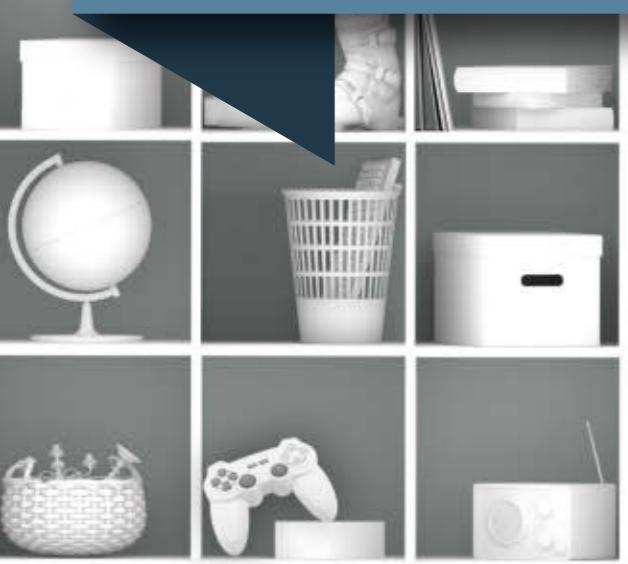
//SEPTEMBER/OCTOBER 2014 /

Java™ magazine

By and for the Java community



WHAT WILL YOU BUILD TODAY?



05

2014 DUKE'S
CHOICE
AWARDS

11

THE MAKER
MOVEMENT
EVOLVES

16

IOT DEVELOPER
CHALLENGE
WINNERS

//table of contents /

05

2014 DUKE'S CHOICE AWARDS

We recognize innovative uses of Java technology.

**COMMUNITY****03****From the Editor****16****Java Nation**

News, people, books, and events

25**JCP Executive Series****Rise of the Machines**

ARM's Zach Shelby on the Internet of Things, the Maker Movement, and the JCP

JAVA TECH**32****New to Java****Fun and Games with Greenfoot and Kinect**

Create a game that is controlled by the movement of the players' bodies.

36**Java Architect****Understanding Java JIT Compilation with JITWatch**

Learn how to examine JIT logs to see what the JIT compiler is doing (and why).

43**Java Architect****Java Apps on iOS and Android—Now a Reality**

Build mobile apps with Oracle Mobile Application Framework, a Java-based hybrid mobile platform.

48**Enterprise Java****Get Your Java Applications Enterprise-Ready with Oracle WebLogic Server**

Make your applications more robust, secure, highly available, and scalable.

55**Rich Client****Location-Based Software Using JavaFX**

Discover the benefits of using JavaFX for visualization software.

60**Embedded****Getting Started with Home Automation**

Leverage openHAB and the huge Java ecosystem to create a truly smart home.

66**Fix This**

Take our streams code challenge!

EDITORIAL**Editor in Chief**

Caroline Kvitka

Community Editor

Yolande Poirier

Java in Action Editor

Michelle Kovac

Technology Editor

Tori Wieldt

Contributing Writer

Kevin Farnham

Contributing EditorsClaire Breen, Blair Campbell,
Kay Keppler, Karen Perkins**DESIGN****Senior Creative Director**

Francisco G Delgadillo

Senior Design Director

Suemi Lam

Design Director

Richard Merchán

Contributing DesignersJaime Ferrand, Diane Murray,
Arianna Pucherelli**Production Designers**

Sheila Brennan, Kathy Cygnarowicz

ARTICLE SUBMISSIONIf you are interested in submitting an article, please [e-mail the editors](#).**SUBSCRIPTION INFORMATION**

Subscriptions are complimentary for qualified individuals who complete the subscription form.

MAGAZINE CUSTOMER SERVICEjava@halldata.com Phone +1.847.763.9635**PRIVACY**Oracle Publishing allows sharing of its mailing list with selected third parties. If you prefer that your mailing address or e-mail address not be included in this program, contact [Customer Service](#).

Copyright © 2014, Oracle and/or its affiliates. All Rights Reserved. No part of this publication may be reprinted or otherwise reproduced without permission from the editors. JAVA MAGAZINE IS PROVIDED ON AN "AS IS" BASIS. ORACLE EXPRESSLY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED. IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING FROM YOUR USE OF OR RELIANCE ON ANY INFORMATION PROVIDED HEREIN. The information is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle. Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Java Magazine is published bimonthly with a free subscription price by Oracle, 500 Oracle Parkway, MS OPL-3C, Redwood City, CA 94065-1600.

Digital Publishing by GTxcel

PUBLISHING**Publisher**

Jennifer Hamilton +1.650.506.3794

Associate Publisher and Audience**Development Director**

Karin Kinnear +1.650.506.1985

ADVERTISING SALES**President, Sprocket Media**

Kyle Walkenhorst +1.323.340.8585

Western and Central US, LAD, and Canada, Sprocket Media

Tom Cometa +1.510.339.2403

Eastern US and EMEA/APAC, Sprocket Media

Mark Makinney +1.805.709.4745

Advertising Sales Assistant

Cindy Elhaj +1.626.396.9400 x 201

Mailing-List Rentals

Contact your sales representative.

RESOURCES**Oracle Products**

+1.800.367.8674 (US/Canada)

Oracle Services

+1.888.283.0591 (US)

Oracle Press Booksoraclepressbooks.com

COMMUNITY

JAVA IN ACTION

JAVA TECH

ABOUT US



02



SEPTEMBER 28 – OCTOBER 2, 2014
SAN FRANCISCO
oracle.com/javaone

REGISTER NOW

Respond Early, Save \$400*

Discover All Things Java

- Learn all about Java 8
- 500+ educational sessions
- 100+ product demos
- Meet with 500+ Java experts



Diamond Sponsor



Silver Sponsor



Bronze Sponsors



IN-MEMORY COMPUTING



JFrog®



New Relic®



Restlet

*Save \$400 over onsite registration. Visit oracle.com/javaone/register for early bird registration dates and details.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

ORACLE®

B

ecause we get to build stuff." That was my nine-year-old son's response to my question about why aviation camp at our local airport was his favorite camp this summer. While kids seem to always be building stuff—from sand castles to LEGOs—as adults we have to try harder to keep that desire to build alive. Why build when you can buy? Luckily, the reasons to build, create, and do it yourself are endless. In this issue, we celebrate the spirit of making and the people who are creating new and innovative technologies and services.

In our [interview with Dale Dougherty](#), *Make:* magazine founder and Maker Faire cocreator, we explore what's behind the growing Maker Movement and how it relates to software development. "We're on the cusp of a new creative industry that is taking product design and the web and beginning to link them," he says. "One of the great starting points for developers is a standardized board such as the Arduino."

We also recognize the winners of the [2014 Duke's Choice Awards](#), which honor compelling and innovative uses of Java technology. These winning makers help ensure safer travel, feed hungry refugees, improve crop yields, provide cool technology, and aim to end a decade-long debate in the Java community.

We're also excited to recognize the [winners of the first-ever IoT Developer Challenge](#), sponsored by Oracle Technology Network and Oracle Academy. Winners used embedded Java with computer boards, such as the Raspberry Pi, and other devices and IoT technologies to showcase innovations in domotics, robotics, office design, authentication, and rating systems.

Congratulations to all of the winners, who show us that the spirit of making is alive and well. Inspired? Get out there and build something.

Caroline Kvitka, Editor in Chief [BIO](#)



PHOTOGRAPH BY BOB ADLER



Find the Most Qualified Java Professionals for your Company's Future

Introducing the *Java Magazine* Career Opportunities section – the ultimate technology recruitment resource.

Place your advertisement and gain immediate access to our audience of top IT professionals worldwide including: corporate and independent developers, IT managers, architects and product managers.

//send us your feedback /

We'll review all suggestions for future improvements. Depending on volume, some messages might not get a direct reply.



For more information or to place your recruitment ad or listing contact:
tom.cometa@oracle.com

The answer is right in front of you

Java Image Enabling SDKs that Help You See the Big Picture

At first glance it may seem difficult, but it's really quite simple. Atalasoft's JoltImage product is a proven SDK for image enabling your Java-based web applications, easily. Image enabling helps to add dimension to your data, so you can uncover insights such as correlations and causations hidden inside your 2-dimensional documents. Our SDK does the heavy lifting for you, saving time, money, and the headaches of figuring it out yourself. Backed by our highly knowledgeable & caffeinated support engineers, JoltImage will enable your success and make the big picture so much easier to see.

Click for tips on viewing the stereogram



2014 DUKE'S CHOICE AWARDS

This year's winners help make travel safer, feed hungry refugees, improve crop yields, provide cool technology, and aim to end a decade-long debate in the Java community.

BY PHILIP J. GILL

This year's Duke's Choice Award winners are an eclectic bunch of developers, innovators, and educators. Some are using the Java platform to address real-world challenges to improve road and air safety, assist refugees, and increase grain crop quality and quantity to help feed the hungry. Others are promoting Java in home automation and game development, solving important technical issues, educating Africa's Java community, and hoping to resolve a longstanding community debate.



THIS YEAR'S WINNERS (in alphabetical order by project name)

Air Traffic Controller Selection System, TUBITAK BILGEM Informatics and Information Security Research Center

Apache DeltaSpike
Apache Software
Foundation

Apache TomEE,
Apache Software
Foundation

Autonomous Vehicle

Test System, Perrone Robotics for the Insurance Institute for Highway Safety

DukeScript, Eppleton IT Consulting and the NetBeans project

IBFieldbook,
International
Maize and Wheat
Improvement Center

JavaFXPorts, LodgON

JCertif International

libGDX

PiDome

Project JEDI, Java Education & Development Initiative

WFP Subsidy Card

e-finance for the
United Nations
High Commissioner
for Refugees and
the World Food
Programme

ART BY I-HUA CHEN



2014 DUKE'S
CHOICE AWARDS

SAFETY FIRST

The driverless car is still a few years away, but crash avoidance systems are already available in some of today's popular luxury automobiles. Working with the [Insurance Institute for Highway Safety](#) (IIHS), Perrone Robotics has developed the Java-based **Autonomous Vehicle Test System** (AVTS) to independently test these safety systems.

JAVA WINS OUT
"Java took the highest score against other platforms."

*—Omer Faruk Arar,
Chief Researcher,
Air Traffic Controller Selection System*

"Many of today's vehicles already have 'crash-imminent braking' and other features," says Paul Perrone, CEO of his eponymous Charlottesville, Virginia-based firm. "But up until now, the IIHS hasn't really had a way to test how well these technologies work in real-world situations with cars traveling at highway speeds."

Perrone Robotics began work on the AVTS in early 2013 and has completed phase one of its contract—which required delivery of one complete system, including one target robot vehicle and one drop-in actuator kit, for use by the IIHS. "They are now ramping up on using the system," says Perrone.

The TUBITAK BILGEM Informatics

and Information Security Research Center, a Turkish government agency, and Air Navigation Service Provider and Airport Operator of Turkey (DHMI) used Java to develop a new computer-based testing system to help select and vet air traffic controllers: the **Air Traffic Controller Selection System** (atcSES). Before atcSES, another computer-based selection system was used, explains atcSES Chief Researcher Omer Faruk Arar. "This system obviously had some drawbacks," says Arar. "It could not provide enough administrative flexibility. With atcSES, necessary flexibility is provided to authorized users and the skills needed for an air traffic controller could be evaluated more interactively by new tests."

With atcSES, the agency evaluates potential candidates on eight essential skills: spatial awareness, reasoning, complex attention, psychomotor abilities, visual memory, auditory memory, cross-control, and basic mathematics.

Before starting the atcSES project, the agency evaluated suitable technologies for the project. "Java took the highest score against other platforms," says Arar. "We needed to implement tests quickly and adapt the system easily. By using Java, we easily achieved that."



PHOTOGRAPHS BY PAT JARRETT/GETTY IMAGES

Left: Paul Perrone at the Insurance Institute for Highway Safety. **Right:** Perrone and team install and test their Autonomous Vehicle Test System.



Tomas “Tito” Sanchez (right) of the International Maize and Wheat Improvement Center analyzes crops with IBFieldbook on a tablet computer.

Breeding Field Book, or **IBFieldbook**. This application helps design field trials and generate crop analysis and enhancement solutions. In conjunction with tablet computers in the field, it efficiently and accurately captures crop phenotype data and saves it in a local crop database before submitting that information to CIMMYT for analysis.

“Before IBFieldbook, people recorded data using Excel spreadsheets and Excel macros before storing the information in Access databases,” explains development team member Tomas “Tito” Sanchez. “We also used some tools developed in Delphi and VB to query the database, consuming a lot of time and generating serious bottlenecks to access the information for analyses. The problem was that, due to different Excel versions, the application was not able to run on different operating systems. That’s why we decided to use Java and NetBeans—because we will be able to deploy IBFieldbook on Linux, Windows, and the Mac OS.”



2014 DUKE'S
CHOICE AWARDS

FIGHTING HUNGER

Two of this year's winning projects are helping fight hunger. The first is the **WFP Subsidy Card**, which is based on Java smartcard technology and was developed by [e-finance](#) for the United Nations High Commissioner for Refugees (UNHCR) and the World Food Programme (WFP).

“The WFP Subsidy Card helps refugees listed by the United Nations get the food they need through secure-card wallets controlled by both organizations,” explains Mohamed Taman, systems architect and project manager for Cairo, Egypt-based e-finance. “Right

now they’re being used by 100,000 refugees; by the end of the year they will be helping 500,000.”

This is the second Duke’s Choice Award for a UNHCR project; the U.N. global refugee agency won in 2012 for Level One, a light client application that enables U.N. personnel to register refugees and assess their immediate needs.

In Texcoco, Mexico, the International Maize and Wheat Improvement Center—el Centro Internacional de Mejoramiento de Maíz y Trigo (CIMMYT)—has used Java, JavaFX, and NetBeans to create a crop and field analysis tool to improve crop quality and yields around the world: Integrated

PHOTOGRAPH BY RAFAEL MONROY/
GETTY IMAGES



2014 DUKE'S CHOICE AWARDS

STRONGER AS ONE

"We are trying to take all those great home automation projects... and put them together in one single package."

—John Sirach, Project Leader, PiDome

PHOTOGRAPH BY TON HENDRIKS

COOL TECH

No Duke's Choice Award winners roster would be complete without some "cool" technology. This year the honors go to the **PiDome** home automation and Internet of Things (IoT) platform, which marries Java and the Raspberry Pi credit-card-sized single-board computer. Some important features are real-time USB device recognition; Raspberry Pi hardware interfaces; a visual trigger editor for any controllable device; and a visual floor planner, which is visualized in 3-D in the PiDome JavaFX OS-themed client with real-time updates.

"What we are trying to do is to take all those great home automation projects by electronic hobbyists and fellow Java developers and put them together in one single package. By supporting existing products, open source projects via plugins, and our own hardware created by Marcel Wiebenga, it will be able to compete with the larger home automation systems," says John Sirach, project leader at PiDome.

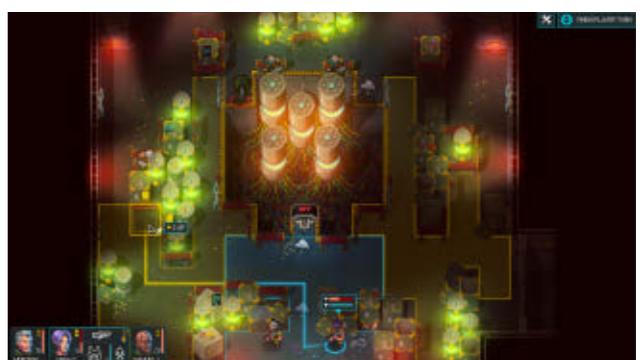
From a diverse team of open source enthusiasts comes **libGDX**, a cross-platform game development framework that allows programmers to



write, test, and debug Java games on a desktop PC running Windows, Linux, or Mac OS X and deploy that same code to Android, iOS, and WebGL-enabled browsers—something not widely available right now. The goal of libGDX, says creator Mario Zechner, "is to fulfill the 'write once, run anywhere' promise of the Java platform specifically for game development."

DukeScript is an alternative to Swing, SWT, and JavaFX, with the benefit that it also runs on Android and iOS and enables true cross-platform Java.

"Although the name somehow seems to imply otherwise, DukeScript is not a scripting language," says DukeScript collaborator Anton (Toni) Epple of Munich, Germany-based Eppleton IT Consulting. "DukeScript applications are plain Java applications that internally



Top: Mario Zechner of libGDX. Center and bottom: screenshots of Halfway, a game developed in libGDX.



JCertif Conference attendees in Africa



use HTML5 technologies and JavaScript for rendering. This way developers only need to write clean Java code and can still leverage the latest developments in modern UI technology." DukeScript is developed as a subproject of NetBeans.

Helping educate future technologists about Java and open source technologies in Francophone Africa is **JCertif International**. Since its founding in 2010, this organization has trained

Three-Way Tie Makes 12 Winners

How fitting that in its 12th year the Duke's Choice Awards has 12 winners, thanks to the first-ever three-way tie in the voting for the coveted Community Choice Award.

The first of the trio is **Project JEDI**, for the Java Education & Development initiative. This community-driven organization provides free courseware and training on Java and open source technologies to colleges and universities around the world. Originally started in the Philippines, Project JEDI now partners with Java user groups (JUGs), academic institutions, government agencies, and businesses around the world to translate and provide the course materials in other languages.

The second winner is the Apache Software Foundation's **Apache DeltaSpike** developer toolbox for Java Contexts and Dependency Injection (CDI) for the Java EE 6 platform. Apache DeltaSpike allows developers to introduce and test new features in Java EE before they are standardized, such as transaction support for CDI beans, which wasn't available until Java EE 7.

The third winner, **JavaFXPorts**, from LodgON, is also an open source toolkit. For the first time, it makes graphically rich Java and JavaFX available to mobile and tablet devices running iOS and Android operating systems.

5,000 Java developers across Africa, according to Founder and Managing Director Max Bonbhel. "JCertif International is an independent, non-profit organization that brings together African community leaders dedicated to growing and empowering developer communities through training and events," explains Bonbhel. "JCertif also provides free training materials and speakers and promotes collaboration

between tech user groups across the African continent."

Its JCertif University, held several times a year in various cities across the continent, offers two- to five-day intensive classes and workshops, typically for 20 to 50 students at a time. Its annual Certif Conference is a one- to two-day conference with lectures and speakers that typically attracts up to 2,000 attendees, says Bonbhel.



SETTLING A DEBATE

The Apache TomEE platform from the Apache Software Foundation effectively ends a debate that has raged for a decade between technologists preferring Java EE-compliant web application servers and those devoted to Apache Tomcat, says David Blevins, the foundation's vice president for Apache TomEE.

The TomEE server merges a certified Java EE 6 Web Profile stack with Tomcat, the leading Java web application server by market share. TomEE is available in two versions. TomEE sticks strictly to the minimum definition of the Java EE Web Profile, says Blevins, while TomEE Plus is a less strict approach and offers some of the key technologies that didn't make it into the Web Profile, such as JAX-WS, Java Message Service, and Java EE connectors.

"The primary question that motivates us is, why should people have to choose between Tomcat and a Java EE-compliant application server?" says Blevins. "After 10 years it still comes up, and people argue against each other as to which is better and why. The debate has raged on for far too long and could be best described as a stalemate. TomEE is one half of an industry movement to answer that question and put that debate to bed permanently." **</article>**

Philip J. Gill is a San Diego, California-based writer and editor.

JUDGES AND PROCESS

The winners of the 12th annual Duke's Choice Awards were selected in a three-part process. First, all members of the Java community were invited to submit nominations to this year's judges, who then selected nine winners from more than 50 submissions and nominated six candidates for this year's Community Choice Award. In the final round, those Community Choice Award nominees were posted on Java.net and all members of the Java community were invited to vote for their favorite.

This year's judges were

Stephen Chin, Java evangelist, Oracle

Michelle Kovac, Java marketing and operations, Oracle

Yolande Poirier, Oracle Technology Network

Reza Rahman, Java evangelist at Oracle and GlassFish community member

Yara Senger, cofounder and president, SouJava

Bruno Souza, cofounder, SouJava

Martijn Verburg, co-organizer, London Java Community

John Yeary, president, Greenville Java Users Group

The 12 winners will be honored at JavaOne September 28–October 2, 2014, in San Francisco, California.



David Blevins, Apache Software Foundation vice president for Apache TomEE, presents at DevOxx.



THE MAKER MOVEMENT EVOLVES

A conversation with Maker Faire cocreator **Dale Dougherty**

BY TIMOTHY BENEKE

The Maker Movement, a fast-growing do-it-yourself (DIY) culture focused on such innovative pursuits as robotics, electronics, and 3-D printing, along with traditional activities such as metalworking, woodworking, and arts and crafts, has reached critical mass. At the first-ever White House Maker Faire, held on June 17, 2014, President Barack Obama called on "people across the country to join us in sparking creativity and encouraging invention in their communities." He proclaimed June 18, 2014, a "day of making" and stated, "Thanks to the democratization of technology, it is easier than ever for inventors to create just about anything. Across our nation, entrepreneurs, students, and families are getting involved in the Maker Movement."

The convergence of the growing Internet of Things (IoT), big data, machine-to-machine technology, and cloud computing, along with enhanced embedded applications, has made it clear that the hardware innovation fostered

PHOTOGRAPH BY ORACLE



by the Maker Movement is opening up rich possibilities for Java developers. The recent [MakerCon Conference](#), hosted at Oracle headquarters, reinforced that as embedded devices get cheaper, more powerful, and more connected, and as the IoT grows, Java developers will face new challenges and opportunities and will look to Oracle for strong support. To that end, a major effort to unify the Java platform is underway—specifically, Java ME with Java SE—to shrink Java SE into the embedded space for smaller devices.

Dale Dougherty, CEO of Maker Media, founder and publisher of [Make: magazine](#), and cochair of MakerCon, is at the center of the Maker Movement. Make:, which Dougherty launched in 2005, provides a wealth of wide-ranging information about do-it-yourself and do-it-with-others projects, along with step-by-step instructions on how to make things.

Dougherty, the cocreator of Maker Faire, has a rich and creative history, much of it tied to O'Reilly Media. Prior to Make:, he developed Global Network Navigator (GNN), the first commercial website, which was launched in 1993 and sold to America Online in 1995. From 1995 to 1999 he was publisher of Web Review, an online magazine for web designers. As publisher of the O'Reilly Network, which he founded with Tim O'Reilly, he developed the Hacks series of books and was honored by the White House as a "Champion of Change." We talked with him to get an insider's view of the Maker Movement.

PHOTOGRAPH BY BLAINE FISHER/GETTY IMAGES



Java Magazine: What's been happening lately in the Maker Movement?

Dougherty: For starters, we had a White House Maker Faire recently, which tells us that the Maker Movement is moving from the margins to the mainstream. I'm quite happy about the impact it can have in areas such as manufacturing, innovation, and especially education—and even in civic and community areas as well. The Maker Movement is a widely distributed social network that anyone can join. You identify yourself as a maker and you're in.

We are trying to reach people who don't identify as makers and show them

that there are pathways to become makers. We believe that there is a set of values that you acquire by making. And one of the key ones is to see yourself as a producer—a shaper and a builder—and not just a consumer. It really doesn't matter whether you're working with wood, plastic, 3-D printing, or flour and egg. Making can take many different forms. We feel that making is important in helping people lead more productive and meaningful lives.

We also see that making is a source of innovation. Tools for people who have ideas for services and things are becoming increasingly available and

Dale Dougherty gives a presentation at Maker Faire Kansas City on June 28, 2014.



Left: Dougherty at Maker Faire Kansas City. **Right:** Kids learn and play at Maker Faire Bay Area.

easy to use. And the time it takes to make things is collapsing. So you might be scratching your head with a good idea and feel like, well, I don't know how to get this done. Making initially has a DIY nature, but as people get involved they meet others and there is more collaborative production.

I met a man recently who was paralyzed from a diving accident and is confined to a wheelchair that has a vertical pole like a joystick that is used for control. He has a paralyzed hand and can't wrap his hand around the pole to use it effectively—in fact, he was just resting his hand

on the top tip of it. So, using a 3-D program, he designed a mitt interface that enables him to wrap his hand around the stick and navigate his wheelchair. He had it 3-D printed in plastic and then printed in metal. It's empowered him to enhance his mobility.

He represents the core of the Maker Movement. Once the door is opened he starts thinking, "Well, there must be other people like me who need this. So what I designed for myself could help other people."

Java Magazine: The Maker Movement seems to affirm certain core human needs to create,

Java, Meet Maker Movement

We asked Noel Portugal, principal user experience developer at Oracle, to weigh in on Java's role in the Maker Movement.

"Java has become the de facto language for enterprise applications. A lot of libraries and code examples that are commonly used in enterprise applications can now be used in conjunction with the Maker Movement through embedded Java and the Internet of Things [IoT]," says Portugal. "Data collected by sensors needs to be analyzed by microprocessors and then sent to the cloud. This is easily achieved using a Java Virtual Machine [JVM] and the Java language."

Java developers already have the skills to create the IoT and build devices with the potential to change the way we live and interact with the world, he says. "It's up to you to dream up and create a physical device that will connect to the cloud. There are no limits to the imagination."

Portugal adds that the Oracle Java team has worked closely with the maker community to make sure that embedded Java works out of the box with embedded ARM microprocessors. "Performance is key for real-time applications, and embedded Java applications are up to the task. The maker community is always welcoming and ready to help."

contribute to society, and connect with other people through community—all in one fell stroke.

Dougherty: Yes, and consumer culture doesn't really promote this. Instead, it promotes the idea that you buy something and you're satisfied. The Maker Movement is saying that when you really make something, whether it's a gift for someone else or something for yourself, something that you put a lot of energy and time into, you value that process a lot and you value what you end up with.

Java Magazine: How does a Java developer plug into the Maker Movement?

Dougherty: It's exciting that software developers have begun to get interested in hardware again—there are a lot of great ways to plug in. We're on the cusp of a new creative industry that is taking product design and the web and beginning to link them. My starting point for *Make: magazine* was hacking the physical world so we could change it, customize it, and interact with it. One of the great starting points for developers is a standardized board such as the Arduino that allows you to log sensors in or make connections and write simple code to do things. That's a really good introduction to what's possible.

People are creating other boards and things that have Wi-Fi baked into them. The Raspberry Pi runs a full Linux environment where you have the tools available for your program source development. The key is, how do you connect with the physical world? You may be

able to do anything you want on a computer, but it's harder to connect an individual sensor to your computer if it doesn't come built into it. So that's the general direction.

A lot of developers have grown up programming just the web, and the idea that you can interact with the physical world is fascinating. Even something like a drone—how do I program a drone to make it fly? Or control a robot? I think what fascinates a lot of people, and especially young kids, is the ability to gain control over the physical world. So you have a robot that you can make go faster or slower or left or right just by writing code. Mastering sensor functionality is one key area for developers.

And what happens when lots of different things in your "built" environment are smart and interacting with you and each other? Take an area such as wearables, which might be a watch or clothing. What behaviors do we want associated with them? Someone recently told me that they were interested in having a sweater that was programmed so that it knew that people were looking at it and reacted. It's a playful idea that gets us leaping into the world of the possible. It gets



us thinking about interface designs, which we tend to think of in terms of the rectangle of the computer screen, but it can include clothing and gadgets. We can ask what we want our sneakers to tell us. Developers have the capacity to enable us to control and integrate lots of devices and to change our environments in ways that make the world more fun and easier to live in.

An architect at a Maker Faire was showing a [Tesla coil](#) that arced through the air in sync with music that was playing. We could have whole buildings doing that. On the other hand, there is a side to this that is very practical.

Dougherty talks with a young maker at Maker Faire Kansas City.



A Maker Faire Kansas City maker shares her story with Dougherty.

A friend of mine told me about a “cat detector” that he built. He and his family had adopted a stray cat that had a lot of scratches and wasn’t doing so well, so they started putting out food for it. They rarely saw the cat, but they saw the empty food bowl and wondered who was eating the food. So he took a little camera and connected it to a motion sensor, along with a trigger light set up by the cat food, so that when something approached the bowl, it took a picture and sent it to his cell phone. They discovered that both the cat and a raccoon were eating the food. It’s a very practical example of solving a specific problem, such as the one for the man in the wheelchair. People now can see

a problem and whip something together off the shelf and solve it. I don’t know if there’s a big market for a “cat detector,” but you would be surprised at the number of similar stories from people who want to know what’s happening in their yard or their chicken coop at night.

Java developers, once they get acquainted with electronics, are in a position to do things on a much grander scale. There’s something a bit disorienting about the Maker Movement, but it’s a kind of creative disorientation because you become aware of so many possibilities. You go to a Maker Faire and see 10 things in a row that are extremely different from each other, each of which leads you to think of new options. You see someone doing some new kind of crochet and someone else with a robot, and soon you start seeing robots doing crochet. And if you are good at coding, you have a great advantage with things such as the Arduino platform, BeagleBone, the Raspberry Pi, and new areas such as the Spark Core board.

We published a really great book called *Make: Electronics* that can help developers make the journey into electronics. One interesting direction is JSON [JavaScript Object Notation] interfaces, which should enable web developers to talk to sensors in a language they are familiar with.

Java Magazine: Any final remarks?

Dougherty: We hear a lot about the IoT. It’s important to keep in mind the new social and human space that gets created when there is a lot of data being passed between devices. We are talking about new kinds of human experiences and behaviors for everyone. The Maker Movement is raising a flag and saying that everyone has something to contribute in creating a new-and-improved world.

What I’m most thrilled about is the number of kids showing up in the Maker Movement. They are so excited and engaged. I really want to help create a world where kids can find places in their community where they learn to make things and become makers themselves. The key to this is the creation of maker spaces in schools and libraries, museums and community centers, formal and informal settings, so that kids can have access to tools and materials and find mentors. This will not only create new innovators but it will help people gain better control of their lives. </article>

Timothy Beneke is a freelance writer and editor who has written for *Mother Jones*, the *East Bay Express*, and the *Chicago Reader*.

LEARN MORE

- [MAKE: magazine](#)
- [White House Maker Faire](#)
- [“A Perfect Match: Java and the Internet of Things”](#)



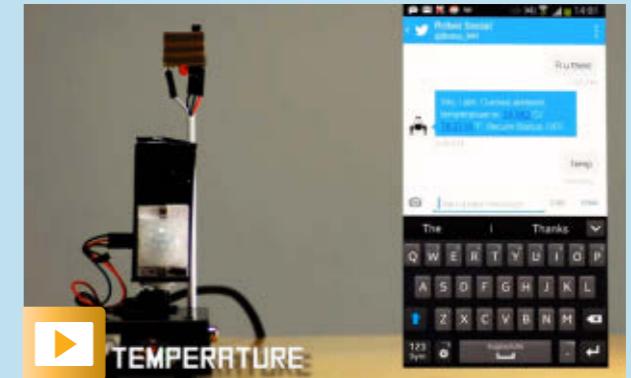
//java nation /



[Smart Greenhouse Project details](#)



[The Lhings Connected Table at work](#)



[The Bot-So project in action](#)



[The ePot Smart Gardening project](#)

IoT DEVELOPER CHALLENGE

Developers from around the world, inspired by the potential of the Internet of Things (IoT), participated in the first-ever IoT Developer Challenge, sponsored by Oracle Technology Network and Oracle Academy. Developers submitted applications that used embedded Java with computer boards such as the Raspberry Pi, and other devices and IoT technologies. The six winning projects showcased innovations in domotics, robotics, office design, authentication, and rating systems.

Winners were broken into professional and student categories. In the professional category,

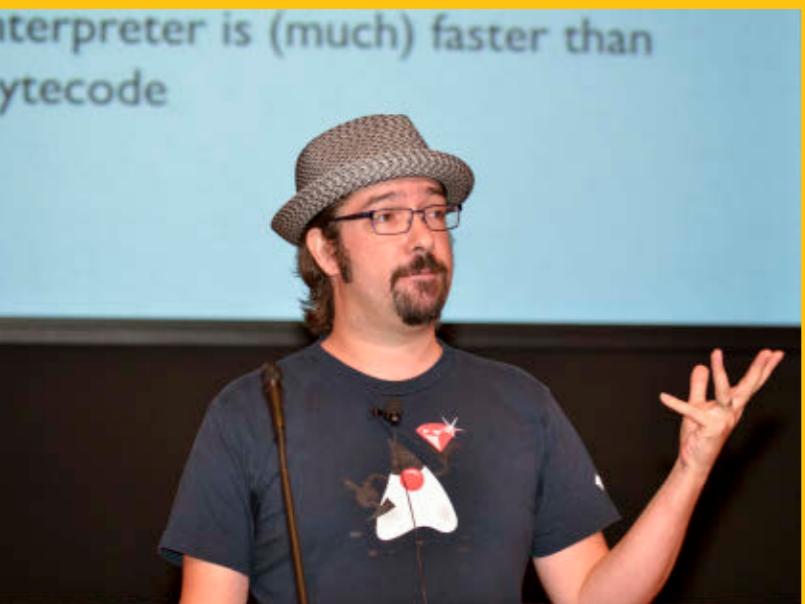
the winning projects were Smart Greenhouse Project, a full-featured, automated greenhouse to grow indoor plants; Bot-So, a smart social robot that interacts via Twitter; and Lhings Connected Table, an innovative design for shared office space. The student winning projects were ePot Smart Gardening, a new concept of indoor gardening; Voice Access, a voice recognition system for authentication; and JCon Rates, a conference voting system.

The winning professional teams and the top student team, ePot Smart Gardening, won trips to JavaOne 2014.

[Learn more about the projects.](#)



Interpreter is (much) faster than bytecode



JVM LANGUAGE SUMMIT



John Rose, JVM Architect at Oracle

Oracle JVM Architect John Rose discusses summit topics.



Nigel Daniels, Product Manager at Waratek

Waratek's Nigel Daniels talks about security solutions for the JVM.

Summit highlights (from top to bottom): Participants mingle during a break; summit presenter Charles Nutter; Brian Goetz (center, in yellow) chats with attendees.

The seventh annual Java Virtual Machine (JVM) Language Summit took place July 28–30, 2014, at Oracle’s Santa Clara, California, campus. This event was an open technical collaboration of 90 attendees, including language designers, compiler writers, tool builders, runtime engineers, and VM architects. The format, a mix of technical presentations and workshops, facilitated discussion groups and enabled deeper dives into the subject matter.

"I'm enjoying the company of a lot of brilliant colleagues," said **John Rose**, JVM architect at Oracle. "For many of us, the JVM Language Summit is the most enjoyable day of the year, professionally, because we get to exchange ideas with our peers from Oracle and from other parts of the Java ecosystem."

Get slides and more from the summit here.

PHOTOGRAPHS BY YOLANDE POIRIER

FEATURED JAVA USER GROUP

ROME JAVA USER GROUP



When in Rome . . . attend a Java User Group (JUG) Roma meeting. Rome JUG was founded in December 2005 by **Mara Marzocchi** and **Ugo Landini**; the current leadership includes **Vitalij Zadneprovskij**.

Rome JUG initiated the annual Java Day in Rome, and its success has been expanded by Marzocchi, a cofounder of the startup Codemotion, which organizes conferences and workshops in Germany, Israel, Italy, and Spain.

The group has more than 950 members on its mailing list, says Zadneprovskij. The group meets every month or two, with an average of 15 attendees. Meetings are held

at startup accelerators and small firms, which host their meetings.

Meeting topics include machine learning; geographic information systems; search engines; and common programming problems and tasks, such as testing, web browser compatibility, and continuous integration.

While not everyone can make it to Rome for the meetings, Rome JUG has a considerable following among the Italian Java developer community. "Many people around Italy receive our e-mails and participate in our discussions about programming, NoSQL databases, project management, employability, and books about software development," Zadneprovskij says.

Zadneprovskij understands that Java is extraordinarily powerful, but that power entails some complexity. "Compared to other languages used for web development, it takes more time to learn Java, and there are multiple tools for everything and no consensus about which one is better," he says. "We consider JSR 274: The BeanShell Scripting Language to be very important for people wanting to learn the language."

Learn more about Rome JUG on its [website](#) and [Yahoo Groups](#) page.

OTN Virtual Technology Summit

Oracle Technology Network held its first Virtual Technology Summit in July 2014. The free event included tracks on database, middleware, and systems and—of course—Java. In the Java track, titled "Big Trends and Technologies," attendees learned how Java lets you mine big data; build robust apps with HTML5, JavaScript, and Java EE; and expand into the Internet of Things (IoT). Java sessions and presenters included "Efficient Architectures for Creating MapReduce Pipelines," by **Fabiane Nardon**, Java Champion and big data expert; "JavaScript/HTML5 Rich Clients Using Java EE 7," by **Reza Rahman**, Java evangelist and Java EE expert; and "How to Become an Embedded Developer," by **Angela Caicedo**, Java evangelist and Java embedded expert. The summit content is now available [on demand](#).

INTRODUCING ROBOBURGH



C.H.I.M.P., the CMU Highly Intelligent Mobile Platform, was showcased at the Launch CMU event at Oracle headquarters.

Technology is changing Pittsburgh's image from the steel city of the 1970s to a twenty-first-century hub for innovation. And much of that is due to the efforts of the Center for Innovation and Entrepreneurship at Carnegie Mellon University (CMU). One area that is thriving is robotics—so much so that Pittsburgh may have to change its name to "Roboburgh."

A select group of Roboburgh residents made the trip to another famous hub of innovation, Silicon Valley, to attend the Launch CMU event at Oracle headquarters in Redwood Shores, California, earlier this summer.

As CMU faculty and alumni began to showcase their inventions to possible investors, it quickly became clear that robots are much more than just a box-of-bolts android that can go out to the front lawn and pick up a newspaper. As **Rob Daley**, CEO and cofounder of CMU-incubated startup 4moms, put it, "There is a whole lot of very core fundamental technology that's related to robotics that's really powerful and really important if you divorce yourself from the media

concept of humanoid robotics."

Daley's company was just one of the many organizations that showed up to explain how they use robotics technology in their products. Representatives from 27 startup companies in industries ranging from healthcare to music were represented at the two demo and poster sessions, which bookended the showcase talks.

Manuela Veloso, a professor of computer science at CMU, said the most-powerful robots are the ones that can ask for help. The "CoBot" robots in Veloso's lab use Java in an Android app as a GUI to enable humans to speak with the robots, providing an interactive capability Veloso called "symbiotic autonomy." Java also powers an app that humans can use to summon the robots. "In our work, Java is used to enable robots to interact with people through phone apps or by accessing the web, or to visualize the robots' state," said Veloso.

Whether these robots transform Pittsburgh into Roboburgh is yet to be seen. But one thing's for sure: These aren't your mother's robots.

//java nation /

Left: a session at Open Cloud Day.
Right: Cody Herriges of Puppet Labs presents on open cloud APIs.



OSCON: OPEN CLOUD DAY

Open Cloud Day, held July 21, 2014, during the OSCON open source convention, focused on the latest innovations in public and private clouds, infrastructure as a service (IaaS), and platform as a service (PaaS). In his session, "The Enterprise Challenge for Cloud Computing," **David Nalley**, committer to Apache CloudStack, tackled the question, "Why hasn't cloud computing dominated the enterprise already?"

Nalley said that the cloud can be great for certain things, such as application test beds, but thinking that the cloud will solve all your problems is a little like believing in unicorns. He asked the audience to consider value, pay for use, fiefdoms, and fragility when thinking about moving enterprise

applications to the cloud.

The cloud is also forcing enterprises to rethink their IT services. "People are really more empowered at home than at work," explained **Chris Launey**, director of cloud services at Disney, in his session. Launey doesn't want Disney developers to come up with an idea over the weekend, begin developing it using free services they can access at home, and then run into a brick wall when they try to implement it at work on Monday. Things need to be much, much faster. "I challenge my team to do things 'cloudly,'" he said. "That means we need to make requests stupidly easy, provide fast delivery, allow for painless management, and provide data transparency and granular billing."

JCP Celebrates 15 Years



The Java Community Process (JCP) celebrated its 15th anniversary with a gathering at the Computer History Museum in Mountain View,

California, on June 18, 2014. The event, cohosted by the JCP program office and the Silicon Valley Java User Group, was a chance for members of the Java community and some of the people instrumental in the JCP's success to network and check out the cool museum exhibits.

"For 15 years the JCP has been working with corporations, individuals, nonprofits, and Java user groups to advance and improve the Java platform," said JCP Chair **Patrick Curran**. "Every few years we like to get together and say thank you, because Java would be nothing without you."

He recognized many of the luminaries who had played a part in the JCP over the years, including **James Gosling**, **George Paolini**, **Rob Gingell**, **Onno Kluyt**, **Bill Shannon**, and **Vineet Gupta**, and then asked everyone to raise a glass to the JCP. Later, Duke shook hands and posed with attendees.

Watch **Stephen Chin**'s interviews with [Gosling](#), [Gingell](#), [Van Riper](#) and [Kevin Nilson](#), and others.

//java nation /



Clockwise from top: a Devoxx UK expert panel, "Fearless Change" presenters Linda van der Pal (left) and Régina ten Bruggencate, a red-hatted attendee with a message



DEVOXX UK: MIND THE GEEK

If you're a curious developer, [Devoxx UK](#) is an amazing conference.

Held June 12–13, 2014, the conference had more attendees, more sessions, and more sponsors than last year. Java and Java 8 got lots of attention, with several sessions on lambdas and other Java 8 features. **Venkat Subramaniam**'s session, "Programming with Lambda Expressions," was full. He explained that lambdas are good in themselves, but their real value is showing you what's possible with the Stream API. "Lambdas are the gateway drug to streams," he said with a laugh. Experts discussed JavaFX; security; Java EE; and, of course, the Internet of Things (IoT).

In his session, **Dick Wall** showed his IoT Venn diagram and explained that IoT is exploding. He admitted that his dog has a fitness band and is Wi-Fi-enabled. Wall said it's not just the internet anymore; it's the "ubiquinet."

At the closing keynote, London Java Community co-organizer **Martijn Verburg** encouraged attendees to start using Java 8 and to hook it into their continuous integration servers. He also said that every developer is an architect, and if you are an architect who doesn't code regularly, "you should quit right now." If you are a developer who isn't comfortable considering the architecture of applications and working on a whiteboard, he said, "you should consider another profession." Finally, he encouraged developers to become DevOps pros, and to automate everything that they can.

Mark your calendar for June 17–20, 2015, for the next Devoxx UK.

The Next Big Thing?

Do you have an idea for the next killer app or a practical business service? You could win a trip to GeeCON in Kraków, Poland, or present your ideas at the NextGen conference in London, England. Orange Labs is hosting the [Business Intelligence hackathon API \(BihAPI\)](#) and is opening its labs for developers to write cool, innovative applications that extract core network features such as device location, in-application charging, messaging, and Web Real-Time Communications. The Polish cities of Warsaw, Gdańsk, and Poznań will expose government records such as bus and tram timetables, park and ride locations, maps, and much more. Registration opens October 2, and the hackathon runs through November 2014.



JAVA CHAMPION PROFILE FABIANE NARDON



Fabiane Bzinella

Nardon is a computer scientist, innovator, entrepreneur, and impossible projects expert. She won a Duke's Choice Award in 2005 for her development work on Brazil's National Health System and became a Java Champion in February 2006.

Java Magazine: Where did you grow up?

Nardon: I grew up in a small town in the

south of Brazil called Soledade.

Java Magazine: When and how did you first become interested in computers and programming?

Nardon: My father gave me a computer when I was 12, and I found out that I could do a lot with it. I think I spent most of my free time programming on it in the next few years.

Java Magazine: What

was your first computer and programming language?

Nardon: The first computer was a CP200S, a Brazilian clone of the Sinclair ZX81. My first programming language was BASIC.

Java Magazine: What side effects of your career do you enjoy the most?

Nardon: A career in computer science is amazing. There are plenty of jobs (at least where I live), and you have the power to create something new and really change lives.

Java Magazine: Has being a Java Champion changed anything for you with respect to your daily life?

Nardon: I live in São Paulo, a city where there is always something happening, so I try to enjoy the city as much as

I can. Usually I do long cycling rides or Pilates, walk through the city, meet with friends, and sometimes enjoy just staying home with a good book or movie.

Java Magazine: What side effects of your career do you enjoy the most?

Nardon: A career in computer science is amazing. There are plenty of jobs (at least where I live), and you have the power to create something new and really change lives.

Java Magazine: Has being a Java Champion changed anything for you with respect to your daily life?

Nardon: I feel that I'm better connected, and that can be really helpful in your career.

I also learn a lot from my fellow Java Champions, and I feel I'm a better developer because of them.

Java Magazine: What are you looking forward to in the coming years?

Nardon: I think the big data revolution will affect the way we think about software and exciting new applications will emerge. It will also bring new challenges in terms of privacy and how to deal with all these new discoveries in the most ethical way. I'm looking forward to being a part of all this, and I hope to create software to improve our lives.

You can find Fabiane Nardon on [Twitter](#) and on [LinkedIn](#).

//java nation /



EVENTS

DEVOXX 2014 NOVEMBER 10–14 ANTWERP, BELGIUM

Devoxx is the largest Java conference in Europe. This year's theme is "Infinite Possibilities" for Java development. The five-day conference is a gathering of world-renowned speakers and developers from 40 countries. The schedule is packed with keynotes, sessions, hands-on labs, quickies, and Birds-of-a-Feather (BOF) sessions. Tracks include startup, architecture and security, methodology, Java SE, Java Virtual Machine (JVM) languages, mobile, future, server-side Java, cloud and big data, and web and HTML5.

PHOTOGRAPH BY JEFF HAWKINS AT FLICKR

Code.talks 2014

OCTOBER 9–10
HAMBURG, GERMANY

Designed by developers for developers, this conference focuses on trends and topics about web application development. Topics include Java, big data, networking, Ruby, PHP, and more.

Silicon Valley Code Camp

OCTOBER 11–12
LOS ALTOS HILLS, CALIFORNIA

At this free community event, developers learn from fellow developers about software development, software branding, and legal issues around software.

JDD 2014

OCTOBER 13–14
KRAKOW, POLAND

This conference for Java enthusiasts includes two days of information-packed lectures, special workshops, and interactive training. The conference attracts international speakers, who present in English.

JAX London

OCTOBER 13–15
LONDON, ENGLAND

The conference brings Java, Java Virtual Machine (JVM), and enterprise professionals together for a technology- and methodology-focused event. It offers a deep dive for the modern developer and architect aiming to transform open technologies.

GeeCON Prague 2014

OCTOBER 23–24
PRAGUE, CZECH REPUBLIC

GeeCON focuses on Java-based technologies, dynamic languages, rich internet applications, enterprise architectures, patterns, distributed computing, software craftsmanship, and much more.

W-JAX

NOVEMBER 3–7
MUNICH, GERMANY

W-JAX focuses on the Java platform, web, architecture, agile, and the cloud.

//java nation /

EVENTS

JMaghreb

NOVEMBER 4-6

CASABLANCA, MOROCCO

JMaghreb is the largest annual Java developer conference in North Africa. Organized by the Morocco Java User Group, the three-day conference provides high-quality content by top-notch global speakers.

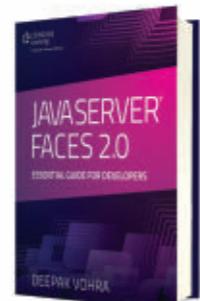
J-Fall

NOVEMBER 5

NIJKERK, THE NETHERLANDS

J-Fall is the annual Java developer conference organized by NL JUG, the Dutch Java user group. About 1,200 developers attend the conference, which features more than 40 technical sessions.

JAVA BOOKS



[JAVASERVER FACES 2.0: ESSENTIAL GUIDE FOR DEVELOPERS](#)

By Deepak Vohra

Cengage Learning, June 2014

In *JavaServer Faces 2.0: Essential Guide for Developers*, Java programmer and web developer Deepak Vohra covers all the new features of JavaServer Faces (JSF) 2.0 using examples in Oracle JDeveloper and the integrated Oracle WebLogic Server. The book covers all aspects of web application development with JSF 2.0, including setting up the environment for an application, using the Oracle JDeveloper wizards and the component palette, and running a sample application. Code examples used in the book are available for download.



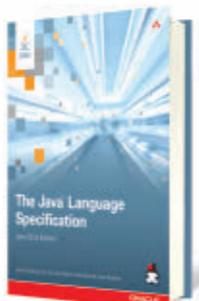
[OCM JAVA EE 6 ENTERPRISE ARCHITECT EXAM GUIDE](#)

By Paul R. Allen and

Joseph J. Bambara

Oracle Press, August 2014

Written by industry experts Paul R. Allen and Joseph J. Bambara and crafted from hundreds of hours of experience teaching Java technologies to students, this Oracle Press certification guide covers 100 percent of the exam objectives and addresses the three exams (1Z0-807, 1Z0-865, and 1Z0-866) and their formats. The book's end-of-chapter multiple-choice questions are accompanied by in-depth answer explanations. A sample case study with essay questions helps you prepare for both the programming assignment and the essay exam.



[THE JAVA LANGUAGE SPECIFICATION, JAVA SE 8 EDITION](#)

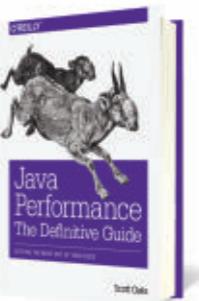
By James Gosling, Bill Joy,

Guy L. Steele, Gilad Bracha,

Alex Buckley

Addison-Wesley Professional,
May 2014

This definitive technical reference provides complete, accurate, and detailed coverage of the Java programming language. It fully describes the new features added in Java SE 8, including lambda expressions, method references, default methods, type annotations, and repeating annotations. The book also includes many explanatory notes and carefully distinguishes the formal rules of the language from the practical behavior of compilers.



[JAVA PERFORMANCE: THE DEFINITIVE GUIDE](#)

By Scott Oaks

O'Reilly Media, April 2014

Coding and testing are often considered separate areas of expertise. In this comprehensive guide, author and Java expert Scott Oaks takes the approach that anyone who works with Java should be equally adept at understanding how code behaves in the Java Virtual Machine (JVM), as well as the tunings likely to help its performance.



JCP Executive Series Rise of the Machines

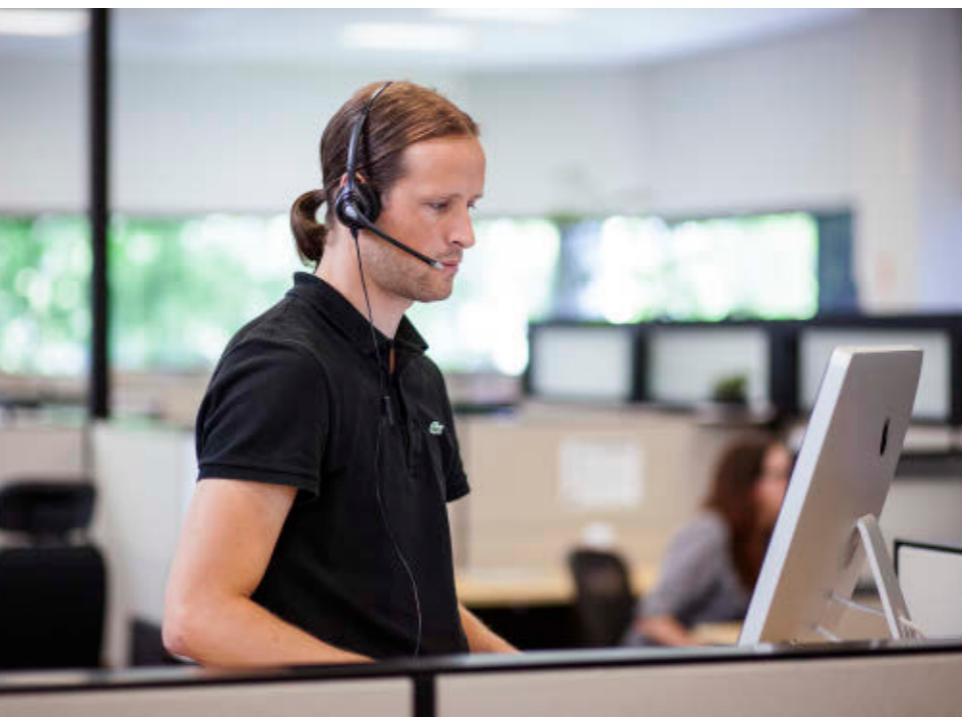
ARM's Zach Shelby discusses the Internet of Things (IoT), evolving standards, the Maker Movement, and how the JCP helps to facilitate these dynamic technologies. **BY STEVE MELOAN**

PHOTOGRAPHY BY BOB ADLER

Left to right: ARM's Zach Shelby, Patul Sharma, and Sergio Scaglia walk to a meeting.

Zach Shelby is director of technical marketing for Internet of Things at ARM and a recognized industry thought leader. Shelby cofounded Sensinode, where he served as CEO, CTO, and chief nerd before the company's recent acquisition by ARM. Prior to Sensinode, he led wireless networking research at the Centre for Wireless Communications and at the Technical Research Center of Finland.

Shelby is a key contributor at the Internet Engineering Task Force (IETF), Open Mobile Alliance (OMA), and ZigBee for IoT standards and has contributed to 6LoWPAN, routing, web services, and security-related standards, and to several international research programs. Shelby is known as a pioneer in the use of IP and web technology in low-power networks with 6LoWPAN and Constrained Application Protocol (CoAP) standards development, and is coauthor of the book 6LoWPAN: The Wireless Embedded Internet. His large portfolio includes courses, publications,



Left: Shelby confers with colleagues. **Right:** Shelby takes a call at his standing desk.

CLOUD ECOSYSTEM

"The IoT domain is using standards and open source projects to build an ecosystem through the cloud, allowing a variety of devices to be enrolled in services on the fly."

public talks, broad research cooperation, and key patents. Shelby has served on the technical advisory board and is currently on the board of directors at the Internet Protocol Smart Object (IPSO) Alliance. He was elected to the Java Community Process (JCP) Executive Committee in 2014.

Java Magazine: Give us a quick overview of the interplay between innovation and standards in the IoT arena.

Shelby: If we look back to the development of internet protocols, it was all about facilitating innovation. That should be the goal. Standardized methodologies were created for moving packets between sites, which led to the creation of HTTP and the web. The ability to create web pages, host them on any web server, and disseminate the data

through any internet service provider, created an explosion of information and widespread access, leading to a level of innovation we have never seen before.

We're about to see similar creativity unleashed in the IoT domain. Many of the standard building blocks have been coming together during the past ten years. Hardware and software technologies are now mature enough to create a wildly fertile environment for IoT innovation.

Java Magazine: What are the distinctions and interactions between M2M and IoT?

Shelby: Machine-to-machine [M2M] technologies have been around for a long time, typically associated with cellular connected devices that were one-off siloed systems. Enterprises that wanted M2M systems generally built

the entire technology kit themselves. They engineered their own protocols and data formats and created the back-end systems more or less from scratch. There was very little reuse from one M2M system to the next.

Going back to the internet analogy, this situation is similar to the state of technology before the web, when providers such as AOL and CompuServe were silos controlling the entire data dissemination experience with proprietary software. M2M has been in that state for quite some time. But the IoT is going to change the landscape significantly with agreements between significant business entities in this space.

The IoT domain is using standards and open source projects to build an ecosystem through the cloud, allowing a variety of devices to be enrolled in services

on the fly. We're really breaking down the silo approach that was associated with M2M. This transition from M2M to IoT is very much like the evolution of web technologies in an earlier era.

Java Magazine: Tell us about some of your experiences as a JCP Executive Committee member.

Shelby: ARM Holdings develops architecture and instruction sets for ARM-based products, but does not manufacture. We've been involved in making sure Java works well on all kinds of devices, from embedded Java ME applications all the way up to 64-bit ARM-based server technology. We put a huge amount of effort into that endeavor, along with Oracle and the rest of the Java community.

My role on the JCP has been to pro-

mote IoT protocols and security technologies. Programmers are comfortable accessing data via HTTP or Transport Layer Security [TLS]. We need to make sure that IoT protocols are no different. They must have the same standardized ubiquitous availability. In addition, I want to promote standard APIs to access IoT protocols. From the JCP perspective, it will be important to standardize the APIs, not just the protocols.

There is a high level of openness in the Java ecosystem. And that's a catalyst for innovation. But the JSR process is designed to ensure, through standards, that the needs of millions of developers are being met. The Java platform has to function as a well-integrated totality, serving a diverse community.

Java Magazine: How does the open ARM mbed development platform promote innovation and higher-quality applications?

Shelby: I've been working in the embedded industry for 20 years, and development has always been associated with expensive, very specialized tool chains, archaic languages, and software libraries that are difficult to use and not integrated. There has been a scarcity of high-quality open source software.

The mbed development platform is designed to defragment the embedded microcontroller software ecosystem. The focus is under-the-hood software for secure boot loading, drivers, crypto

support, networking protocols, Wi-Fi, and so on. The goal is to redefine the developer experience, disengaging from cumbersome tool chains on specialized PCs, toward development in the cloud. This is high-quality open source software, maintained by professionals. We hope mbed will inject a sense of creativity and excitement.

The cloud suddenly became cool when the right tools were available, and people were taught how to access the technology. With mbed, we have a unifying platform for microcontroller development. It also supports Java ME across all ARM Cortex-M-based devices.

Together with our licensing partners, we shipped over 3 billion Cortex-M-based devices last year. The mbed platform will provide a wide range of highly efficient programming tools to support those devices.

Java Magazine: What is 6LoWPAN, and why is it important?

Shelby: 6LoWPAN is a technology for running internet protocols over low-bandwidth networks. It's essentially a simple method of header compression using some of the features of Internet Protocol version 6 [IPv6].

Moore's Law does not apply to radio networks; they don't escalate in speed and efficiency the way processor technologies do. We've achieved some very high data rates, but that comes at the cost of power and range.

Many networks, however, are in the range of tens of Kb/sec, to a few hun-

Shelby unwinds with Sharma over a game of foosball.



MAKING DATA MEANINGFUL

"For the IoT, we need to build semantics and meaning into data received from devices, going out to cloud services. For instance, what does temperature mean?"

dred KB/sec, shared across tens or even thousands of devices. Without these low-bandwidth networks, we don't really have an IoT; we just have mobile cellular devices and Ethernet-connected industrial controllers, which has been the past paradigm.

6LoWPAN technology is really no different from any other IP networking technology. It can be run on almost any network—ZigBee, Sub-GHz, Bluetooth Low Energy, and so on. It's really just a way of compressing IPv6 over all those different technologies.

A new standard called Thread will simplify implementing 6LoWPAN technology into home devices. The focus is home automation and home security. The goal is to make these implementations easily available and certifiable by the product manufacturers.

From the Java perspective, Thread is just another networking interface. It will look like a Wi-Fi network, or any other network that transmits IP packets.

Java Magazine: How important is the Maker Movement in the IoT domain?

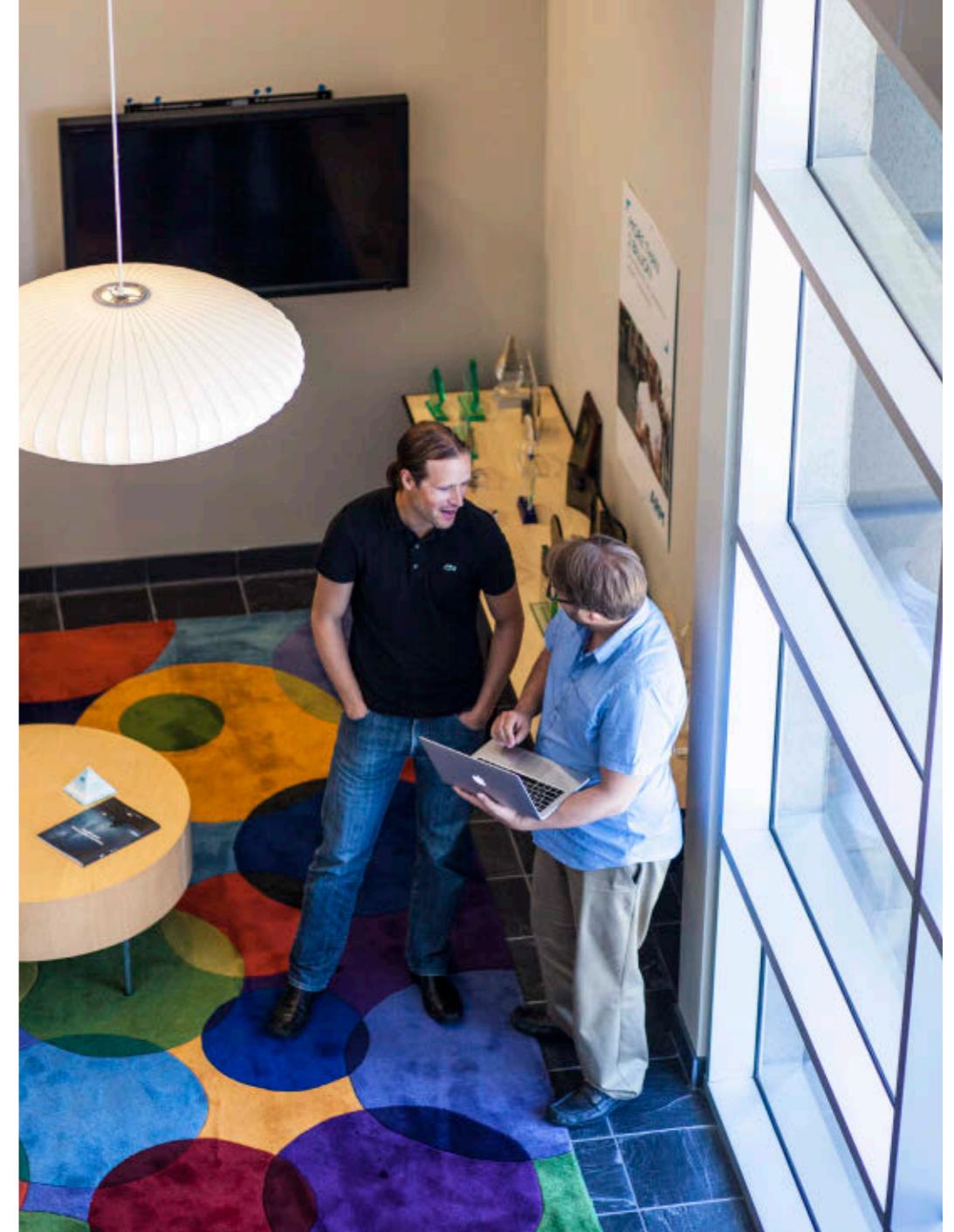
Shelby: The traditional picture of embedded developers has been nerds with horn-rimmed glasses hacking a

test harness and debugging by looking at signals on a wire with an oscilloscope. That's still important work; we need people to create those underlying technologies. But now we need products that are exciting and sexy.

The Maker Movement is creating this kind of energy, facilitating inventors in 3-D printing; robotics; Computer Numerical Control [CNC]; and even woodworking, metal working, and traditional arts and crafts. In response to this movement, developers are producing tools that have broader applicability. The mbed platform is an example of this evolution. The Arduino single-board microcontroller is another example. It has made the experience of configuring an embedded device very simple. We've just launched a new ARM-based Arduino device.

The Maker Movement will motivate more and more technology to service these widely diverse activities. Professional developers, product manufacturers, startups, and hobbyists will all be in the game.

Java Magazine: What are some of the standards evolving in this space? And how does the IPSO Alliance promote innovation?



Shelby: For the IoT, we need to build semantics and meaning into data received from devices, going out to cloud services. For instance, what does temperature mean? And what is the location of this device?

In the world of M2M, everyone baked their own semantics into the protocols. And it was created from scratch every time. A satisfying challenge for engi-

Shelby chats with Dominic Pajak at ARM's San Jose, California, offices.

neers, but not very efficient.

This is contrasted by the web, where a great deal of research has been done on semantic markup ontologies. How do we give a name and an abstract datatype to everything in the known universe? That's the other extreme.

A key issue for the IoT has been the creation of simple semantics that embedded devices, as well as cloud services, can use. And progress has been made in a number of areas. Some significant work has been done by the IETF, which is a large open international community of network designers, researchers, and vendors, dedicated to the evolution of internet architecture. The IETF has done work on application-layer protocol mechanisms. HTTP is standardized there. The CoAP, which allows simple electronic devices to communicate interactively over the internet, has also been standardized by the IETF.

The OMA delivers open specifications for creating interoperable services. They've created a standard called Lightweight M2M, which was released last year. It provides a standard methodology for formatting objects, and defines six normative objects: LWM2M Server, Access Control, Device, Connectivity Monitoring, Firmware, and Location.

The IPSO Alliance is a nonprofit organization promoting IP for smart object communications in consumer, healthcare, energy, and industrial applications. ARM and Oracle are both mem-

bers. Currently the IPSO Alliance is helping to define semantics for objects that have been created by OMA and the IETF. A great example is general lighting-control power-measurement sensors. Those have recently been defined, and are being published for public use with no fees.

Java Magazine: What motivated you to join the JCP?

Shelby: I wanted to join the JCP to ensure that there would be more representation from the IoT and embedded perspectives. Traditionally the JCP has been weighted toward financial or large back-end services—big metal creating web services. There hasn't been nearly as much embedded influence, and we need that.

I joined ARM through the acquisition of my company, Sensinode. Becoming part of the JCP was a natural step, because I wanted to contribute my experience in this area and add to the diversity of the ecosystem.

Right now we have some interesting agenda items. We're working to decide which protocols need standard APIs. Also, open source software for the CoAP protocol is a priority.

Java Magazine: Has Oracle delivered on the promise of increased transparency and openness in the JCP?

Shelby: I'm very new to the JCP endeavor. As a startup, you don't tend to become involved in the political processes behind Java. You just use Java as a tool of innovation.

Now that I'm part of a larger enterprise, we have broader responsibilities to make sure that technologies are standardized in ways that benefit the entire ecosystem. I think the process of defining APIs for the big Java releases works quite well.

In truth, the transparency issue is more related to perceptions than practicalities. There's sometimes a perception that everything is locked down by Oracle. But opportunities for participation by a broad range of interested parties are definitely there.

Java Magazine: Do you have any recommendations regarding how the JCP could better serve the Java community?

Shelby: We could do a better job of teaching the Java community how new technologies, software, and APIs work. What are the optimal ways to deploy these solutions and best scenarios of usage? I think that area is underserved.

Shelby presents to colleagues in a meeting room at ARM's San Jose, California, offices.





Shelby takes a stroll on the ARM campus between meetings.

The JSR process is very technical; it sometimes seems a bit clinical. The political processes involved in developing specs are important, of course. But I think we need to more effectively promote the viability and applicability of Java technologies.

Java Magazine: What is the future for Java in the embedded world? And how will these developments affect us all?

Shelby: It's important for Java developers to remember that native code will never completely disappear. It didn't happen in the PC world, and it won't in the embedded space either. A lot of native software is running under and in Java in order to bring all these technologies together.

The Maker Movement will create new demand for tools and technologies that are standardized and widely available. With nine million developers worldwide, Java is well positioned to provide tools that are easy to update and maintain. But developers will need to know a bit of both. Native elements can be created and configured using technologies such as mbed.

There will never be a one-size-fits-all solution in the embedded world. A mix of various tools and technologies will be required to service this dynamic and quickly evolving space.

Regarding the repercussions of IoT in real-world applications, I'd say the key word is *seamless*. The forward evolution is not just about the IoT; it's also about the cloud. Everything will be interconnected.

In the home, pressure sensors and motion sensors will know when you're ready to watch television, and your smartphone will convert to a remote controller—automatically launched. Intelligence and connectivity will be added to everyday devices to make them function usefully and elegantly, as a well-orchestrated totality.

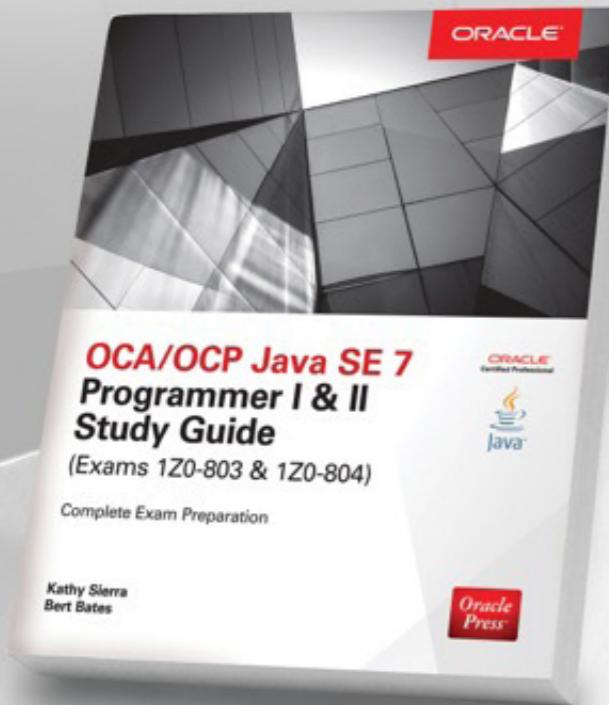
But we're not just talking about the consumer domain. The IoT will be a part of industrial environments and cities, in ways that we can hardly even imagine right now. The possibilities are truly exciting. </article>

Steve Meloan is a former C/UNIX software developer who has covered the web and the internet for such publications as *Wired*, *Rolling Stone*, *Playboy*, *SF Weekly*, and the *San Francisco Examiner*. He recently published a science-adventure novel, *The Shroud*, and regularly contributes to *The Huffington Post*.

LEARN MORE

- [ARM mbed platform](#)
- [Open Mobile Alliance](#)
- [IPSO Alliance](#)

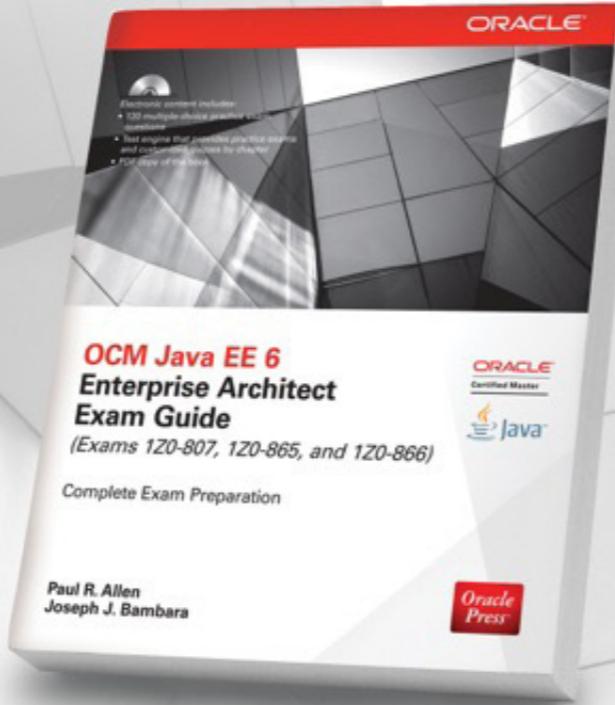
Written by leading Java experts, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Java available.



OCA/OCP Java SE 7 Programmer I & II Study Guide

Kathy Sierra, Bert Bates

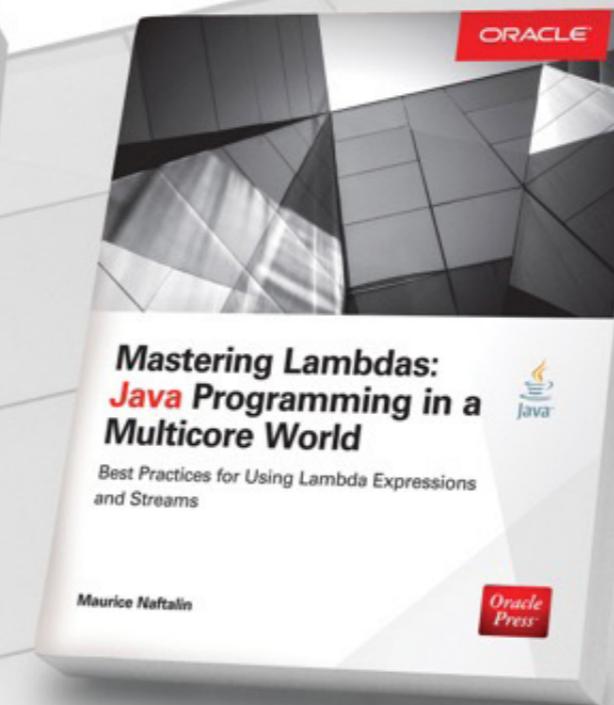
This self-study tool offers full coverage of OCA/OCP Exams 1Z0-803 and 1Z0-804. Electronic content includes 500 practice exam questions.



OCM Java EE 6 Enterprise Architect Exam Guide

Paul R. Allen, Joseph J. Bambara

Get complete details on Oracle Certified Master Java EE 6 Enterprise Architect Exams 1Z0-807, 1Z0-865, and 1Z0-866. 120 multiple-choice practice exam questions are included.



Mastering Lambdas: Java Programming in a Multicore World

Maurice Naftalin

Effectively use Lambda expressions to maximize performance improvements provided by multicore hardware.



Java EE 7: The Big Picture

Danny Coward

Master the code, applications, and frameworks that power Java Platform, Enterprise Edition 7.





MICHAEL KÖLLING

BIO

Part 1

Fun and Games with Greenfoot and Kinect

Create a game that is controlled by the movement of the players' bodies.

The early issues of *Java Magazine*, from the [premier issue](#) in late 2011 to the [May/June 2012 issue](#), contained a series of articles introducing the [Greenfoot](#) environment.

Greenfoot is an educational development environment aimed at novice programmers, from teenagers to adults. It was developed at the University of Kent, UK, and has had long-standing support from Oracle. It provides an engaging, graphical, and interactive introduction to learning to program.

Greenfoot makes it very easy to create interactive, animated graphical applications, such as games and simulations, and millions of young programmers have learned to program with it. And—even better—the language used in Greenfoot is pure Java.

The initial series of articles introduced the environment and discussed how beginners who have no prior programming experience can develop a simple computer game in five easy steps. If you have not used Greenfoot before, you might want to read those articles and try it for yourself.

In this series of articles, we will make an even more exciting game: We will add hardware and start controlling a game by waving our arms. In other words, we will use Greenfoot with the Microsoft Kinect sensor.

The Kinect Sensor

Kinect (see **Figure 1**) is a motion sensor board developed originally for the Microsoft Xbox game console. It uses a high-resolution web cam, an infrared projector, and an infrared camera to

**Figure 1**

track the motion of human bodies in three dimensions.

By connecting Kinect to Greenfoot, we can create games that we can control with our bodies: No hitting of keys and no mouse clicks—just jumping up and down and waving your arms in the air.

Installing the Drivers

Before using Kinect with Greenfoot, you need to install some driver software: OpenNI, to talk to the hardware, and the Greenfoot Kinect server, which lets Greenfoot talk to the Kinect

driver. Follow the detailed installation instructions [here](#).

Let's Get Started

The easiest way to get started is by taking an existing Greenfoot Kinect project, which is called a *scenario*, and making some alterations to it. We will use a scenario called *paint*. [Download it](#) and then do the following to try it out.

Connect your Kinect to a USB port of your computer, start the Greenfoot Kinect server, and then start Greenfoot. Unzip the *paint* scenario you downloaded (this will create a scenario

//new to java /

folder called **paint**) and open it using Greenfoot's **Scenario->Open** menu item. Run the scenario by clicking the **Run** button in the main window.

If all is running well, you will see a small thumbnail of the camera image in the bottom right of your screen. This helps you determine

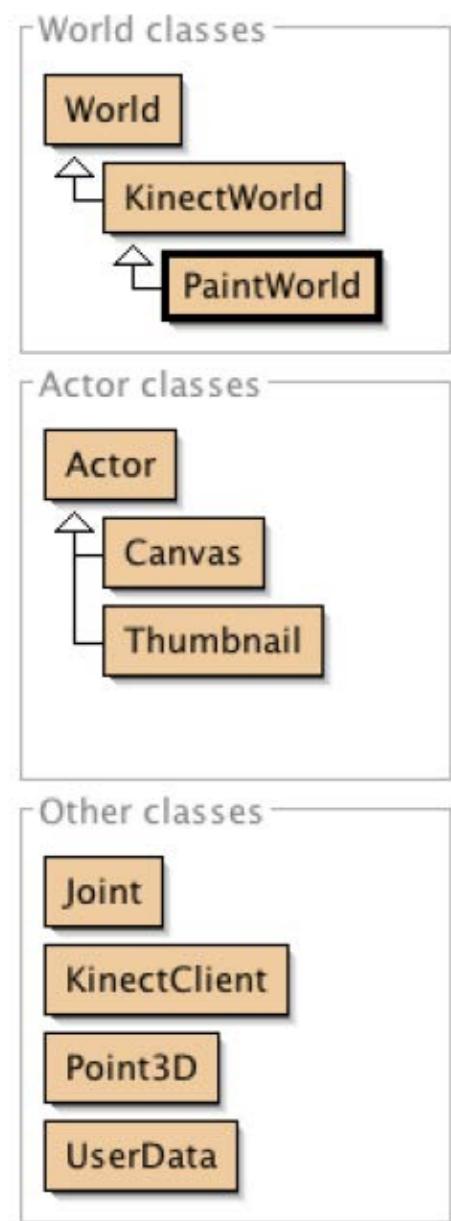


Figure 2

whether you are in Kinect's view and to position yourself. Once you are in Kinect's view, a stick figure that follows your own movements will appear on the screen.

Move around and see what happens. You will notice that you can “paint” on the screen using your right hand. The scenario can track multiple people; try having two or three people on the screen.

The Source Code

The paint scenario includes several classes, shown in the class diagram in **Figure 2**. The **KinectWorld** class and all the classes in the “Other classes” section (**Joint**, **KinectClient**, **Point3D**, and **UserData**) are part of the Greenfoot Kinect framework, and will always be present in Greenfoot scenarios that use Kinect. The **PaintWorld** class and all classes under **Actor** are specific to this scenario, and this is where we can work and make modifications.

In this article, we will do all our work in the [Canvas](#) class.

Tracking users. When we work in Greenfoot with Kinect, we always start at the `world` object, which will always be a subclass of `KinectWorld`. From the `KinectWorld` object, we can obtain an array of `UserData` objects that give us information about each user Kinect can currently see (see

Figure 3). KinectWorld provides a

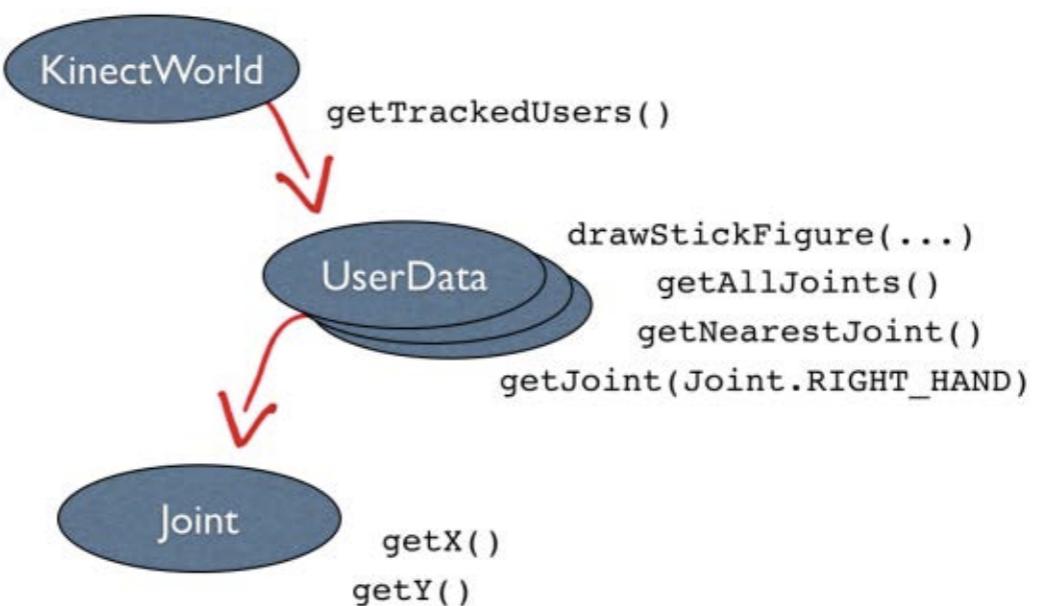


Figure 3

`getTrackedUsers()` method for this purpose.

Once we have a **UserData** object, we can work individually with each user that is being tracked. Some of the methods available for a user are methods to draw a stick figure of the user and to receive information about individual joints, such as those for the hands, knees, feet, and head. Kinect tracks the 15 joints shown in **Figure 4**.

You can see all the available methods by opening the [UserData](#) class in Greenfoot, and then switching the editor from Source Code view to the Documentation view.

When we use a method, such as `getJoint`, to receive information about a joint, we will receive an object of class `Joint`. We can

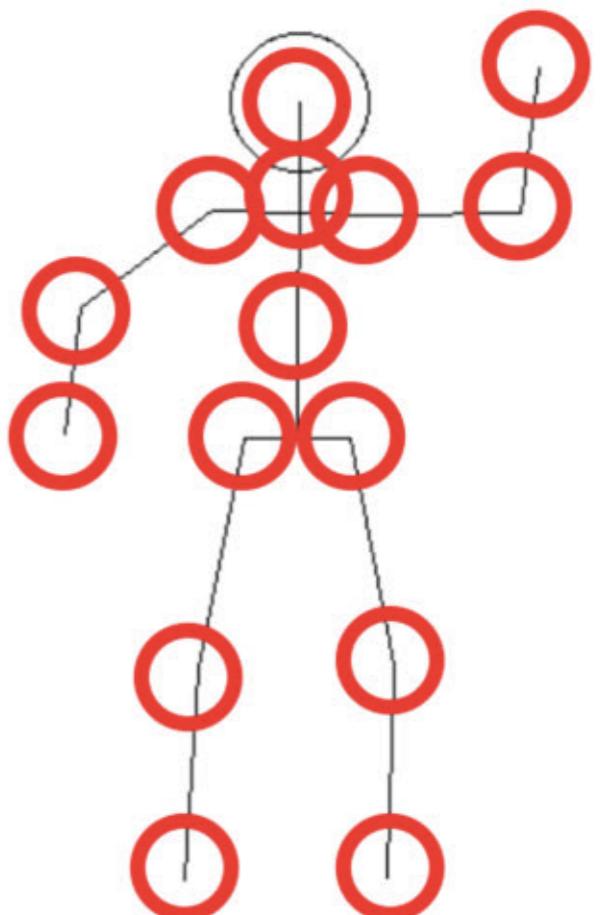


Figure 4

then use that object to get more-detailed information, such as the x-coordinate and y-coordinate of the joint on the screen. (Refer to **Figure 3**.) The **Joint** class also contains constants for identifying each individual joint.

The setup. Our own scenario gets initialized in the **PaintWorld** class. We will not discuss the code in detail here; you can read the code yourself if you are curious, or you can ignore it for now. But here is a summary of its purpose.

The **PaintWorld** object, in every Greenfoot act cycle, looks for all *tracked users* (that is, all users who are in the view of the Kinect camera) and creates a **Canvas** object for every user it detects.

A **Canvas** is a Greenfoot object that has a transparent image the size of the entire Greenfoot world, and it is linked to one tracked user. Since it is entirely transparent, it will initially be invisible. So if Kinect currently sees, for instance, two users, there will be two **Canvas** objects in the Greenfoot world acting as (invisible) painting layers, and each user paints

on his or her own layer.

The Canvas class. We will do all our work in the **Canvas** class. This is where all the fun is.

Let's have a quick look at the constructor code shown in **Listing 1**.

We can see that the **UserData** object for the user associated with this canvas is received and stored for later use. Then a transparent Greenfoot image on which to paint is created. Lastly, a random color is selected from an array of predefined colors (defined further up in the class), and it is set as the color for future painting operations.

The act method.

Although the constructor is executed only once, the Greenfoot **act** method is executed repeatedly as long as the scenario is running. It, too, is fairly simple (see **Listing 2**).

In this method, we use the **user** object to get information about the right hand by using the **user.getJoint(Joint.RIGHT_HAND)** method call. This will give us an object of type **Joint**.

We can then use the joint's **getX()** and **getY()** methods to determine the right hand's position on the screen and then

GET PHYSICAL
By connecting Microsoft Kinect to Greenfoot, **we can create games that we can control with our bodies:** No hitting of keys and no mouse clicks—just jumping up and down and waving your arms in the air.

LISTING 1 LISTING 2 // LISTING 3 // LISTING 4

```
public Canvas(int width, int height, UserData user)
{
    this.user = user;
    setImage(new GreenfootImage(width, height));
    getImage().setColor(
        colors[Greenfoot.getRandomNumber(colors.length)]);
}
```

 [Download all listings in this issue as text](#)

to paint a circle where the hand is. This is achieved by using the **fillOval** method with the x-coordinate and y-coordinate, and using 20 as both width and height of the circle.

fillOval is a method of the **GreenfootImage** class, which we can call on the canvas's image. We retrieve the image using the **getImage()** method, which is one of the predefined methods of the Greenfoot **Actor** class.

One last thing that we can see in **Listing 2** is an **if** statement that checks whether the right hand is the joint nearest to the camera. We draw only if it is, so we can move the right hand without painting just by holding it farther away from the screen.

This short code segment shows how easy it is to follow a particular joint and to draw at its location.

Now, let's jump in and write some code of our own.

Task 1: Erasing the Screen

Our first task is to erase the screen when we lift our left hand over our head.

Erasing the screen is very easy, since the **GreenfootImage** class has a **clear()** method. So the call to erase a user's image is simply the following:

getImage().clear();

All that is left to do is to check whether the left hand was raised above the head. We can do this by retrieving the y-coordinate for both the left hand and the head, and then comparing them, as shown in **Listing 3**.

Now we just need to put these two bits together in an **if** statement, as shown in **Listing 4**.

Try it out. Add this code to the **act** method of your **Canvas** class (after the painting code), compile,

LISTING 5

```
if (user.getJoint(Joint.RIGHT FOOT).getY() <
    user.getJoint(Joint.LEFT KNEE).getY())
{
    getImage().setColor(
        colors[Greenfoot.getRandomNumber(colors.length)]);
}
```

 [Download all listings in this issue as text](#)

and then run the scenario. Raising your left hand should now erase the screen.

Task 2: Changing the Painting Color

For our second task, we will change our painting color. If we want to come up with a control for determining when to perform such a task, there are endless gestures to choose from. However, it is good to choose something that is relatively easy to check. Let's change the color when the right foot is lifted (since it is easy to check whether the right foot is higher than, say, the left knee).

Since this control is very similar to the one before (right foot higher than left knee instead of left hand over head), the `if` statement is very similar and should be easy to understand now. All that remains to be determined is how to change the painting color.

We do this by picking a random color out of our predefined color array, and then setting it using the `setColor` method of `GreenfootImage`, as shown in Listing 5.

Conclusion

In this article, we have seen that it is very easy to get started writing engaging games that we can control with our bodies by using the Microsoft Kinect sensor, Greenfoot, and Java. Setting up the software was the hardest bit, and now you are ready to try some more tasks.

In Part 2 of this series, we will see some more examples of programming using Greenfoot with Kinect, so check back soon. </article>

[LEARN MORE](#)

- [Greenfoot download page](#)
 - [Instructions for Greenfoot and Kinect setup](#)

3 Billion

Devices Run Java

ATMs, Smartcards, POS Terminals, Blu-ray Players, Set Top Boxes, Multifunction Printers, PCs, Servers, Routers, Switches, Parking Meters, Smart Meters, Lottery Systems, Airplane Systems, IoT Gateways, Programmable Logic Controllers, Optical Sensors, Wireless M2M Modules, Access Control Systems, Medical Devices, Building Controls, Automobiles...



#1 Development Platform

ORACLE



BEN EVANS AND
CHRIS NEWLAND

BIO

Part 2

Understanding Java JIT Compilation with JITWatch

Learn how to examine JIT logs to see what the JIT compiler is doing (and why).

Oracle's Java HotSpot VM is equipped with a highly advanced just-in-time (JIT) compiler. It is one of the most important parts of Java HotSpot VM, and yet many Java developers do not know much about it or how to ensure that their applications are making the most of it.

Java HotSpot VM is able to produce a very detailed log of exactly what the JIT compiler is doing (and why). However, the log is complex and difficult to read, and reading the log requires an understanding of the techniques and theory that

underlie JIT compilation.

In [Part 1 of this series](#), we covered some of the basics of JIT compilation to provide a context for developers who might not be familiar with this topic. In this article, we will discuss a new, free tool designed to process JIT compilation logs. If you aren't familiar with JIT compilation, you might wish to read the previous article before tackling this one.

Introducing JITWatch

JITWatch is a graphical visualization and analysis tool for understanding the JIT

GET THIS TOOL
JITWatch is a graphical visualization and analysis tool for understanding the JIT compilation of applications. It's open source and is being developed as part of the Adopt OpenJDK project.

compilation of applications. It's open source software and is being developed as part of the Adopt OpenJDK project. The source code for JITWatch is available from [GitHub](#).

JITWatch needs a JIT compilation log file in order to work. As mentioned in the previous article, there are two different log formats that Java HotSpot VM can produce. JITWatch requires the more verbose, XML-based log that is produced by the [LogCompilation](#) switch. JITWatch requires the [TraceClassLoading](#) switch as well, so the full set of switches needed to produce a suitable log for JITWatch is the following:

```
-XX:+UnlockDiagnosticVM  
Options  
-XX:+LogCompilation  
-XX:+TraceClassLoading
```

This will result in an output log consisting of a very large XML file that contains a high level of detail about the operation of the JIT compiler and the compilation decisions it makes.

To get started with JITWatch, visit its [GitHub page](#). From there, you can either follow the instructions to build your own binary, or you can follow the links to download prebuilt binaries from the JITWatch continuous integration server.

Note: If you're building your own binary from source code, remember to run [mvn package](#) first to create the JITWatch JAR file, so the GUI will start up.

First Look at JITWatch

You can use the [launchUI.sh](#) command to start up JITWatch in GUI mode (on



Microsoft Windows, use [launchUI.bat](#) instead). You'll be presented with the simple welcome screen shown in **Figure 1**.

From here, you can load a JIT compilation log. You might not have one of these on hand, so let's look at a JITWatch script for generating a test log. Before we do that, however, it's worth discussing assembly code and its importance to JITWatch.

Working with Assembly Code

To get the most out of working with JITWatch, we need to look at the machine code generated by the JIT compiler. This is done by using a HotSpot disassembler (HSDIS) plugin to turn the generated code back into human-readable assembly code. This plugin then enables us to use the [PrintAssembly](#) flag to get assembly code dumps, which means that we can get a great deal more out of JITWatch.

The plugin that is usually used is the [base-hsdis](#) plugin, maintained by John Rose and others. It is available from the Downloads section of [this website](#).

Download the correct plugin for your operating system, and copy it to the directory that contains the Java Virtual Machine (JVM) library file ([libjvm.so](#) on Linux, [libjvm.dylib](#) on Mac, [libjvm.dll](#) on Microsoft Windows). On Java SE 8, this

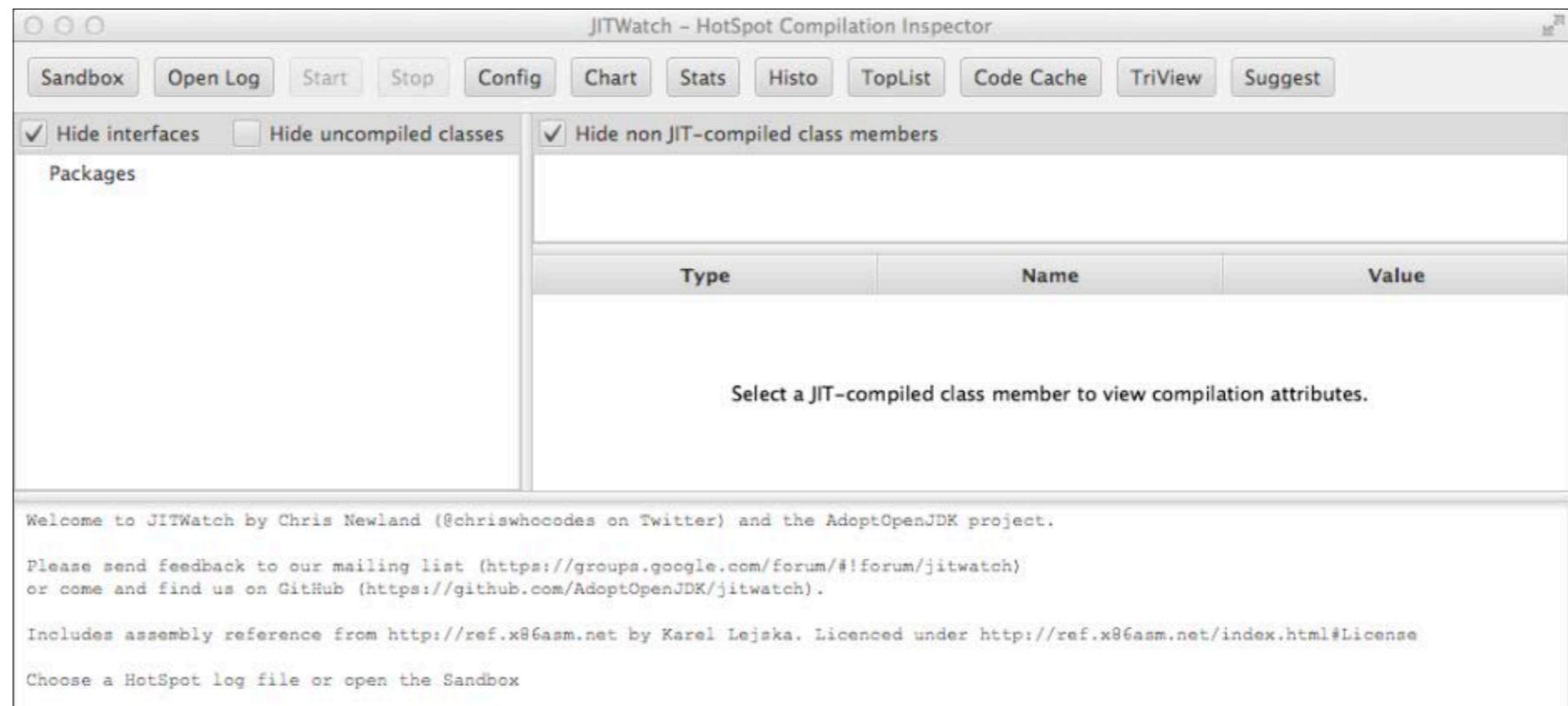


Figure 1

directory is [\\$JAVA_HOME/jre/lib/server/](#).

Note: For Mac users, try the [bsd-libhsdis-i386.dylib](#) file and rename it to [hsdis-amd64.dylib](#) before copying it to the Java library directory.

The instructions [here](#) describe how to build your own HSDIS plugin from scratch.

Many Java developers have only limited experience working with assembly code, so the prospect of using HSDIS seems daunting. However, with time and some diligence, a working knowledge of assembly code is well within the grasp of the majority of developers. Here are some pointers to help

newcomers get comfortable working with assembly code:

- Don't panic. Assembly code isn't the end of the world.
 - Get a good reference (online resource or a textbook).
 - Understand the difference between different syntaxes for writing down assembly code. By default, HSDIS uses AT&T syntax, but it can be made to use Intel syntax instead.
 - If you're familiar with C programming, remember that C compilers can be instructed not to emit machine code but to instead stop at the assembly code stage (for example, by using this command: [gcc -S](#)). Doing this can provide assembly code that corresponds to simple examples, which can aid understanding for beginners.
 - Remember that Java HotSpot VM is a complicated runtime that has to deal with both compiled and interpreted methods. It also heavily optimizes compiled code and needs to cater to speculative optimizations that might need to be backed out. It's OK not to understand the reason for everything that's going on in an assembly code dump that came from Java HotSpot VM.
- Let's create a log that includes

//java architect /

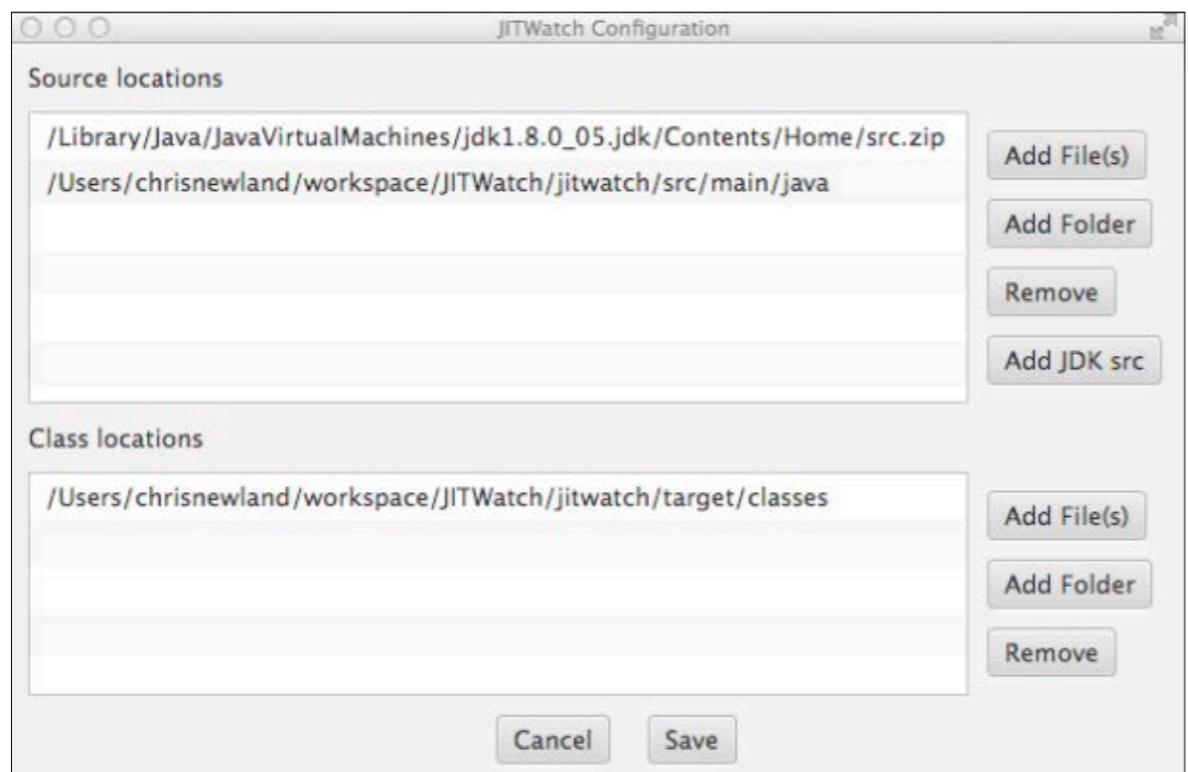


Figure 2

assembly code, so we can see how JITWatch deals with it.

Creating a Test Log

JITWatch ships with a test script (see Listing 1) called `makeDemoLogFile.sh`, which can generate a log for novice users to practice on.

Once you've run this script, you should end up with a log file in the `jitwatch` directory. The log file will be called something like `hotspot_pid20053.log`. To work with this log file, start the JITWatch GUI and open the configuration dialog box by clicking the **Config** button. Doing this should open the dialog box shown in Figure 2.

Next, we need to configure the locations at which to load source and class files so that JITWatch can analyze the log file properly. We want to have the following files available:

- The JDK source (from `src.zip`). Use the handy **Add JDK src** button for configuring this location.
 - Java source files for the demo, which are in the directory `src/main/java`. Use the **Add Folder** button to configure this location.
 - Class files for the demo, which are contained in `target/classes`. Use the **Add Folder** button to configure this location.
- Then click **Save** to close the configuration dialog box, and then click

LISTING 1

```
$ ./makeDemoLogFile.sh
java version "1.8.0_05"
Java(TM) SE Runtime Environment (build 1.8.0_05-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.5-b02, mixed mode)
VM Switches -XX:+UnlockDiagnosticVMOptions
-XX:+TraceClassLoading -XX:+LogCompilation
-XX:-TieredCompilation -XX:+PrintAssembly -XX:-UseCompressedOops
Building example HotSpot log
Java HotSpot(TM) 64-Bit Server VM warning: PrintAssembly is
enabled; turning on DebugNonSafePoints to gain additional output
```

Done

[Download all listings in this issue as text](#)

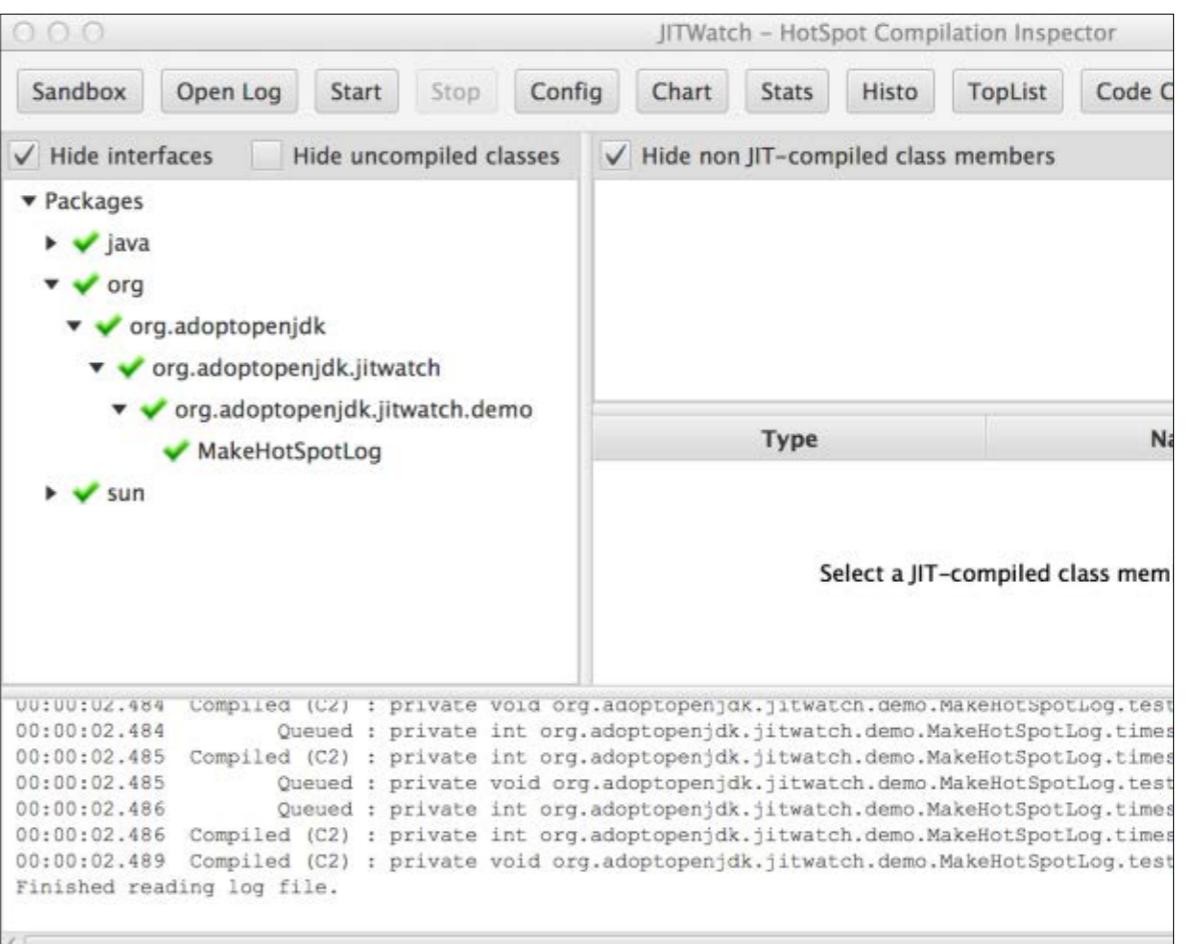


Figure 3

//java architect /

Open Log in the main window to load in the test log.

JITWatch prompts you to click **Start** to process and parse the log, so do that next. You should end up with a screen similar to **Figure 3** that shows, in the left panel, a view of packages containing compiled methods.

If we drill into the packages, we can identify individual classes and methods and see how they've been treated by the JIT compiler, as shown in **Figure 4**.

There are also context-sensitive menus that work directly from the results page. See **Figure 5**.

We can even drill in to particular compilation trees by selecting the **Show compile chain** option, which leads to a screen that is similar to **Figure 6**.

Note: Tooltips available in the compile chain view show more detail about how the compiler handled methods. For example, in **Figure 6**, the tooltip indicates that the compiler compiled the method, but the method was too big to inline.

We're now ready to start working with the log in detail, so let's examine the "triview" screen next.

JITWatch Triview Screen

The triview is one of the most useful JITWatch screens. It shows three different views of the same

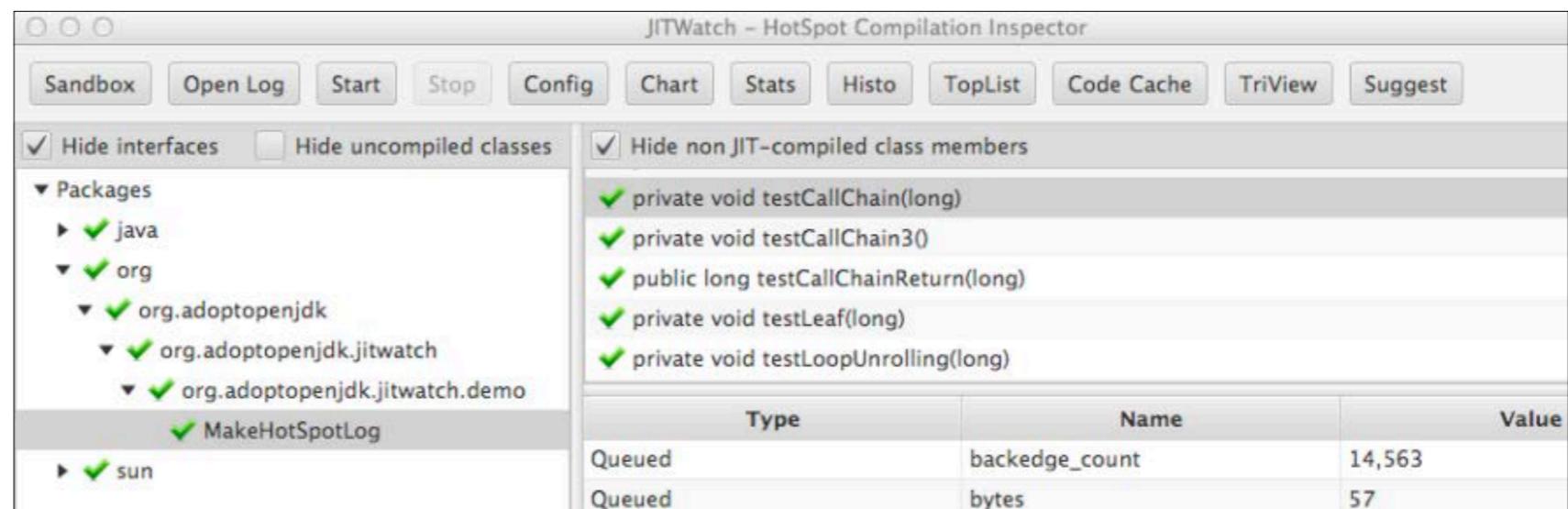


Figure 4

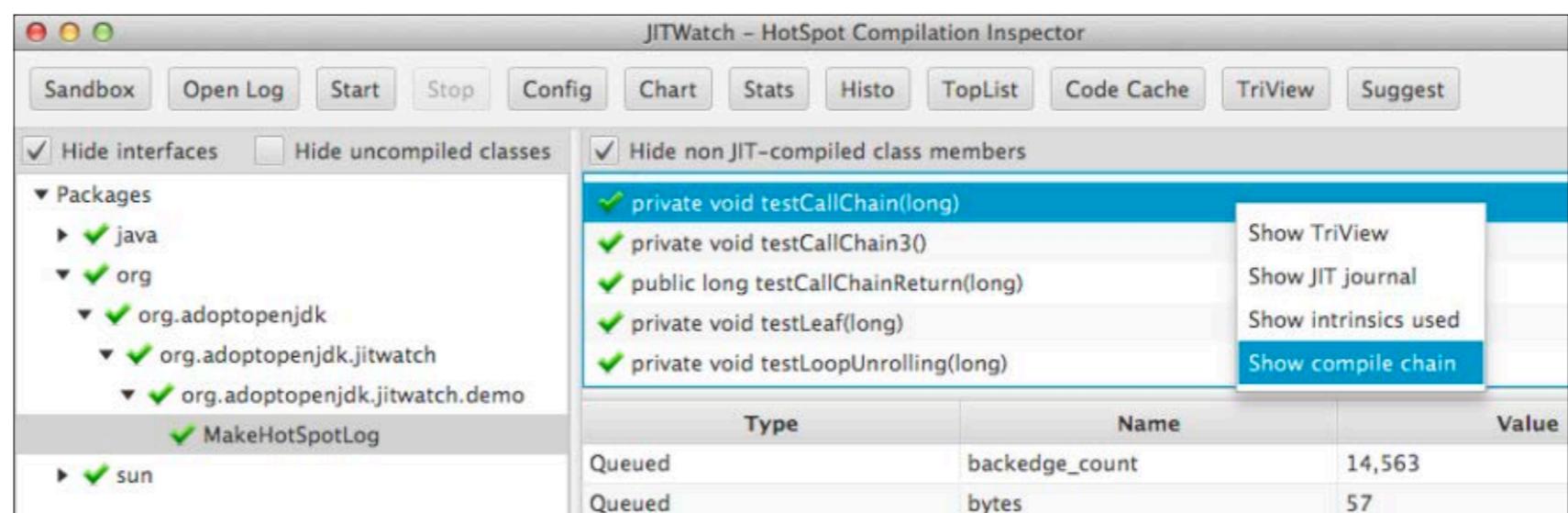


Figure 5

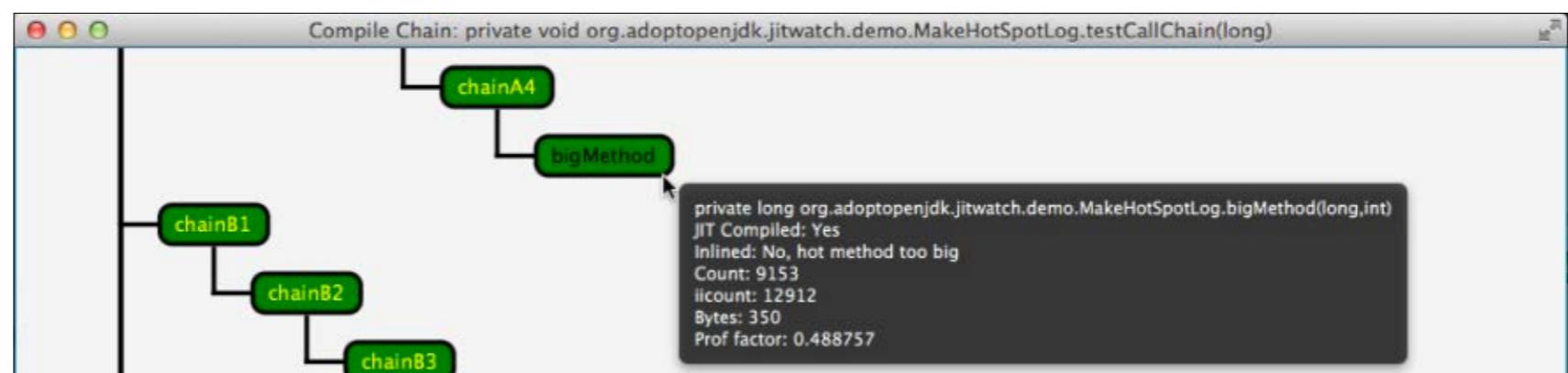


Figure 6

//java architect /

code: Java source code, bytecode, and disassembled machine code.

Figure 7 shows what this view looks like.

Looking at different methods and selecting different bits of Java code causes the triview to respond by showing the equivalent line of bytecode and assembly code that Java HotSpot VM produced. This enables you to see what actually happened to a particular method, which is helpful when learning both bytecode and assembly code.

Figure 7 looks a bit complicated, but that is a side effect of using the demo log. Let's look at

a simpler example by starting to use JITWatch to analyze how Java HotSpot VM handles specific JIT compilation behaviors.

Inlining

One of the common JIT compilation techniques that we saw in the first article is *Inlining*. This is a technique through which the body of a called method (the *callee*) is directly pasted into the body of a method that is calling it (the *caller*), and the call from caller to callee is eliminated. Inlining can provide better performance by removing the need to do virtual (method) dispatch for

The screenshot shows the TriView interface with the following details:

- Class:** org.adoptopenjdk.jitwatch.demo.MakeHotSpotLog
- Member:** private void testToUpperCase(long)
- Panels:**
 - Source:** Shows Java code for methods chainC3, testLeaf, leaf1, leaf2, leaf3, leaf4, and testToUpperCase.
 - Bytecode (double click for JVMS):** Shows the bytecode dump for the testToUpperCase method, including comments explaining the code flow and method calls.
 - Assembly:** Shows the corresponding machine assembly code for the same method.

Figure 7

LISTING 2

```
public class SandboxTest {  
  
    public int a(int x) {  
        return 1 + b(x);  
    }  
  
    public int b(int x) {  
        return 1 + c(x);  
    }  
  
    public int c(int x) {  
        return 1 + x;  
    }  
}
```

[Download all listings in this issue as text](#)

//java architect /

the callee and various other bits of bookkeeping.

Let's look at the example shown in **Listing 2**. This code can be loaded into JITWatch. (There's a convenient feature called Sandbox—which we'll meet properly in the next article—that can help with this.) The result will look like **Figure 8**.

In the Java code, we've highlighted the line `return 1 + b(x);`. This is represented in the bytecode by the `iadd` instruction, but note that

there's an `invokevirtual` instruction directly before that (and the comment tells us that the method being called is `b()`). So the bytecode is a faithful representation of what's written in Java.

The assembly code, however, is a different story. The highlighted line shows that we're adding 3 to the register `%eax`, but where did the constant 3 come from? If we follow the Java code, we can see that the net result of calling `a(x)` will be to return `x + 3`. The methods `b()` and

`c()` are very simple and prime candidates for inlining. This is exactly what's happened, and all references to those methods are gone in the assembly code.

Note: The JIT compiler has both static and dynamic inlining rules. These example methods are small enough to be caught by the static "under 35 bytecodes" rule. In real applications, the dynamic policy would be used in most cases. Remember that the Sandbox is a learning tool for understand-

ing aspects of the JIT compiler's behavior; it is not a professional tuning tool.

Suggestion Tool

We've met JITWatch and seen how to generate compilation logs from applications and then work with them in the JITWatch GUI screens (especially the triview). We've shown how a simple JIT compilation technique shows up in the compilation log and how we can understand it using JITWatch.

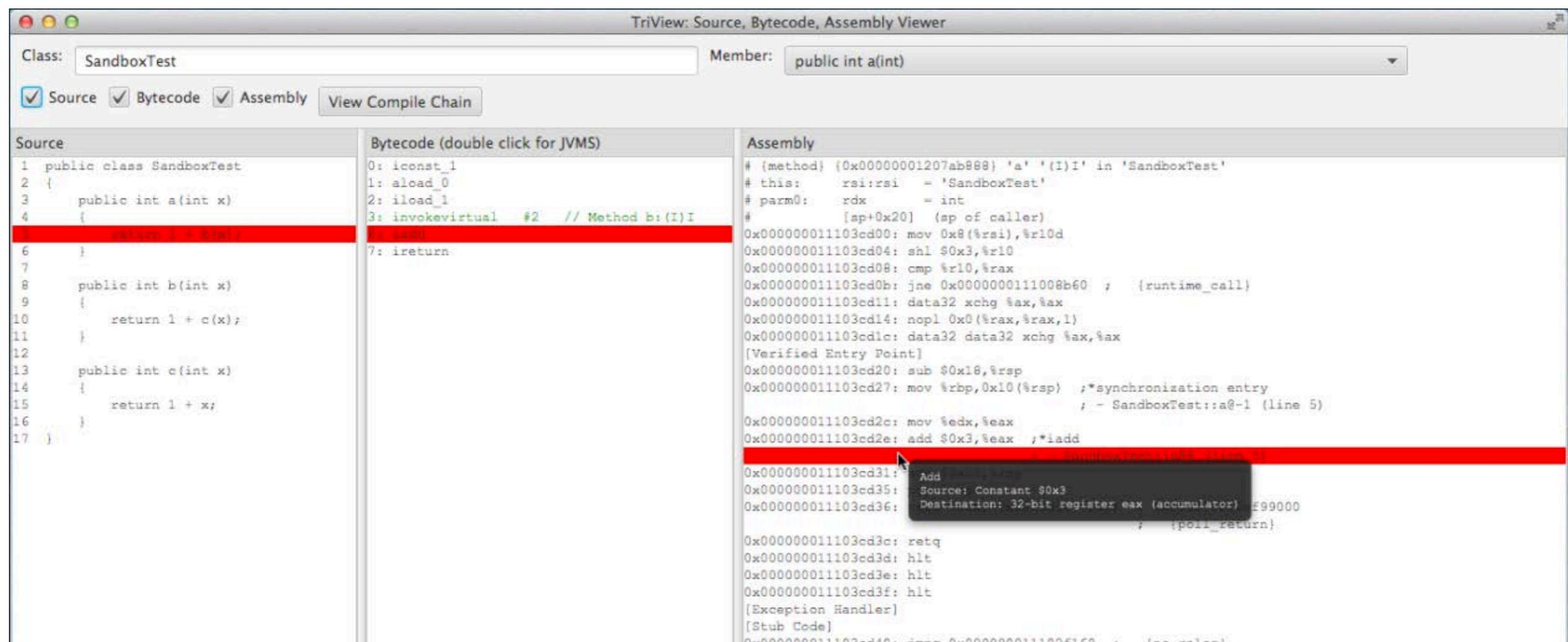


Figure 8

JITWatch Code Suggestions			
Score	Type	Caller	Suggestion
14078	Branch	java.lang.Integer public static Integer valueOf(int)	Method contains an unpredictable branch at bytecode 3 that was observed 28156 times and is taken with probability 0.49826. It may be possible to modify the branch (for example by sorting a collection before iterating) to make it more predictable.
13001	Branch	java.lang.Integer public static Integer valueOf(int)	Method contains an unpredictable branch at bytecode 3 that was observed 26001 times and is taken with probability 0.499212. It may be possible to modify the branch (for example by sorting a collection before iterating) to make it more predictable.
12912	Inlining	org.adoptopenjdk.jitwatch.demo.MakeHotSpotLog private long chainA4(long)	The call at bytecode 3 to Class: org.adoptopenjdk.jitwatch.demo.MakeHotSpotLog Member: private long bigMethod(long,int) was not inlined for reason: 'hot method too big' The callee method is 'hot' but is too big to be inlined into the caller. You may want to consider refactoring the callee into smaller methods. Invocations: 12912 Size of callee bytecode: 350
12912	Inlining	org.adoptopenjdk.jitwatch.demo.MakeHotSpotLog public void tooBigToInline(int)	The call at bytecode 15 to Class: org.adoptopenjdk.jitwatch.demo.MakeHotSpotLog Member: private long bigMethod(long,int) was not inlined for reason: 'hot method too big' The callee method is 'hot' but is too big to be inlined into the caller. You may want to consider refactoring the callee into smaller methods. Invocations: 12912 Size of callee bytecode: 350

Figure 9

To finish, let's briefly discuss another JITWatch feature: the suggestion tool. For some developers, this will be the jewel in the crown, because it can describe the reason why a certain JIT compilation decision was made (and why various JIT compilation techniques failed). **Figure 9** shows an example.

In **Figure 9**, we can see some examples of a couple of different failure types. In one case, branches within methods could not be reliably predicted, and in the other, the methods were too big to be

inlined. Both of these would cause extra overhead and make JIT compilation less efficient than it would otherwise be.

Conclusion

JITWatch provides a convenient way to view the optimization decisions made by the Java HotSpot VM JIT compiler during the execution of code, and it can help you understand how decisions made at the Java source code level affect the performance of your program.

JITWatch is open source software

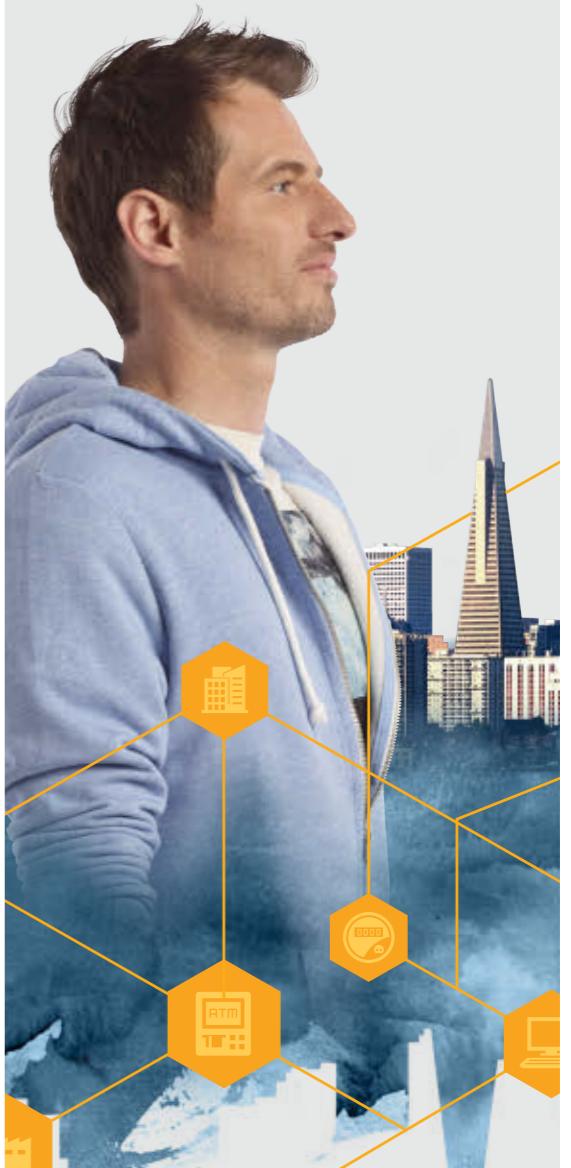
being developed by Chris Newland and the [OpenJDK Adoption group](#) with support from the [London Java Community](#), the official Java user group (JUG) for developers based in London. Please contact us via the JITWatch GitHub project if you would like to help. Your bug reports, assistance, and interest are very welcome. <[article](#)>

LEARN MORE

- [GitHub page for JITWatch](#)

CREATE THE FUTURE

oracle.com/java



Java™

ORACLE®



SHAY SHMELTZER

BIO

Java Apps on iOS and Android—Now a Reality

Build mobile apps with Oracle Mobile Application Framework, a Java-based hybrid mobile platform.

Let me guess a couple of things about you. First, you are fluent in Java. Second, you have a smartphone, tablet, or both within reach. Am I right? Well, with the explosion of mobile device popularity, plus the fact that you are reading *Java Magazine*, these were not really tough guesses to make.

What might be a bit tougher is answering this

question: How do you take your Java skills and start building applications for the mobile devices that you—and your customers—have? That is the focus of this article.

As a Java developer, you can easily address the development of the back-end system that is accessed by mobile apps. But what are your options when it comes to the

actual user interface (UI) and mobile app development?

A variant of Java is the language that is being used to develop Android-based applications, but limiting your app to just that market segment ignores a big chunk of users who are using a different mobile operating system (OS)—specifically iOS, which drives all Apple mobile devices.

To solve the need for cross-device mobile development, many developers turn to hybrid mobile development. These apps run inside a container that is available as a native app for each platform, allowing you to write a single app that runs on multiple platforms. The tricky part is that most of the hybrid mobile solutions today rely on coding HTML5 plus JavaScript code. Do you really need to

abandon Java to go mobile?

The answer is no. This article will introduce you to a Java-based hybrid mobile platform that lets you continue developing in the language and tools you are using today, and enables you to run your apps on both iOS and Android.

Welcome to the world of Oracle Mobile Application Framework.

A Mobile Framework for Java Developers

In June 2014, Oracle released [Oracle Mobile Application Framework](#), a solution that runs Java apps on both iOS and Android devices. Java code handles every layer of a mobile app, from the generating the UI and responding to UI events, to dealing with data objects and providing access to remote services.



This demo shows how to develop a mobile application with Oracle Mobile Application Framework and Oracle Enterprise Pack for Eclipse.

//java architect /

(including access to on-device mobile databases), all the way to handling the interaction with mobile device features such as the Global Positioning System (GPS), the camera, SMS, and more.

Furthermore, going beyond just language similarities, the framework also leverages an architecture you are probably familiar with: model-view-controller (MVC). If you are currently doing MVC-based web development with Java, for example with JavaServer Faces (JSF), you are going to see familiar concepts that will make picking up this framework even more intuitive.

And to further simplify your adoption, the framework has extensive integrated development environment (IDE) support in both Eclipse, through Oracle Enterprise Pack for Eclipse, and Oracle JDeveloper.

Hybrid Mobile Architecture and You

There are three approaches to building mobile applications: native, web, and hybrid.

In the native approach, you work with the SDK and tools for a specific mobile OS, which means that your application is not portable to other devices. In the web approach, you are still building server-side served web pages and accessing them from the browser on your mobile device, but this doesn't

actually leverage the capabilities of modern devices and provides a less-than-ideal user experience.

To overcome the challenge of developing mobile apps with those two approaches, the industry has turned to hybrid mobile applications as the preferred solution for enterprise applications.

Hybrid mobile applications rely on a container that runs your code on the device. The container is available in native format for multiple operating systems, which allows you to code once and run on many platforms.

Does the idea sound familiar? It should, because it is similar to the way the Java Virtual Machine (JVM) provides portability for Java code across operating systems. In the mobile world, most of the hybrid solutions in the market are based on coding all your logic in JavaScript and coding the UI with HTML5. The container then uses the [WebView](#) component of each mobile OS to run the logic and render the UI.

The Oracle Mobile Application Framework hybrid container, while still supporting the ability to code features in JavaScript and HTML5, adds Java and an MVC-based framework as another way to build mobile apps. An Oracle Mobile Application Framework container includes a lightweight JVM that can run your Java logic, a [WebView](#) component

that can show your user interface, an Apache Cordova-based layer for interaction with device features, an encrypted local SQLite database for on-device and offline data storage, a push notification handler, and a security layer for authentication and authorization.

With this approach, Oracle Mobile Application Framework can actually run applications that have a mix of the native and web development approaches. This way, an organization with both Java and JavaScript developers can have them collaborate on the same mobile app.

Powerful Development Framework

Hand-coding every aspect of an application is a recipe for a delayed delivery, which is why in the world of Java, we rely on frameworks to accelerate development. Oracle Mobile Application Framework provides a complete and powerful framework that can accelerate mobile application delivery.

In the world of Java development, instead of hand-coding low-level UI protocols such as HTML, we have opted for the simpler component-based approach to UI development (JSF and Google Web Toolkit are a couple of examples).

For mobile user interfaces, Oracle Mobile Application Framework

provides a set of more than 80 UI components that can be used to create compelling mobile interfaces. With components that range from basic input and selection to advanced charting, mapping, and data visualization, you can define user interfaces much faster.

Figure 1 shows an Oracle Mobile Application Framework UI on an iPhone.

What if a component is missing? You can build your own components with HTML5 and JavaScript

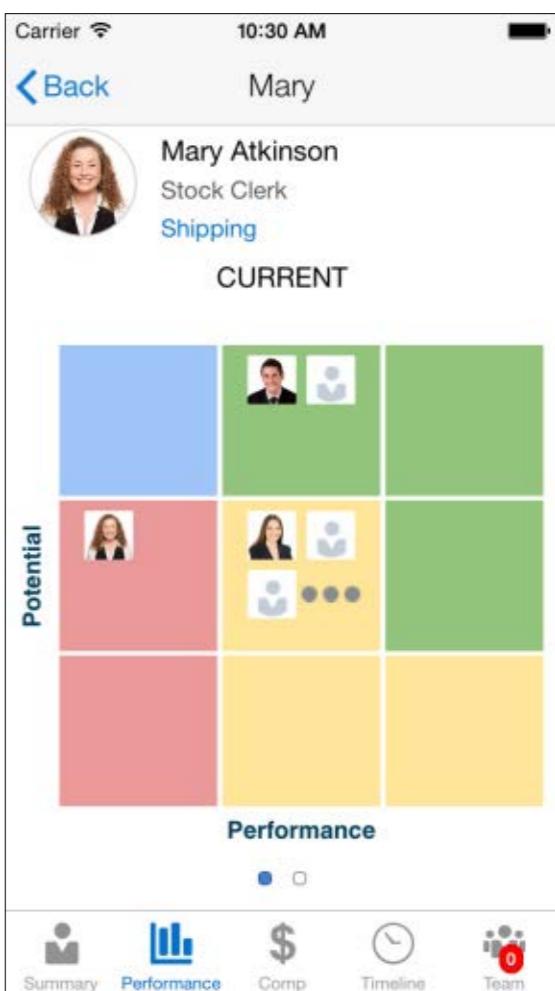


Figure 1



and add them to the list of components you can use later on.

Past experience with spaghetti code-based coding (servlets, JavaServer Pages) taught us about the importance of proper functional layer separation when building and maintaining apps, and in Oracle Mobile Application Framework, this is implemented with the MVC design pattern.

For the controller layer, Oracle Mobile Application Framework provides a process flow engine that allows you to define navigation between pages in your applications and also include method calls, decision points, and other flows in a flow. Code in the controller layer is written in simple Java classes that are similar in concept to managed beans in JSF.

Managed beans can also contain data that will be exposed in the UI layer. Oracle Mobile Application Framework provides various scopes for beans to better handle memory allocation. In addition, data can be incorporated from a remote server through both REST (JSON or XML) and SOAP services.

Oracle Mobile Application Framework uses a declarative data-binding layer to reduce the amount of code you need to manually write to hook up your UI components to the data and business services. At development time, a simple drag

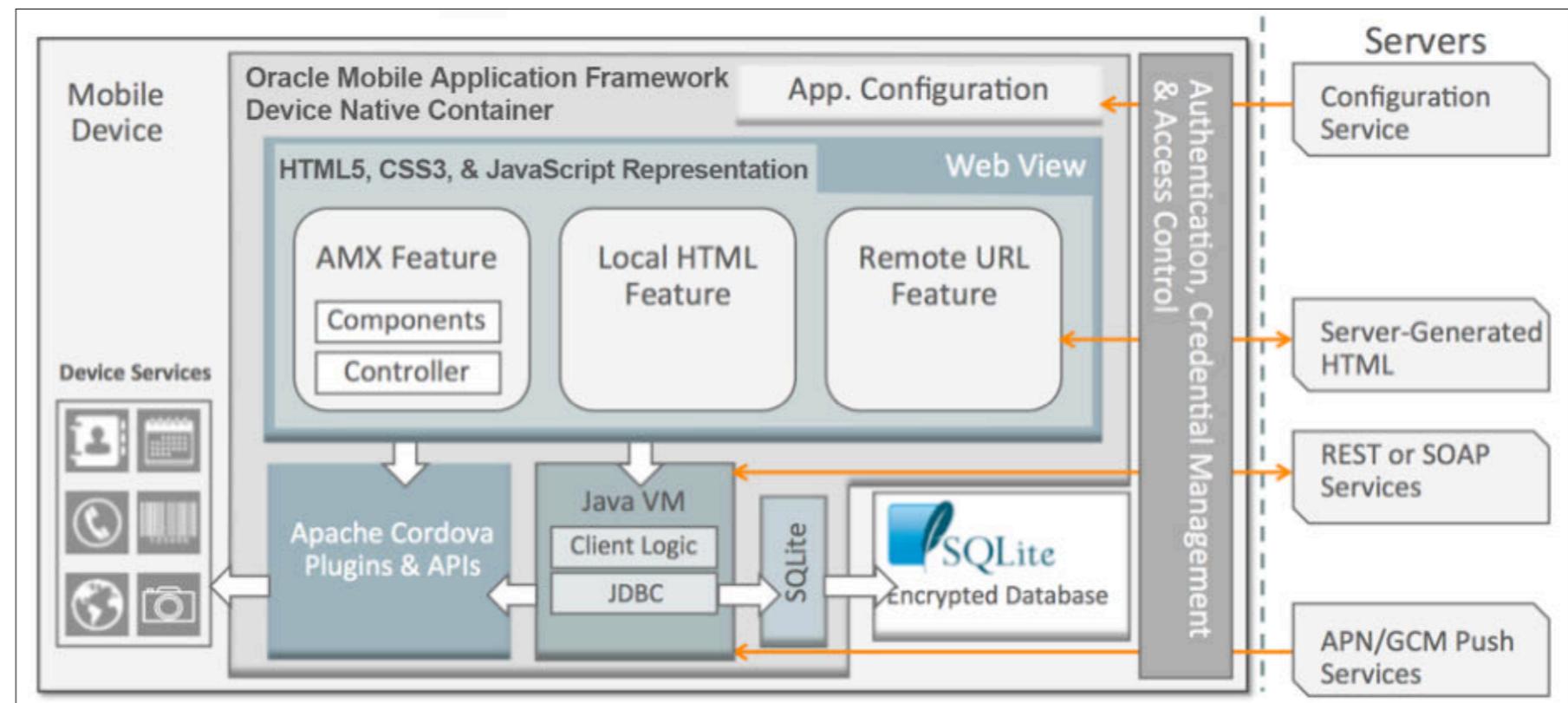


Figure 2

and drop will create the expression language that will connect a UI component to a datasource.

Building Truly Mobile Applications

A truly mobile application leverages the platform it runs on. For example, a truly mobile expense app lets you take pictures of receipts with the device's camera, a truly mobile product catalog uses GPS on the device to show the nearest store where you can buy an item, and a truly mobile HR system lets you quickly add an employee you are looking at to the contact list on your phone.

Oracle Mobile Application Framework lets you build these types of truly mobile apps by exposing device features for easy integration into your application, as shown in **Figure 2**. Using the open source Apache Cordova solution, Oracle Mobile Application Framework provides Java and JavaScript APIs that let you invoke the camera, GPS, SMS, e-mail, and other services directly from your code.

Furthermore, the framework supports the Cordova plugin architecture to let you add other device-specific capabilities to your app for features such as barcode scanning and temperature measurement.

Another aspect of mobile apps that Oracle Mobile Application Framework can leverage is push notification. Leveraging either the Apple or Google push server, a mobile app can receive push notifications and react to them with simple Java listeners that you code.

Oracle Mobile Application Framework further supports offline scenarios for mobile apps with a built-in encrypted SQLite database that apps can use to store data. Access to files on the local device's file system is also supported from within Oracle Mobile Application Framework.

With all of these capabilities at



//java architect /

hand, you can build applications that truly change the way people interact with back-end systems.

IDE Integration

Having a framework is great, but without proper tooling, using a framework can actually reduce your productivity.

To simplify development using Oracle Mobile Application Framework, Oracle built extensive IDE features in both Eclipse and Oracle JDeveloper, providing a complete integrated and declarative development experience.

Download the SDKs for Android and iOS, set the paths in your IDE of choice, and in addition to developing code, you'll be able to directly package, deploy, debug, and run your mobile apps on emulators or devices, all from inside the IDE.

The IDEs provide a productivity boost for development through wizards that help set up projects, declarative editors for configuration files, visual editors for your controller and UI layers, interactive structure panes, and property inspectors that reduce typos. All of this is in addition to the powerful coding features supported by the code editors in both Eclipse and Oracle JDeveloper. **Figure 3** shows an example of the Oracle Mobile Application Framework features in Eclipse.

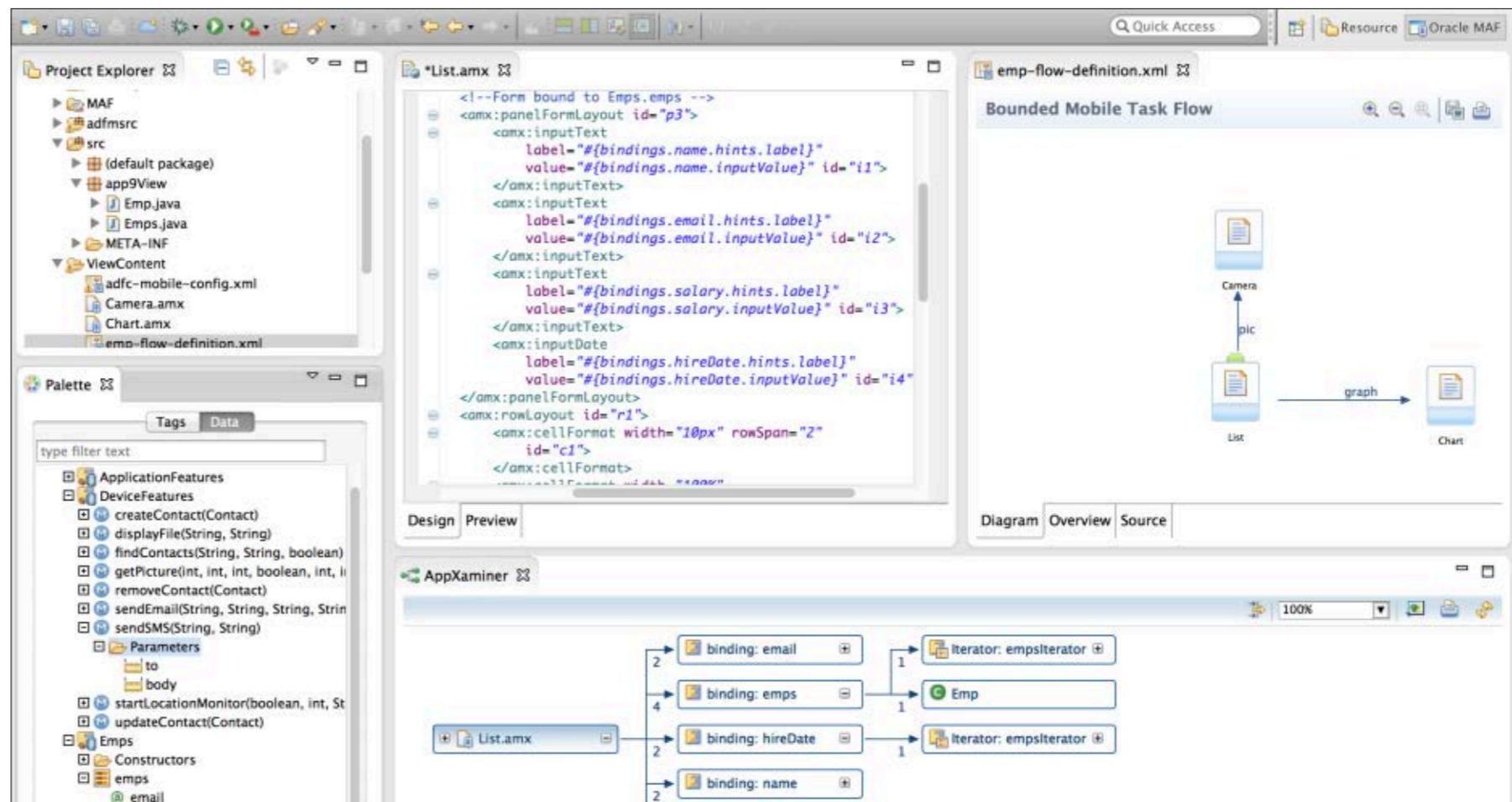


Figure 3

Want to get an impression of the overall development experience? Watch these 10-minute demos showing end-to-end development with Oracle Mobile Application Framework in [Eclipse](#) and [Oracle JDeveloper](#).

Getting Started

Getting started with Oracle Mobile Application Framework is really simple: [Download](#) your preferred IDE and follow the [tutorials](#) to properly set up your environment and build your first application.

Want to learn more? A quick, [free online course](#) will get you through the basics, and more-extensive training is available for free on the Oracle Mobile Platform [YouTube channel](#).

So what are you waiting for? It has never been easier for Java developers to become mobile developers.

Conclusion

Mobile devices are all around us, and on-device mobile application development skills are in demand. With the new Oracle Mobile

Application Framework, you can leverage the Java skills you already have to build this new generation of applications. With a productivity-boosting framework that is open for extensibility, developing cross-platform mobile apps with Java has never been simpler. <[article](#)>

LEARN MORE

- [Oracle Mobile Application Framework website](#)
- [Getting Started tutorials](#)
- [YouTube-based training](#)

CAREER OPPORTUNITIES

MANAGE YOUR CAREER PATH AT DICE.COM

Java Jobs at your Fingertips

Dice is the leading career site for information technology and engineering professionals.

312,503
Java Jobs



*Register, upload your resume and
browse our wide range of tech and
IT positions.*



priceline.com®

“For us, Java has become a way of life”

- Mike Diliberto, priceline.com CIO



We live Java. 24/7.

We're Hiring

Priceline.com is looking for Java Developers.
Apply today at: www.priceline.com/careers



MICHAEL HÜTTERMANN

BIO

Get Your Java Applications Enterprise-Ready with Oracle WebLogic Server

Make your applications more robust, secure, highly available, and scalable.

This article introduces Oracle WebLogic Server 12c application server, provides an overview of its features, and describes some useful tools you can use to get started with writing and hosting your Java EE applications. We'll also discuss how to integrate this server with the persistence layer using Active GridLink for RAC datasources, how to effectively manage the server using its WebLogic Scripting Tool feature, and how to make use of the cloud by integrating the server with Oracle Java Cloud Service.

About Oracle WebLogic Server 12c

In a nutshell, Oracle WebLogic Server is a scalable Java EE application server that offers a full implementation of the Java EE 6.0 specification, which is a standard set

of APIs for creating distributed Java EE applications. In addition to providing the Java EE implementation, Oracle WebLogic Server enables enterprises to deploy robust, secure, highly available, and scalable environments. These attributes make Oracle WebLogic Server enterprise-ready. You can also configure Oracle WebLogic Server to monitor and tune application throughput automatically, so it is an excellent vehicle to underpin your DevOps approach.

Oracle WebLogic Server offers support for the following:

- Web applications, by providing dynamic web pages based on the Java Servlet and JavaServer Pages (JSPs) specifications

▪ Web services, by pro-

viding a shared set of functions that are available to other systems on a network

- XML capabilities, by providing features for data exchange and ways to store content independent of its presentation
- Java Message Service (JMS), which enables applications to communicate with one another

through the exchange of messages

- JDBC, which provides pooled access to database management system (DBMS) resources
- Resource adapters, which provide connectivity to enterprise information systems (EIS)
- Enterprise JavaBeans (EJBs), which provide Java

EDITION	FEATURES INCLUDED
ORACLE WEBLOGIC SERVER, STANDARD EDITION	INCLUDES SUPPORT FOR JAVA EE, JMS MESSAGING, JDBC DRIVERS, ORACLE WEBLOGIC SERVER CLIENTS, AND APACHE WEB SERVER PLUGINS.
ORACLE WEBLOGIC SERVER, ENTERPRISE EDITION	ALL STANDARD EDITION FEATURES. ADDITIONALLY, INCLUDES WHOLE-SERVER MIGRATION AND SERVICE MIGRATION, AND ORACLE VIRTUAL ASSEMBLY BUILDER FOR RAPID PROVISIONING INTO VIRTUALIZED AND CLOUD ENVIRONMENTS.
ORACLE WEBLOGIC SUITE	ALL ENTERPRISE EDITION FEATURES. ADDITIONALLY, INCLUDES APPLICATION DATA MANAGEMENT PROVIDED BY ORACLE COHERENCE, ENTERPRISE EDITION, WHICH PROVIDES FAULT-TOLERANT DATA CACHING, DATA MANAGEMENT, WRITE-BEHIND, TRANSACTIONS, ANALYTICS, AND EVENTS. ALSO PROVIDES MANAGEMENT TOOLING FOR MONITORING, ADMINISTERING, AND PROVISIONING ORACLE COHERENCE.

Table 1

//enterprise java /

objects to encapsulate the data and business logic

- Remote method invocation (RMI), which provides the Java standard for distributed object computing
- A security API, which provides the integration of authentication and authorization services into your Java EE applications
- Oracle Coherence, which provides distributed caching and data grid capabilities

As a Java EE application server, one of Oracle WebLogic Server's main tasks is to control and coordinate resources and provide them to applications, normally via Java Naming and Directory Interface (JNDI). Administrators of Oracle WebLogic Server can configure, maintain, and control resources. The most-important resources include transactions, JDBC-based databases, and JMS messages. For a complete list of Java EE 6 features, see *The Java EE 6 Tutorial*.¹

Oracle WebLogic Server is available in the three editions shown in **Table 1**.

Overview of Using Oracle WebLogic Server with NetBeans and the Cloud

To give you a first impression of how these tools work, let's look at an example of

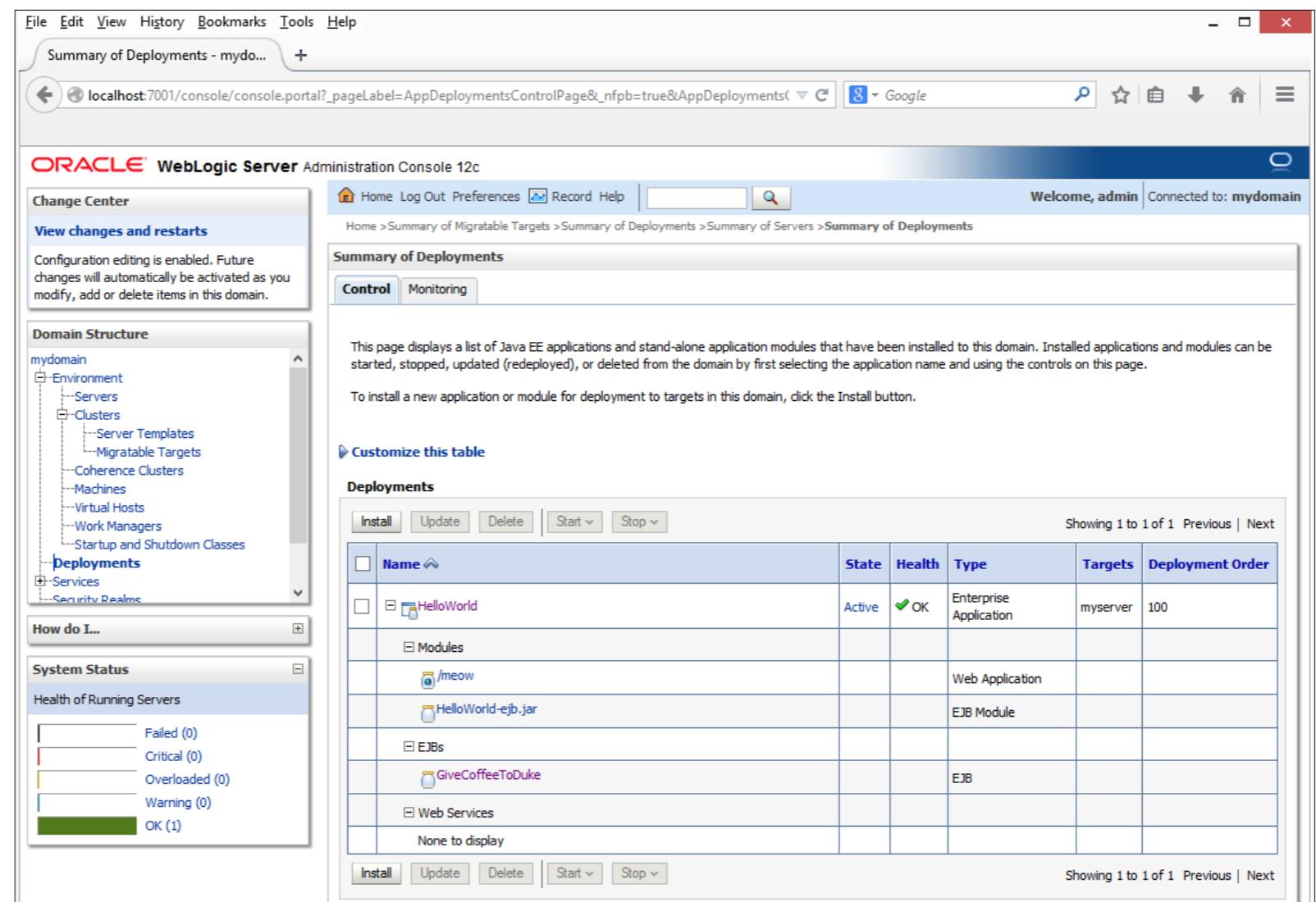


Figure 1

using Oracle WebLogic Server version 12.1.3.0.0.

First, download [Oracle WebLogic Server](#). Also download a recent version of [JDK 8](#) and [NetBeans IDE 8](#).

After you install Oracle WebLogic Server locally, its administration console is reachable at <http://localhost:7001/console>, as shown in **Figure 1**.

We can now configure NetBeans, with its Oracle WebLogic Server integration, in order to code, build, package, and deploy applications from inside NetBeans.

Figure 2 shows how to manage Oracle WebLogic Server from inside NetBeans.

And our mission-critical application—which was built,

deployed, and runs on Oracle WebLogic Server—is started automatically, as shown in **Figure 3**.

You can even deploy to and run on the cloud (see **Figure 4**). With [Oracle Cloud](#) hosting Oracle WebLogic Server, you can benefit from a decentralized, fully managed Oracle WebLogic Server cluster, including thorough monitoring capabilities.

¹ ORACLE WEBLOGIC SERVER 12c SUPPORTS JAVA EE 6. FOR THE MORE RECENT EDITION FOR JAVA EE 7, SEE <http://docs.oracle.com/javaee/7/tutorial/doc>.

//enterprise java /

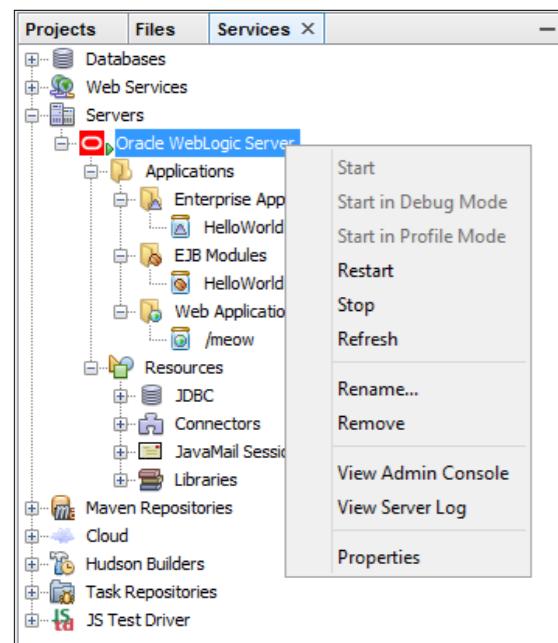


Figure 2

Now that we have quickly explored these tools, we are ready to dive into the building blocks of Oracle WebLogic Server.

Oracle WebLogic Server Components

A *domain* is the basic administration unit for Oracle WebLogic Server. This administration unit consists of one or more Oracle WebLogic Server instances and is managed by one instance that has a special role—the so-called Administration Server (AdminServer for short).

The AdminServer hosts the admin console and manages zero or more Oracle WebLogic Server instances called *managed servers*, which can be hosted locally to the AdminServer or hosted remotely

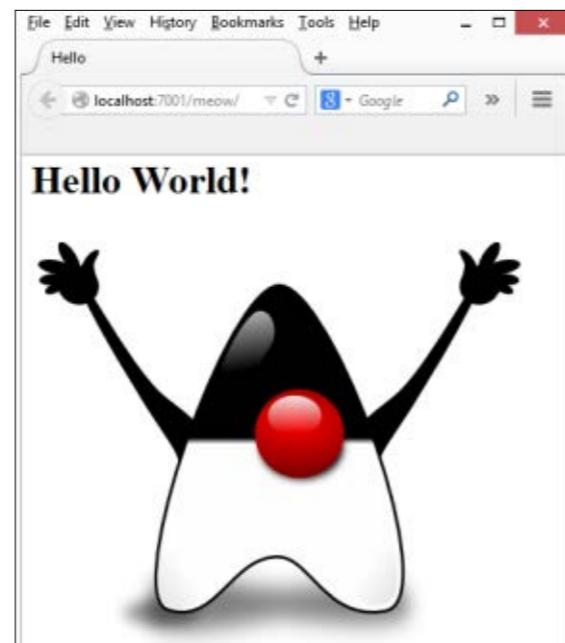


Figure 3

on different physical machines. Managed servers can be grouped into clusters. If managed servers are located on remote machines, separate NodeManager instances can be used for lifecycle operations, such as starting and stopping managed servers.

The central configuration file is called `config.xml`. This file is stored on the AdminServer with other files, such as database configuration files and security files. Each AdminServer manages exactly one domain, and all changes or activities apply only to this domain. Every domain can have multiple clusters. Every cluster can—but does not need to—include managed servers hosted on different machines.

In a nutshell, we have the Oracle WebLogic Server components

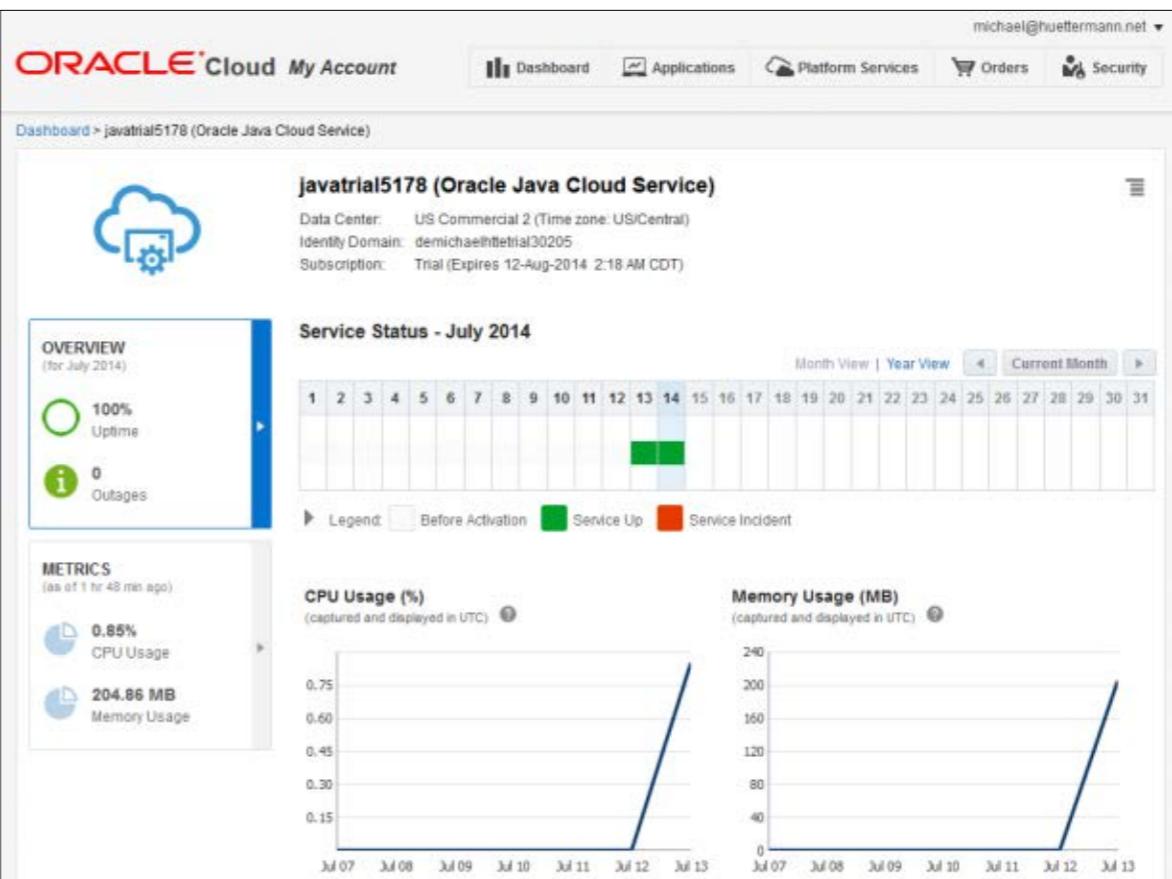


Figure 4

COMPONENT	DESCRIPTION
ADMIN SERVER	A DOMAIN INCLUDES ONE ORACLE WEBLOGIC SERVER INSTANCE THAT IS CONFIGURED AS THE ADMIN SERVER. ALL CHANGES TO THE CONFIGURATION AND DEPLOYMENT OF APPLICATIONS ARE DONE THROUGH THE ADMIN SERVER.
MANAGED SERVERS	ALL OTHER ORACLE WEBLOGIC SERVER INSTANCES IN A DOMAIN ARE CALLED MANAGED SERVERS. MANAGED SERVERS HOST APPLICATION COMPONENTS AND RESOURCES, WHICH ARE ALSO DEPLOYED AND MANAGED AS PART OF THE DOMAIN.
CLUSTERS	A DOMAIN CAN ALSO INCLUDE ORACLE WEBLOGIC SERVER CLUSTERS, WHICH ARE GROUPS OF MANAGED SERVER INSTANCES THAT WORK TOGETHER TO PROVIDE SCALABILITY AND HIGH AVAILABILITY FOR APPLICATIONS. CLUSTERS CAN IMPROVE PERFORMANCE AND PROVIDE FAILOVER WHEN A SERVER INSTANCE BECOMES UNAVAILABLE.

Table 2

shown in **Table 2**.

In production environments, it is highly recommended that

you deploy applications only on managed servers and you reserve the AdminServer for per-

forming management tasks. For more information about these basic components, see **Figure 5**, which shows example content of an Oracle WebLogic Server domain, and “[WebLogic Server Domains](#).”

Let’s now discuss [clustered Oracle WebLogic Server instances](#) and how to get your applications ready for enterprise usage.

Using Clustered Instances of Oracle WebLogic Server

Clustering applications makes them enterprise-ready. With clustering, incoming requests can be routed to an Oracle WebLogic Server instance in the cluster based on the volume of work being processed. In case of hardware failure or other failures, session-state information is available to other cluster nodes that can resume the work of the failed node. In addition, you can implement clusters so that services can be hosted on a single machine with the option to migrate the services to another node in the event of failure.

A clustered application or application component is one that is available on multiple Oracle WebLogic Server instances in a cluster. The following types of objects can be clustered in an Oracle WebLogic Server deployment. If an object is clustered,

failover and load balancing are available for that object.

- Servlets
- JSPs
- EJBs
- RMI objects
- JMS destinations
- Oracle Coherence clusters and managed Oracle Coherence servers
- Timer services

Clustered Oracle WebLogic Server instances behave similarly to nonclustered instances, except that they provide failover and load balancing. The process and tools used to configure clustered Oracle WebLogic Server instances are the same as those used to configure nonclustered instances.

Clustering provides enterprise-level benefits, above all the following:

- Scalability. The capacity of an application deployed on an Oracle WebLogic Server cluster can be increased dynamically to meet demand. You can add Oracle WebLogic Server instances to a cluster without interrupting the application’s services.

Clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your applications.

- High availability (HA). In an Oracle WebLogic Server clus-

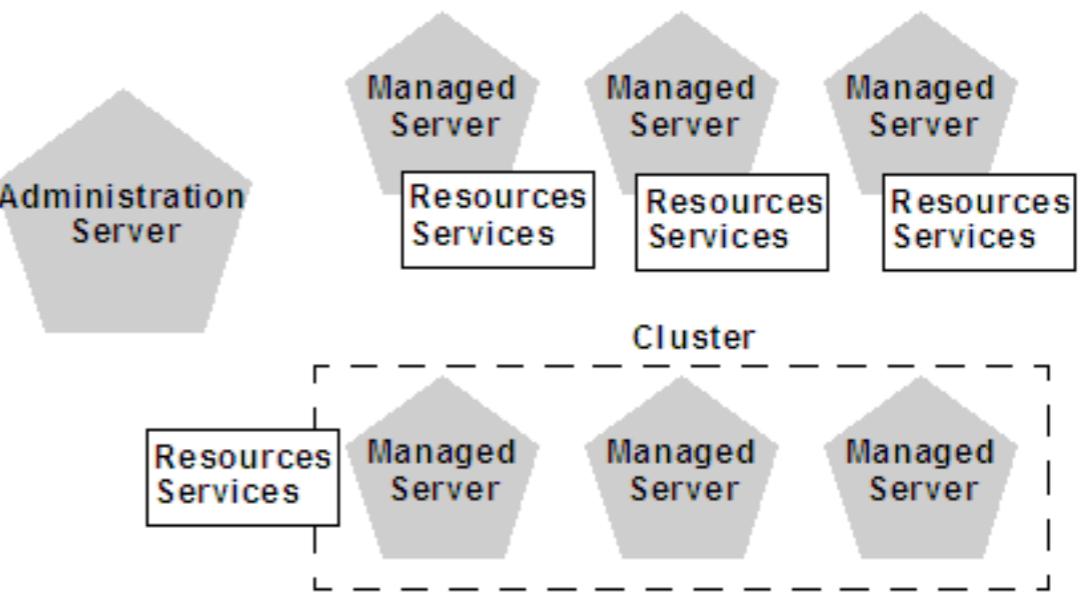


Figure 5

ter, application processing can continue when a server instance fails. You “cluster” application components by deploying them on multiple server instances in the cluster—so, if a server instance on which a component is running fails, another server instance on which that component is deployed can continue application processing.

Key clustering capabilities that enable scalability and HA include the following:

- Application failover. *Failover* means that when an application component doing a particular task becomes unavailable for any reason, a copy of the failed object finishes the task.
- Migration. Oracle WebLogic Server supports automatic and manual migration of a clus-

tered server instance from one machine to another. A managed server that can be migrated is referred to as a *migratable server*. This feature is designed for environments with HA requirements.

- Load balancing. *Load balancing* is the even distribution of jobs and associated communication across the computing and networking resources in your environment.

An example of how to achieve HA and scalability is by using a JDBC GridLink datasource in your application with Oracle WebLogic Server, which we discuss in the next section.

Achieving Persistence

With Oracle WebLogic Server, you have a couple of [options for connecting to a DBMS](#). There are two

main implementations of Oracle WebLogic Server for supporting data persistence in general and for supporting [Oracle Real Application Clusters \(Oracle RAC\)](#) in particular.

Using multi datasources.

Traditionally, multi datasources have been used. A *multi datasource* is a datasource abstraction over one or more individual datasources. It serves JDBC connections from each of the member datasources according to a specified policy. An Oracle RAC multi datasource configuration requires that each member datasource obtain connections to a particular Oracle RAC instance. The round-robin load balancing employed by the multi datasource implementation distributes work evenly across all member datasources.

At first glance, this is great for HA and load balancing, but it also has some limitations. First, the configuration is a bit complex because you need to configure several JDBC modules and a separate multi datasource for each defined service.

The main disadvantage, though, is that the configuration is static. Administration requires intervention to add or remove datasources when changes are made to the underlying database cluster topology. If you use multi datasources, you can lower the pain by applying some common recipes. One

example is to configure the initial capacity by setting the value of the JDBCConnectionPoolParamsBean .InitialCapacity managed bean (MBean) attribute to 0. If you configure a value greater than 0, Oracle WebLogic Server will not start when the database node cannot be connected.

There are also some common pitfalls. For example, don't forget to configure Oracle WebLogic Server to test connections that are on reserve (by using the JDBCConnectionPoolParamsBean .TestConnectionsOnReserve MBean). Otherwise, failover will not work.

In summary, all this is much too static to be a state-of-the-art configuration, in addition to being too cumbersome and too error-prone. But there is an alternative: using Active GridLink for Oracle RAC, which provides a GridLink datasource in Oracle WebLogic Server.

Using GridLink datasources. Using GridLink datasources, you can benefit from fast connection failover to achieve rapid failure detection; abort and remove invalid connections from the connection pool; and adapt to changes in topology such as adding or removing a node.

Runtime connection load balancing allows Oracle WebLogic Server to do the following:

- Adjust the distribution of work

based on back-end node capacities such as availability and response time

- Dynamically react to changes in the Oracle RAC topology; **Figure 6** shows how the Oracle Notification System message transport mechanism uses the

- publish-subscribe model
 - Manage pooled connections for high performance and scalability; **Figure 7** shows weighted or round-robin load balancing
 - Another great capability of Active GridLink for Oracle RAC is its XA affinity, a feature that ensures that

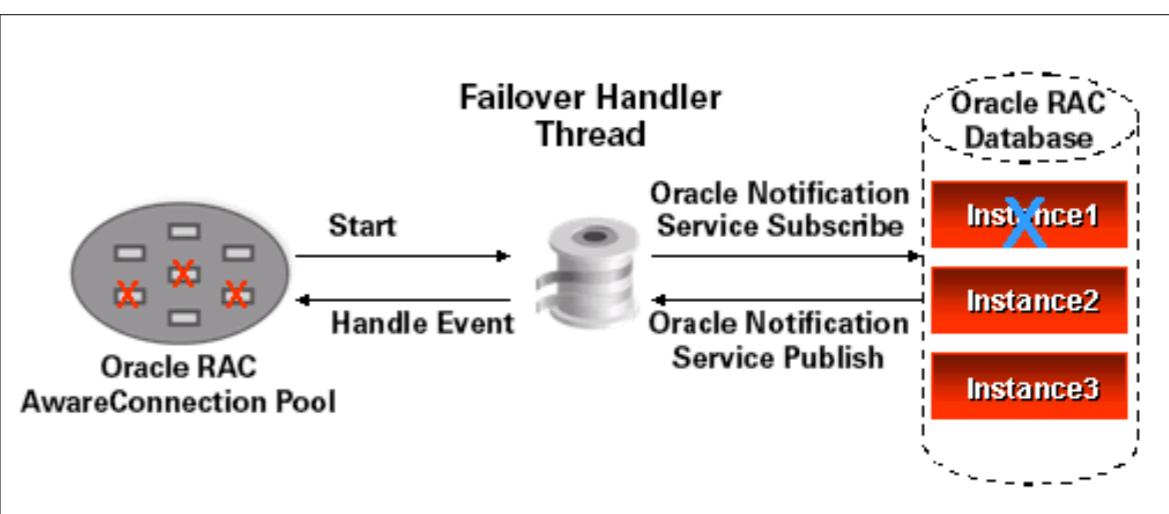


Figure 6

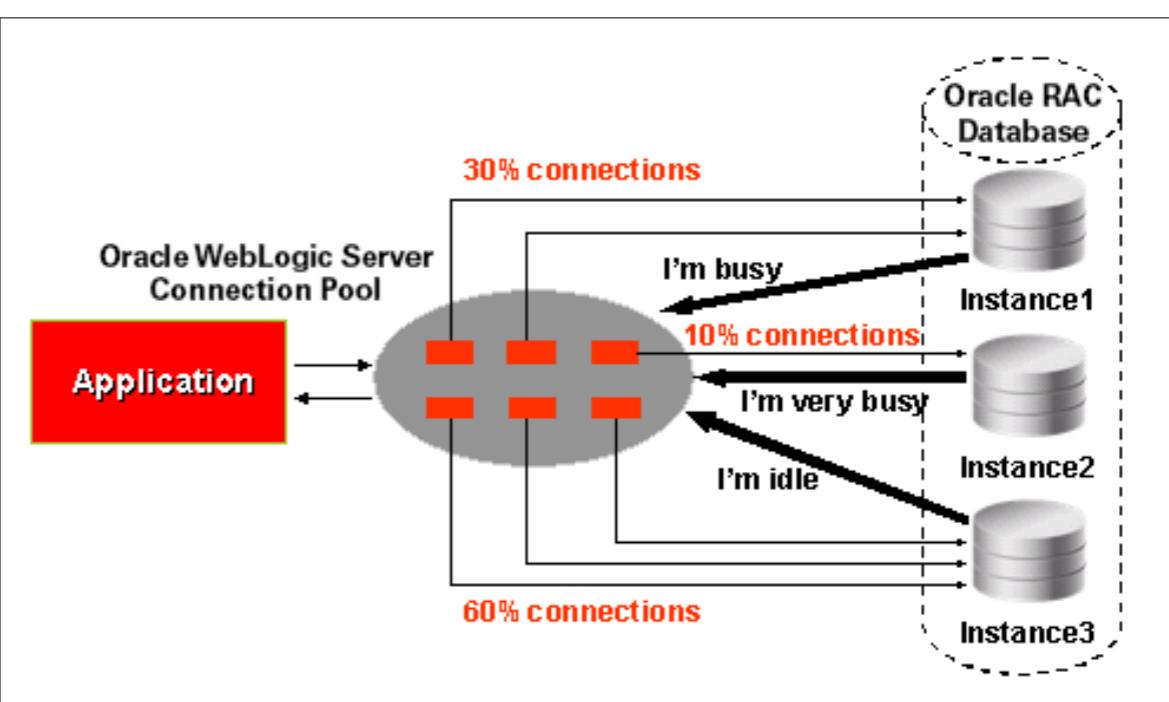


Figure 7



all database operations performed on an Oracle RAC cluster within the context of a global transaction are directed to the same Oracle RAC instance. For XA drivers, the system automatically selects the two-phase commit protocol for global transaction processing. Load is distributed to Oracle RAC database instances dynamically.

Configuration of GridLink data-sources is easy and straightforward. Just click through the fields that Oracle WebLogic Server wants you to fill out and provide information for the service name, host, port, database username, and password. The console automatically generates the complete JDBC URL for you, for example:

```
jdbc:oracle:thin:@
(DESCRIPTION=(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=left)(PORT=1234))
(ADDRESS=(PROTOCOL=TCP)
(HOST=right)(PORT=1234))
(ADDRESS=(PROTOCOL=TCP)
(HOST=center)(PORT=1234)))
(CONNECT_DATA=(SERVICE_
NAME=myService)))
```

You can also enter the complete URL directly.

Now that we've explored data-sources, we can explore another neat feature of Oracle WebLogic Server: WebLogic Scripting Tool.

WebLogic Scripting Tool

[WebLogic Scripting Tool](#) is a toolkit that administrators and operators can use to monitor and manage Oracle WebLogic Server domains. It is based on the Java scripting interpreter Jython. WebLogic Scripting Tool offers scripting features for Oracle WebLogic Server, but because it is based on Jython, which is the Java implementation of Python and runs on the Java Virtual Machine (JVM), it is also possible to use all the common features of the Jython language, such as local variables, conditions, or flow statements. Different execution models are available: scripting, interactive, and embedded. WebLogic Scripting Tool can be enabled for online and offline connection modes, and it can act as a Java Management Extensions (JMX) client.

Jython supports object-oriented programming (OOP), with features such as modularity, structure, and encapsulation. Coding with Jython is more like shell scripting but the structure is a well-defined, strongly typed language. Jython requires proper indentation of the code in order for the code to be interpreted successfully.

It is pretty easy to learn WebLogic Scripting Tool scripting because the tool is based on Jython, and on top of the language, it only adds support for the different Oracle

WebLogic Server MBeans and the navigation among them.

Jython is complementary to Java and is particularly useful for embedded scripting. This means that you can easily add Jython libraries and Jython scripts into your application. Also quick prototyping is possible by using the interactive interpreter that can be used to interact with Java packages or with running Java applications.

WebLogic Scripting Tool program files must have the extension [.py](#) and contain source code consisting of a set of Jython instructions for an Oracle WebLogic Server domain. This source code uses a JVM when it is instructed to run the script, and its invocation internally generates Java bytecode.

WebLogic Scripting Tool offers an interpreter mode in which you can enter commands interactively. WebLogic Scripting Tool does not support a history (for example, using arrow keys as in a UNIX bash history). This means you always have to retype commands you want to repeat. The good news is that the vital Java open source community

DEVOPS TOOL

You can also configure Oracle WebLogic Server to monitor and tune application throughput automatically, so it is **an excellent vehicle to underpin your DevOps approach.**

offers a [helpful project](#) that can be used to add history functionality to WebLogic Scripting Tool.

The Oracle WebLogic Server distribution provides some convenience scripts to help you set up your environment, for example, the paths. Once set up, you can, for example, connect to a running Oracle WebLogic Server instance (in online mode) to get a thread dump (see [Listing 1](#)). A more encapsulated convenience script to generate a thread dump could look like [Listing 2](#).

Please note that the code in [Listing 2](#) is an executable Python file that is called by passing in properties and a connection URL. In its core, the script wraps the WebLogic Scripting Tool [connect](#) command, triggers the [threadDump\(\)](#) command, and offers convenience features around them. After connecting to the server and getting the thread dump, we again disconnect from the server. Such scripts can help in automating the deployment and management of your enterprise applications and middleware.

Crisp, isn't it?

```
>java weblogic.WLST
```

Initializing WebLogic Scripting Tool (WLST) ...
 Welcome to WebLogic Server Administration Scripting Shell
 Type help() for help on available commands

```
wls:/offline>help('connect')
wls:/offline>connect('admin', 'pw', 't3://localhost:7001')
```

Connecting to t3://localhost:7001 with userid admin ...
 Successfully connected to Admin Server "myserver" that belongs
 to domain "mydomain".
 wls:/mydomain/serverConfig>
 wls:/mydomain/serverConfig>threadDump()

 [Download all listings in this issue as text](#)

Conclusion

In this article, you've seen an overview of Oracle WebLogic Server, and gotten some first impressions about its GUIs, as well as its integration with NetBeans and the cloud. We've talked about Oracle WebLogic Server components—above all domains and clusters—and about datasources, including examples of how to implement HA and load balancing in your applications. Finally,

we covered WebLogic Scripting Tool, which is based on Jython and helps you automate your Oracle WebLogic Server activities.

I hope you've seen what Oracle WebLogic Server can offer you. Now have fun trying out this great application server yourself! <[/article](#)>

LEARN MORE

- [Oracle WebLogic Server documentation](#)



Find the Most Qualified Java Professionals for Your Company's Future

Introducing the *Java Magazine* Career Opportunities section – the ultimate technology recruitment resource.

Place your advertisement and gain immediate access to our audience of top IT professionals worldwide including: corporate and independent developers, IT managers, architects and product managers.



For more information or to place your recruitment ad or listing contact:
tom.cometa@oracle.com



JOHAN VOS



Location-Based Software Using JavaFX

Discover the benefits of using JavaFX for visualization software.

The concept of location is becoming more and more important in a number of services. Combining content with location-based information enriches the content. With a growing number of location-aware devices, there is an increasing amount of data that can be associated with a location.

The JavaFX platform provides a number of features that allow Java developers to visualize location-based data in an accurate, fancy, and performant way. In this article, we'll explore some of the benefits the JavaFX platform provides when you are developing visualization software for location-based services.

The [OpenMapFX](#) project is an open source, community-driven project that provides a number of location-based services and tools. The core

of the OpenMapFX project is a renderer that displays a geo-location map on top of which services can be plugged in. The examples we develop in this article, which can be downloaded [here](#), use the same techniques as the OpenMapFX code.

Rendering Tiles

Geo-location information is often rendered using tiles. A *tile* is a small image (typically 256 by 256 pixels) containing a segment of the world map at a given zoom level. A number of map providers (such as Google Maps, OpenStreetMap, and MapQuest) provide their map data using an HTTP service that takes a simple parameterized request and returns the requested tile as an image.

In this article, we see

data obtained from [OpenStreetMap](#), which is a community-driven project that creates and distributes free geographic data for the world. Other tile providers have a very similar interface, and the OpenMapFX project shows that you can easily switch between different providers.

A tile provider offers its services using a tile engine that renders tiles based on three request parameters:

- The zoom level (*z*)
- The horizontal index of the tile (*i*)
- The vertical index of the tile (*j*)

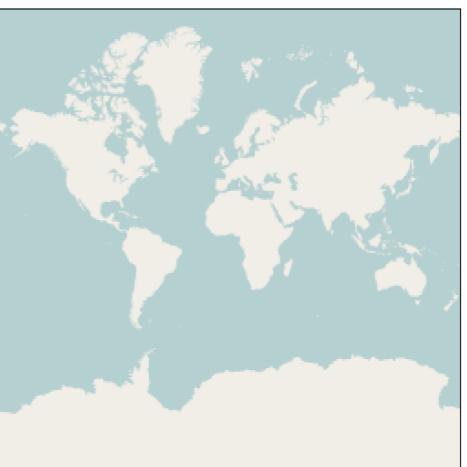
At the lowest zoom level, 0, the whole world map is rendered in a single tile.

Figure 1 shows the only tile at zoom level 0 returned by the OpenStreetMap tile provider. This image can be obtained

by pointing a browser to <http://tile.openstreetmap.org/0/0/0.png>.

This URL, and all the URLs for obtaining tiles, is constructed as follows: <http://base-url/z/i/j.png>, where *base-url* is the URL for the specific tile provider, and *z*, *i*, and *j* are the three parameters described earlier.

Each incremental zoom level contains four times the number of tiles as the previous zoom level. For each tile

**Figure 1**

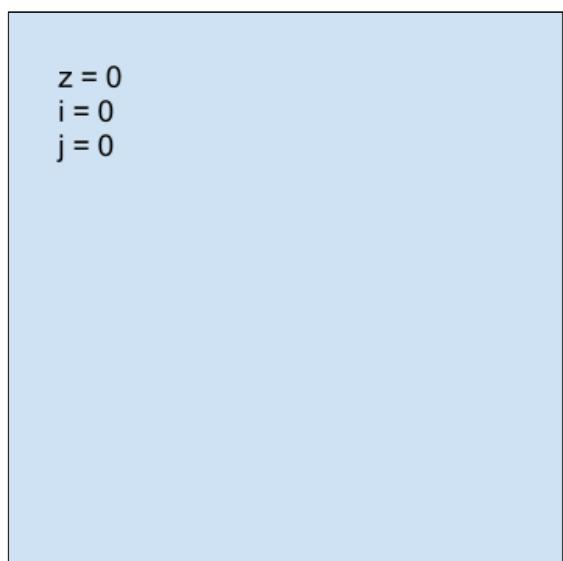


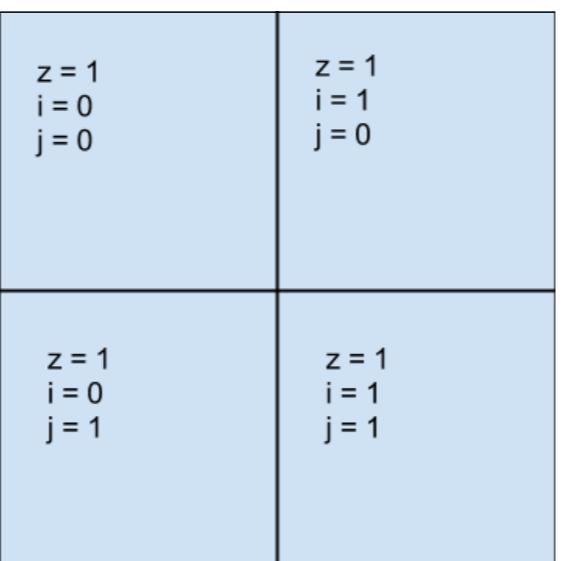
Figure 2



Figure 3

at zoom level z , there exist four tiles at zoom level $z + 1$ that cover the same area in more detail. **Figure 2** shows how the single tile at zoom level 1 corresponds to the four tiles at zoom level 2.

Rendering a single tile in JavaFX is very straightforward. We create



an [Image](#) object and pass the URL to the constructor. Next, we wrap the [Image](#) object in an [ImageView](#) instance, which can be rendered in the scene. The code in **Listing 1** shows how to do this.

The result of running the code shown in **Listing 1** is a 256 by 256 image that shows a static world map (see **Figure 3**).

Dragging and Scaling a Map

As a first enhancement, we will make our map draggable.

The JavaFX platform provides a number of features to deal with drag events and, by extension, with drag-and-drop events. We will take a simple approach here and listen for [mousePressed](#) and [mouseDragged](#) events and move the [imageView](#) accordingly. The code in **Listing 2** shows how to do this.

[LISTING 1](#) [LISTING 2](#) [LISTING 3](#)

```
@Override
public void start(Stage primaryStage) {
    Image image = new Image(
        "http://tile.openstreetmap.org/0/0/0.png");
    ImageView imageView = new ImageView(image);
    StackPane root = new StackPane();
    root.getChildren().add(imageView);
    Scene scene = new Scene(root, 300, 300);
    primaryStage.setTitle("TileMap 1");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```

[Download all listings in this issue as text](#)

We define the [onMousePressed](#) and [onMouseDragged](#) handlers on the root of the scene. In the [onMouseDragged](#) handler, we calculate the distance and change the [translate](#) property of the [imageView](#) node.

The map is now draggable, and we can move it inside the application window. As a next step, we want to be able to scale the map to make it bigger. Depending on the platform, different approaches for scaling are available. On a system with a mouse that has a scrollwheel, moving the scrollwheel often indicates the desire to zoom in or out. In JavaFX, this behavior can easily be detected, and an appropriate handler can be used

to scale the image. **Listing 3**, which is taken from the TileMap3 source code sample, shows how to do this.

In **Listing 3**, we introduce a [DoubleProperty](#) named [scaleProperty](#), which holds the scale factor of our map. We use the JavaFX binding concept to bind the scale of the node containing the image to the value of the [scaleProperty](#). Rather than have the event handlers directly change the scale of the node, we let the event handlers change the [scaleProperty](#). One advantage of using this approach is that we can have a number of event handlers—or pieces of code, in general—that modify the [scaleProperty](#) and, hence, cause the scale of the node to change.



//rich client /

As shown in **Listing 4**, the `scaleProperty` itself is modified when the user uses the scrollwheel to zoom in or zoom out.

On systems that support zoom events (for example, a system that has a multizoom trackpad), we can detect zoom events.

Listing 4 shows how we can modify the `scaleProperty` when zoom events are detected.

With the current code, we can render a map of the world, drag the map around a window, and scale the image. However, simply scaling an image does not add more detail. Typically, when you zoom into a map, you want to see a more-detailed view of the map. In order to achieve this, we need to use more-detailed tiles to replace parts of the tiles at a lower zoom level.

Using the MapTile Class

At this point, we will add a `MapTile` class to our application for holding information about the tiles. At the very least, a tile has the following properties:

- The zoom level (`tileZoom`)
- The horizontal index of the tile (`i`)
- The vertical index of the tile (`j`)
- The image

This information is stored in the `MapTile` class, which extends the JavaFX `Region` class. The image that has to be shown is added to the children of the `Region`. The con-

structor of the `MapTile` contains the code shown in **Listing 5**.

Note that we use the two-argument constructor of the `Image` class, which allows us to specify that we want to load the image using a background thread. This is very useful in applications that require the retrieval of a number of images from a remote source and that don't want to block the JavaFX Application thread. If you omit the `true` parameter in the constructor, all images will be loaded one by one on the JavaFX Application thread, which would be very bad for the user experience, because users won't be able to drag or scroll while tiles are still being loaded.

We have to provide support for changing zoom factors in the `MapTile`. The scale at which a tile should be rendered depends on the zoom level of the tile and the scale factor for the whole map. The zoom level is a property of a tile (called `tileZoom`), and it does not change. The scale factor of the map can vary with input actions (for example, scrolling or using a multitouch zoom gesture).

If the zoom level of a tile equals the current scale factor, we don't have to scale that specific tile. When the scale factor becomes larger than the tile's zoom level, we have to scale up the tile. Conversely, a scale factor lower than the zoom

LISTING 4

LISTING 5 / **LISTING 6** / **LISTING 7**

```
root.setOnZoom(t -> scaleProperty.set(
    scaleProperty.get() + t.getZoomFactor() > 1 ? .1 : -.1));
```

[Download all listings in this issue as text](#)

level requires us to scale down the tile.

In general, the scale that has to be applied to a tile can be calculated as follows:

■ $2^{(scaleFactor - tileZoom)}$

The JavaFX platform allows transformations to be applied to nodes. A `Scale` is a specific transformation that allows horizontal and vertical scaling of the node, and it is well suited for this use case. The following snippet creates a `Scale` transformation and applies it to the `MapTile`:

```
Scale scale = new Scale();
scale.setPivotX(0);
scale.setPivotY(0);
getTransforms().add(scale);
```

Note that we set the pivot point explicitly to the top-left origin of

the node (the point with coordinates 0,0). This causes the scaling to be centered around this origin, rather than in the center of the node. As a consequence, the top-left corner of the node will always be at the same position, which is important for further calculations.

When the `scaleProperty` of the map changes, we have to change the value of the `Scale` transformation. This is done by adding an `InvalidationListener` to the `scaleProperty` of the map, as shown in **Listing 6**. The `calculatePosition` method shown in **Listing 7** will apply the correct scale factor to the `Scale` instance.

Setting the `x` and `y` property of the `Scale` instance will cause the `MapTile` node to be scaled up or down. When the `scaleFactor` is equal to the `tileZoom`, the scale value (`sv`) will be 1 and no stretching or shrinking will be done.



//rich client /

Apart from the scaling, we also have to make sure the tiles are positioned at the correct location. If the `scaleFactor` is equal to the `tileZoom`, all tiles should be 256 pixels apart from each other in both the horizontal and the vertical direction. If the `scaleFactor` is larger, the tiles should be farther apart, and if the `scaleFactor` is smaller, the tiles should be closer to each other.

Thanks to the `calculatePosition` method, every `MapTile` that is added to the scene graph will always be positioned and scaled appropriately. The calculations for the `scale` and `translate` properties are valid regardless of whether the tile is inside the visible window.

Determining Which Tiles Should Be Created

We now have a generic `MapTile` class that can be used to load and position tiles from different zoom levels onto the map, but we still need to create instances of the class when needed. We will use an array of `Map` entries for storing the tiles, as shown in [Listing 8](#).

Note that the `MAX_ZOOM` value can take

RICHER CONTENT
Combining content with location-based information enriches the content. With a growing number of location-aware devices, there is an increasing amount of data that can be associated with a location.

any value, but a zoom level of 20 is already very accurate, and it is the limit provided by OpenStreetMap.

A naive approach would be to create all the tiles and add them to the scene graph. Indeed, thanks to the previous calculations, tiles that are outside the visible window will not disturb the visible tiles, because they are rendered at positions that are not inside the visible window.

At zoom level 1, there is a single tile. At zoom level 2, we have four tiles, and at zoom level z , we have $2^{(2z)}$ tiles. Zoom level 10 leads to more than a million tiles, and zoom level 10 is not even very detailed.

It would be a bad idea to have all those tiles in memory. Rather,

we need to detect which tiles should be rendered at a given moment. The algorithm shown in [Listing 9](#) is better than the completely naive approach, but it is still not optimal and will cause memory or performance problems because tiles are never evicted. OpenMapFX contains an improved algorithm, and I strongly encourage interested readers to inspect the code for this project.

[LISTING 8](#) [LISTING 9](#) [LISTING 10](#) [LISTING 11](#) [LISTING 12](#)

```
final int MAX_ZOOM = 20;
private final Map<Long, MapTile4>[] tiles =
new HashMap[MAX_ZOOM];
```

[Download all listings in this issue as text](#)

There are two actions that can require the creation of new tiles:

- The map is dragged, and an area with unloaded tiles becomes visible.
- The user zooms in or out on the map, and the new `scaleFactor` now matches a different `zoomLevel`.

In both cases, an algorithm has to be applied that will detect which tiles need to be created and added to the scene graph. The algorithm in [Listing 9](#) does exactly that. It first determines the desired zoom level, which is the truncation of the double `scaleFactor` to its lower `int` value.

The theoretical highest `i` and `j` index for a tile is easily obtained as follows:

```
long max_i = 1 << lowScale;
long max_j = 1 << lowScale;
```

We now try to detect the absolute lowest `i` index for a tile that would at least partially fit on a screen. With a little mathemat-

ics, we define these lower bounds as shown in [Listing 10](#), where `tx` and `ty` are the current values of `tileGroup.getTranslateX()` and `tileGroup.getTranslateY()`, respectively.

Similarly, we have to detect the highest `i` index for a tile that would still fit at least partially on a screen. Again, some mathematics lead to the values shown in [Listing 11](#).

We now loop over all the `i` values between `imin` and `imax`, and a second internal loop ranges over the `j` values between `jmin` and `jmax`.

Based on the `i` and `j` indexes, each tile can be assigned a unique key:

```
long key = i * max_i + j;
```

This key is unique for a tile at a specific zoom level.

For this unique key, we check whether we already have the specific tile at the specific zoom level in the map in the array and, if not, we create the tile, add it to the scene graph, and put it in the map, as shown in [Listing 12](#).

//rich client /

Running the code in the TileMap4.java source code, and playing a bit with zoom and drag movements, results in the image shown in **Figure 4**.

Possible Improvements

While the code we created here shows a map and allows us to drag and scroll the map, it is clearly not ready for production. I encourage you to look at the OpenMapFX project for more-robust code, services, and samples, and for ports to mobile devices.

A number of obvious improvements that are implemented in OpenMapFX are listed below:

- Tiles that are not visible on the screen are not required to be in the scene graph, and they may be reclaimed by the garbage collector if we are running low on memory. This behavior is implemented using a [Map](#) with [SoftReference](#) instances.
- While a more-detailed tile is being loaded, a tile with less detail that is already loaded should be rendered (and scaled accordingly).
- When the user zooms into the map, the clickpoint stays at the same location before and after the zoom is applied.
- It is possible to switch between different tile providers on the fly.

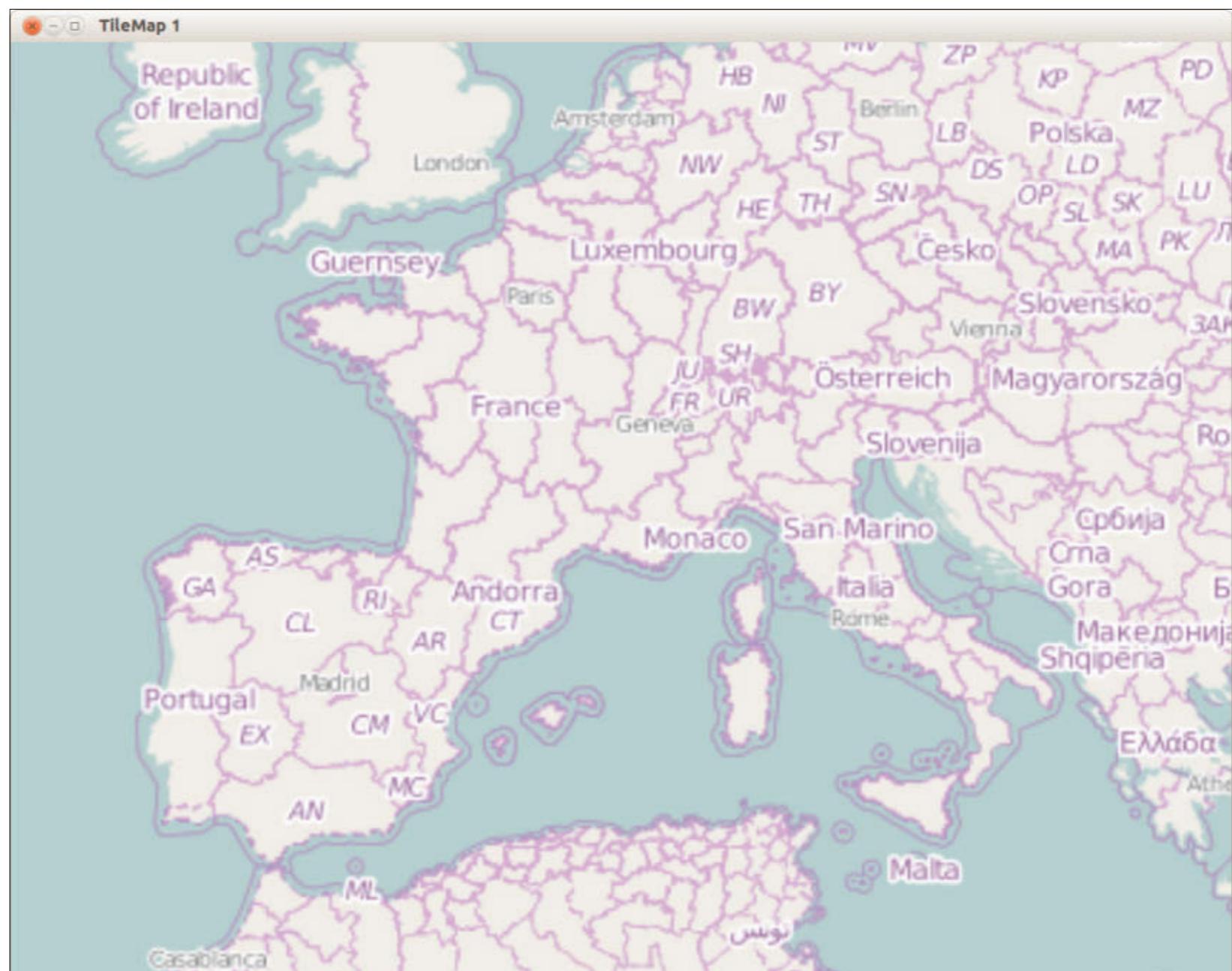


Figure 4

Conclusion

With a small amount of code, we created a very basic map renderer in JavaFX. The code is kept small thanks to a number of features available in the JavaFX platform:

- Background loading of images

- Property binding
- Lambda expressions
- Transformations (for example, [Scale](#))
- Ability to specify all coordinates relative to their parents (thanks to the scene graph concepts)
- Easy drag and zoom detection

LEARN MORE

- [OpenMapFX project](#)
- [OpenStreetMap](#)



Getting Started with Home Automation

Leverage openHAB and the huge Java ecosystem to create a truly smart home.



KAI KREUZER AND
THOMAS EICHSTÄDT-
ENGELLEN

BIO

Smart homes have promised for more decade, and with the of smartphones, they finally within reach for the masses. Besides the classic home automation projects, such as lighting and HVAC, there are more and more Wi-Fi-enabled gadgets on the market that target specific use cases, such as plant sensors and smoke detectors.

For homeowners who want to get started with home automation, this very fragmented market raises more questions than answers. Not only are there a large

WORK TOGETHER

A smart home cannot be built with a single system or technology, especially if you want to avoid vendor lock-in. So the only solution is to integrate different systems and devices to make them work together.

number of vendors, but there are also many competing technologies. In addition, smart homes require a very distributed setup with sensors and actors in various places around the house, so a lot of constrained, embedded hardware is used. For example, a small window contact sensor does not come with lots of CPU power.

Connectivity and security are challenges as well: Wireless communication is perfect for retrofitting, but it is generally vulnerable to attacks unless a secure protocol is used

(which usually requires strong CPU power). The easiest connection method is to use a dedicated data cable, which is done in bus systems such as KNX. Another possibility is to use existing power cables, called *power-line communication* (PLC). All options have their pros and cons, and the best solution highly depends on each situation.

Integration Through openHAB

Consequently, a smart home cannot be built with a single system or technology, especially if you want to avoid vendor lock-in. So the only solution is to integrate different systems and devices to make them work together. The [open Home Automation Bus \(openHAB\)](#) focuses exactly on these integration needs. It can connect to devices

using more than 90 different technologies or protocols and allows overarching automation rules and uniform user interfaces across all of these. It is a software stack for home gateways that is fully written in Java and, thus, can be installed on a great variety of platforms, thanks to the Java Virtual Machine (JVM), which allows the same code to run on different CPUs and system architectures.

openHAB is open source and vendor-neutral and has a huge community and a growing list of supported technologies. This community benefits from the huge Java ecosystem, in which libraries are available for almost anything you might need. Developers can, thus, build on the work of others to quickly achieve professional results.





Example openHAB Setup

In the following sections, you will see how to set up openHAB in order to take your first steps toward a smart home. The example uses data from the Yahoo weather service and visualizes the forecasted temperature by changing the color of a Philips Hue LED bulb. This simple setup will give you a good idea of what can be achieved through openHAB and how you can realize your own ideas.

Installing the software. Before starting the installation, you have to choose a hardware platform for running openHAB. Since openHAB is implemented in 100 percent pure Java, a wide range of platforms is available. A platform can be as small as a low-budget Raspberry Pi, or it can range from existing Synology network-attached storage (NAS) to a full-blown server appliance.

Even the operating system does not matter. openHAB runs on UNIX derivates such as Oracle Solaris, Debian Linux, Mac OS X, and Microsoft Windows; the only requirement is a Java SE virtual machine. On ARM-based embedded devices, the JVM provided by Oracle Java SE Embedded is a perfect fit.

Once you have installed a JVM on the selected platform, download the [openHAB binaries](#). First, download the openHAB runtime.

To install it, simply unzip it into an appropriate application folder, such as <C:\openhab> or </opt/openhab>.

Next, download the add-ons archive and unzip it into a separate directory. Then copy the files relevant for your setup into the `addons` directory of openHAB. For our example setup, the relevant files are `org.openhab.binding.http-1.x.y.jar` (HTTP binding), `org.openhab.binding.hue-1.x.y.jar` (Philips Hue binding), and `org.openhab.persistence.rrd4j-1.x.y.jar` (RRD4j persistence service).

Then download and install the openHAB Designer (see **Figure 1**), which automatically highlights and checks the syntax of the relevant configuration files and helps identify the options you have (content proposals). Unzip the Designer file that is appropriate for your platform to a directory on your desktop machine, such as `C:\openhab-designer` or `/opt/openhab-designer`. Launch the Designer by executing the `openHAB-Designer.exe` file (or `openHAB-Designer` if you are on a Mac or Linux OS).

Once the Designer has started, select the **configurations** folder of your runtime installation. If you are running the

Designer on a different machine than your runtime, you might have to expose the [configurations](#) folder through a Samba share or a similar method.

Configuring openHAB. Before you start, get acquainted with the most-important configuration settings. The central user configuration file is [configurations/openhab.cfg](#). This file does not exist after a clean installation, so you have to create it by copying the file [configurations/openhab_default.cfg](#), which contains reasonable default values and descriptions of all options. Thus, the [openhab.cfg](#) file needs to contain only user-specific information.

A recommended setting to change is **security:option**. Setting it to **ON** causes authentica-

tion to be required for HTTP and HTTPS requests. Valid credentials can be configured in the file `users.cfg` using the syntax `<username>=<password>`

All other sections after the general settings in the `openhab_default.cfg` file are relevant only if you have installed the corresponding add-ons. Therefore, for our example setup, we have to check the settings of only the Philips Hue and HTTP bindings (no settings are necessary for the RRD4i persistence service).

For the integration of the Philips Hue binding, there are two parameters to set: `hue:ip` and `hue:secret`. The first is simply the IP address of the Philips Hue bridge through which the LED bulbs can be accessed. The secret is any arbitrary

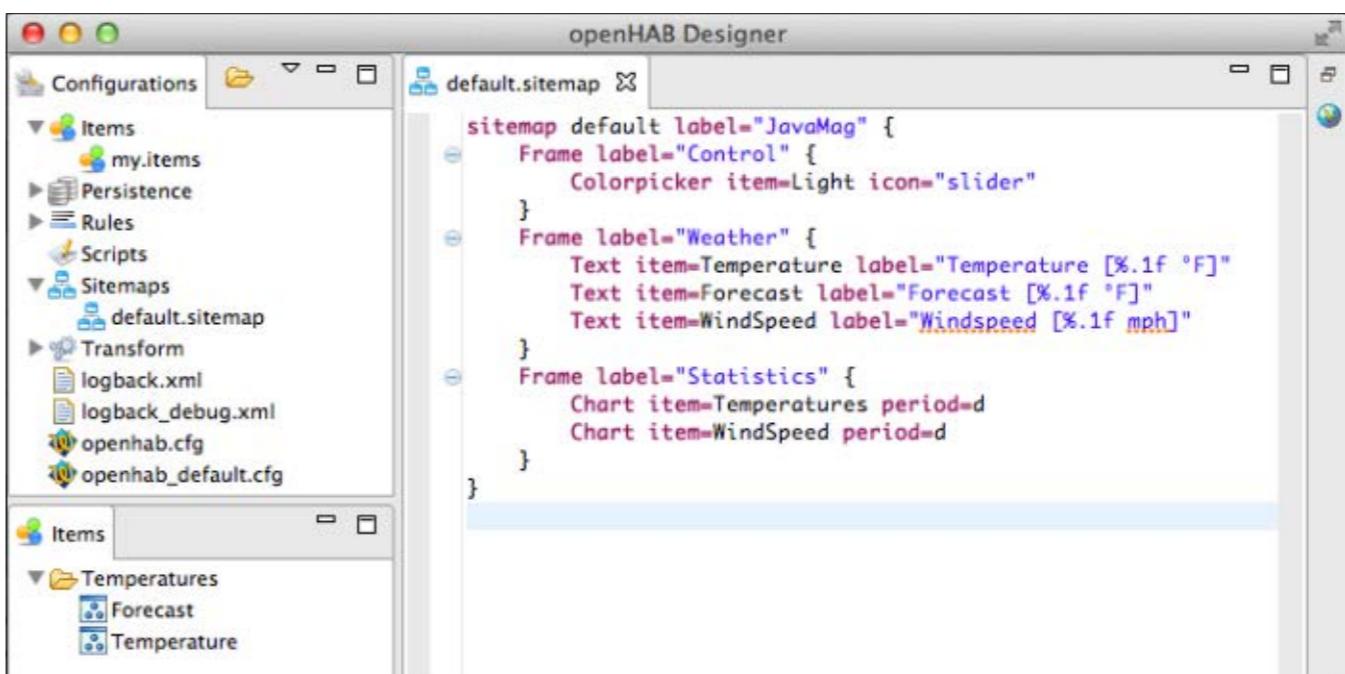


Figure 1

//embedded /

string that is used to identify and authorize your openHAB instance. Note that the string should be at least 10 characters long.

For the HTTP binding, we need to configure the `http:weather.url` property, which holds the URL for the location from which data should be retrieved. We will use Yahoo weather for San Francisco, so set the parameter to `http://weather.yahooapis.com/forecastrss?w=2487956&u=f`.

Additionally, set `http:weather.updateInterval` to `60000`, which means the URL is refreshed once every minute. **Listing 1** summarizes the content of our resulting `openhab.cfg` file.

Setting up a domain model. Once the general configuration is done, we can start designing the world inside openHAB.

Many home automation systems reflect the real devices in as detailed a manner as possible in the software. At first sight, this seems to be the most natural approach. However, every new device type has to be supported individually, which means it has to be implemented in the system before it can be used by the users. Sometimes, device hierarchies help mitigate this problem, but there are still compromises.

Rather than modeling a tumble dryer or a radio as a device, open-

HAB follows a capability-based approach. It splits devices into their functionalities, which are dubbed *items* in openHAB. This way, a radio can be modeled as three items: a switch representing the power, a dimmer for the volume, and a number specifying the station preset frequency.

Items can receive commands and hold a state. In order to keep the domain model compact, there are only a limited number of item types, for example, `String`, `Number`, `Dimmer`, `Contact`, `Rollershutter`, and `Color`.

Items are defined in files with the extension ".items" and they are stored in the folder `configurations/items`. First, create a new items file called `my.items` in this folder. Once the file is created, it can be edited by the Designer.

Let's now create an item for a light that is capable of changing color based on the temperature in San Francisco. We use the type `Color` and give the item the unique name `Light`. To model a temperature, we use the type `Number` and the name `Temperature`. Since openHAB cannot infer an appropriate icon from the generic `Number` type, we specify a default icon `<temperature>` (or `<wind>` for the wind speed). Using parentheses an item can be assigned to a group, which is itself defined as an

LISTING 1 LISTING 2

```
#####
##### General configurations #####
#####
security:option=ON

#####
##### HTTP Binding #####
# configuration of the first cache item
http:weather.url
  http://weather.yahooapis.com/forecastrss?w=2487956&u=f
http:weather.updateInterval=60000

#####
##### Philips Hue Binding #####
hue:ip=192.168.0.42
hue:secret=myPersonalSecret
```



[Download all listings in this issue as text](#)

item of type `Group`. The resulting `my.items` file is shown in **Listing 2**.

Next up is the user interface. Instead of offering a visual designer, openHAB follows a declarative UI approach. Users merely define the content to be shown on a certain page. The various openHAB UI implementations take care of an appropriate layout and the rendering themselves. These declarations are done in files called `sitemaps`. With a few lines, it is possible to define the structure and the content of the screens. Sitemap files are stored in the folder `configurations/sitemaps`.

Each sitemap has an ID and a label (for example, "JavaMag") and

consists of lists of user interface elements for the content. (A list of all available UI elements can be found in the [GitHub wiki](#).) Screens can be portioned using the `Frame` sections.

Our `Light` item can be represented using the `Colorpicker` element with a slider icon. And our `Temperature` item can be represented using a `Text` element with a label that contains additional formatting options for the item's state. These formatting options allow for specifying the decimal fraction or adding a unit description. **Figure 1** shows the resulting sitemap for these items.

Binding the domain model to the world. Now that we have created

//embedded /

our domain model, we need to bind the items to the real world. This is done by adding a binding configuration to the item definitions. For the Philips Hue bulb, `{ hue="1" }` is added to the `Color` item, where `1` specifies the number of the bulb in the Philips Hue system (which has multiple bulbs) that we want to control.

The HTTP binding uses a slightly more complex syntax, but in essence, we need to reference the URL that we set up in `openhab.cfg` and provide a suitable transformation function that extracts the piece of data that we require. This is done using XSLT stylesheets, which are readily prepared in the `configurations/transform` folder. The final `my.items` file is shown in **Listing 3**.

Storing and visualizing data. For all use cases where data needs to be permanently stored, openHAB comes with built-in persistence support, which is not restricted to a single datastore. Multiple



Figure 2

stores—such as relational databases, NoSQL databases, round-robin databases, Internet of Things (IoT) cloud services, and simple log files—can coexist and can be configured independently. Note that some options might be suitable only for exporting data (for example, IoT services or log files), while others can be queried as well, so they can be used for visualizing data through charts.

For every persistence service, a configuration file named `<persistenceservice>.persist` (for example, `rrd4j.persist`) is placed in the folder `configurations/persistence`. These configuration files provide a simple way to define different time-based or event-based persistence individually for items.

For our example, **Listing 4** shows how data for all items belonging to the group `Temperatures` is stored in a round-robin database (RRD) every minute. In addition to storing item states, the persistence service also restores their states at system startup.

Once the data is stored to a queryable persistence service, the built-in chart engine can generate charts to visualize the data. There is a dedicated `Chart` element that can be used within sitemaps. **Figure 2** shows the result of ren-

LISTING 3

LISTING 4 / LISTING 5

```
Color Light { hue="1" }
Number Temperature <temperature> (Temperatures) \
{ http="<[weather:60000:XSLT(yahoo_weather_temperature.xsl)]>" }
Number Forecast <temperature> (Temperatures) \
{ http="<[weather:60000:XSLT(yahoo_weather_forecast_high.xls)]>" }
Number WindSpeed <wind> \
{ http="<[weather:60000:XSLT(yahoo_weather_wind_speed.xls)]>" }
Group Temperatures
```



[Download all listings in this issue as text](#)

dering data as charts in the user interface.

Automating your world. Most of the fun use cases for a smart home are related to automation. In order to allow complex as well as unusual use cases, openHAB comes with a textual rule language. A *rule* in openHAB consists, in general, of one or many triggering conditions and a code block. The triggering conditions can be either event-based or time-based.

If multiple triggers are listed for a rule, any of them will trigger the code execution. The syntax of the code block of a rule uses a custom script language that is similar to Java and makes it easy to directly refer to items and their states.

Let's create the rule for our example use case. The triggering condition should be the change of the `Forecast` item's status, which is simply formulated as `Item Forecast changed`. In the code block, we have to transform the temperature value into a hue value (which represents a degree on the Hue's color circle). Let's map 60°F to 240° (=blue) on the `hue scale` and map 80°F to 0° (=red).

All that is left to do is to send a command that determines the appropriate color to show for the projected forecast encoded as HSB (hue, saturation, brightness) to the item `Light`. The resulting rule is shown in **Listing 5**. We store this rule for weather forecast visualiza-

//embedded /

tion in a file called `weather.rules` in the `configuration/rules` folder.

Once everything is set up, it is time to launch the openHAB runtime. All we need to do is to execute `start.sh` or `start.bat` from the command line, which in turn launches the JVM with the openHAB application. A typical startup looks like **Figure 3.**

When the runtime is up, you can point your browser to <http://localhost:8080/openhab.app> to see the Classic UI, which is a web UI optimized for touch devices. Alternatively, you can use the native Android or iOS openHAB apps, which automatically discover the openHAB runtime in the local network. All UIs will show the same content based on the sitemap definition. **Figure 4** shows the openHAB apps with a cold-weather forecast visualized by blue colored light (in the background).

What Else Can openHAB Do?

Our example use case gives a first glimpse into what can be achieved with openHAB, but openHAB has many other interesting features.

For example, in some situations, it is more convenient to be notified by a voice rather than by your smartphone. Imagine you are at home. When there is an incoming call, openHAB can announce the caller's name through a connected

speaker system. Similarly, openHAB can inform you that the clothes dryer has finished its work. Such use cases can be realized using text-to-speech (TTS) engines that can be plugged into the runtime. openHAB comes with two pure Java implementations (maryTTS and freeTTS) and an integration with Apple's built-in TTS engine.

Another useful feature when dealing with recurring tasks is openHAB's support for Google Calendar. With this add-on, you can add events to your calendar (see **Figure 5**) that are automatically synchronized by openHAB and executed at the specified time. This feature lets you easily change the trigger times for automation rules, such as a rule for wake-up times.

my.openHAB

One typical issue for a "do it yourself" (DIY) smart home is accessing the system remotely. This usually involves punching holes in the router's firewall by setting up port forwarding, and it also requires a dynamic domain name service (DNS) entry at some service provider. Alternatively, a safer solution is to set up a dedicated virtual private network (VPN). Neither option is easy or straightforward and, thus, users shy away from them.

But although it is not usually important to be able to control

```
haleiwa:runtime kai$ ./start.sh
Launching the openHAB runtime...
osgi> 06:32:41.276 INFO  o.o.c.internal.CoreActivator[:61] - openHAB runtime has been started (v1.4.0).
06:32:44.827 INFO  o.o.i.s.i.DiscoveryServiceImpl[:72] - mDNS service has been started
06:32:44.961 INFO  o.o.io.rest.RESTApplication[:141] - Started REST API at /rest
06:32:52.013 INFO  o.o.m.c.i.ModelRepositoryImpl[:79] - Loading model 'default.sitemap'
06:32:52.181 INFO  o.o.m.c.i.ModelRepositoryImpl[:79] - Loading model 'rrd4j.persist'
06:32:52.241 INFO  o.o.m.c.i.ModelRepositoryImpl[:79] - Loading model 'my.items'
06:32:54.243 INFO  o.o.u.w.i.s.WebAppServlet[:79] - Started Classic UI at /openhab.app
06:32:58.175 INFO  o.o.c.s.AbstractActiveService[:169] - HTTP Refresh Service has been started
06:32:58.952 INFO  runtime.busevents[:26] - Temperature state updated to 56
06:32:58.964 INFO  runtime.busevents[:26] - Forecast state updated to 57
06:32:58.986 INFO  runtime.busevents[:26] - WindSpeed state updated to 3
06:33:02.389 INFO  o.o.m.c.i.ModelRepositoryImpl[:79] - Loading model 'weather.rules'
```

Figure 3

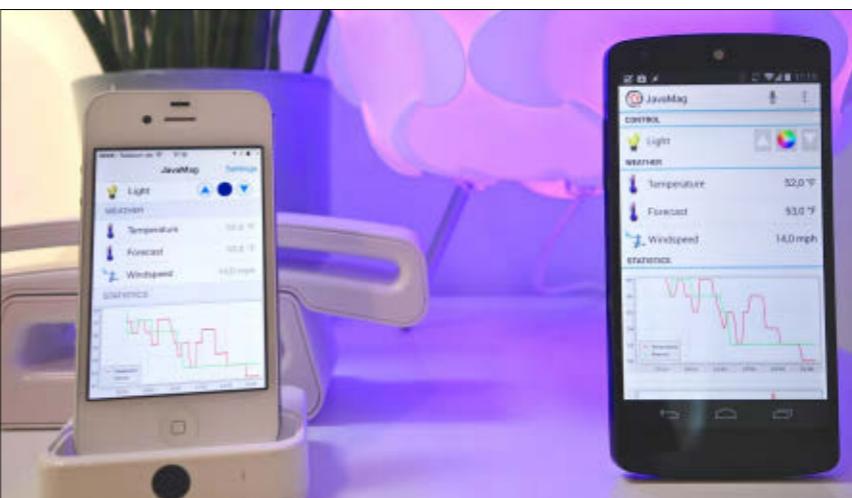


Figure 4

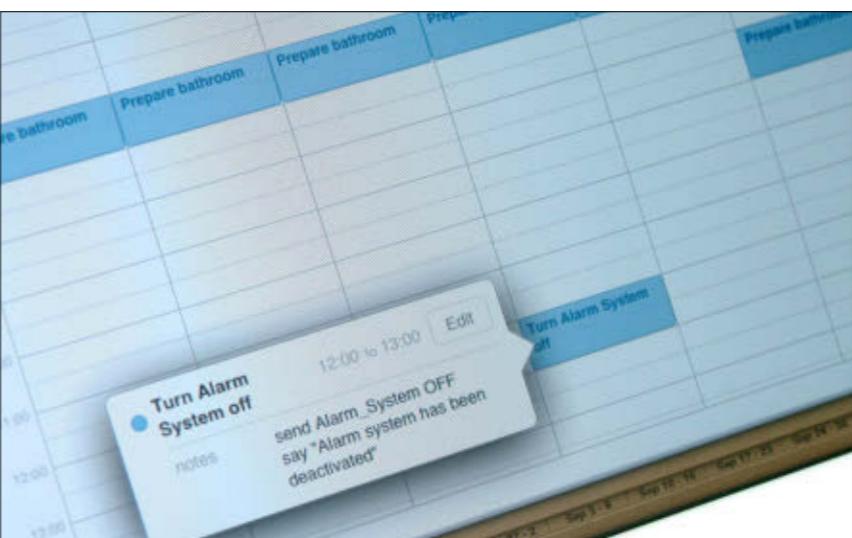


Figure 5

//embedded /

lights and shutters remotely, security aspects are a major reason why people want to have remote access. Are all windows (still) shut? Has any motion detector fired? For questions like these, remote access is, hence, more about monitoring than about controlling.

To make life easy for openHAB users, there is a free service called my.openHAB, which is currently in a closed beta phase. All you need to do to get this free service is install a special add-on in the openHAB installation, which causes openHAB to connect securely to the my.openHAB server and to maintain a permanent Secure Sockets Layer (SSL)-encrypted tunnel. The my.openHAB server can then anytime act as an internet proxy to a personal openHAB installation based on account credentials. Not only can the web UIs be accessed, but this add-on also works transparently for native Android and iOS clients.

Another very useful feature of my.openHAB is the possibility to send push notifications to your smartphone. While this is also possible through third-party apps such as Prowl and NMA (Notify My Android), the my.openHAB integration

tion adds this capability directly to the openHAB smartphone app itself. In the future, more functionality will be added to the free service for the openHAB user community.

The Future

At present, openHAB is a smart home solution that appeals to Java developers and others who are tech-savvy. This is mainly due to the fact that its configuration is done through text files. In order to broaden the target audience, new features for users who do not have a technical background are being developed, such as convenient UIs for system setup and configuration.

In 2013, the openHAB core was donated to the Eclipse Foundation as the new [Eclipse SmartHome](#) project.

This project provides a flexible and modular framework for building home gateways in Java, and it is meant to allow other solutions besides openHAB to be built on the same foundation. The upcoming openHAB 2.0 will, thus, be based on the Eclipse SmartHome stack, and further solutions are expected to become available as well.

Conclusion

Smart homes currently involve many different technologies that use a great variety of protocols, and this is unlikely to change in the future. In such environments, integration platforms are important for realizing truly smart homes. A competitive advantage of openHAB is its huge, vibrant community, which was formed because openHAB is open source, flexible, extensible, vendor-neutral, and hardware-agnostic; its use of Java helped with all of these aspects.

Besides smart homes, quite a few scientific projects and companies already count on openHAB and contribute their work back to the community. Furthermore, many universities use openHAB for lectures and student courses.

What is appreciated, in particular, is openHAB's modular design. Through Java, openHAB has access to a huge ecosystem of libraries, great debugging tools, and a professional developer community. Java's "write once, run anywhere" paradigm makes openHAB available on all sorts of different platforms. Java and openHAB form a versatile environment for writing unique and extraordinary success stories. </article>

[LEARN MORE](#)

• “Smartest House on the Street”



//fix this /



In the July/August 2014 issue,

Abhishek Gupta asked what happens when we mix generic and raw types. The correct answer is #3. Compilation is successful but there is a `java.lang.ClassCastException` at runtime. This exception is due to the fact that a `java.lang.Integer` type makes its way into the `List` because it is not parameterized, and hence the compiler fails to apply type safety. It is fixed by changing the signature of the `add` method to use generics.

Answer #1 is incorrect because adding a primitive datatype (for example, `int`) is perfectly legal.

Answer #2 is incorrect because the program compiles successfully since compile type safety cannot be applied to a raw `List` type.

Answer #4 is incorrect. The program does generate a `java.lang.ClassCastException`, but usage of `List<Object>` as the method parameter does not work because generic types are invariant.

This issue's code teaser comes from Cyril Lapinte, a senior Java consultant in Geneva, Switzerland, who presents us with a streams challenge.

1 THE PROBLEM

The Java SE 8 `java.util.stream` package is a powerful toolset providing many optimizations for manipulating data within collections. The complexity is hidden, letting developers concentrate

on the "what," not on the "how."

Nonetheless, developers should still keep in mind the underlying mechanisms.

2 THE CODE

This program finds the COUNT number of prime numbers that are greater than some random starting value. It runs slowly because it's not making effective use of the Stream API. What change could be made that would result in a considerable speedup?

```
static final int MAXSEEDVALUE = 200_000;  
static final int SEEDVALUE = new Random().nextInt(MAXSEEDVALUE);  
static final int COUNT = 10;
```

```
System.out.println(  
    IntStream.rangeClosed(SEEDVALUE + 1, MAXSEEDVALUE)  
        .parallel()  
        .filter(i -> IntStream.range(2, i)  
            .filter(j -> i % j == 0)  
            .count() == 0)  
        .limit(COUNT)  
        .mapToObj(String::valueOf)  
        .collect(Collectors.joining(" ")));
```

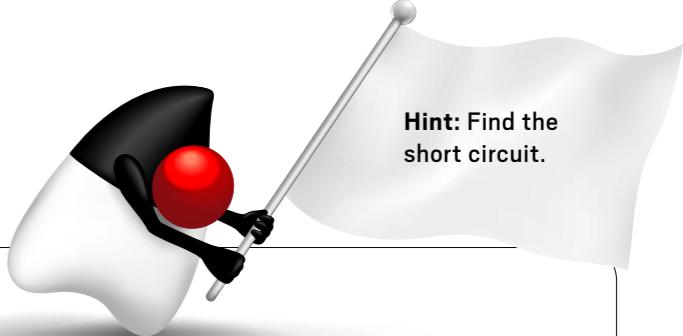
3 WHAT'S THE FIX?

- 1) Replace the lambda expression (`j -> i % j == 0`) by an anonymous class implementing the `IntPredicate` interface.
- 2) Move the `limit()` function before the outer `filter()`.
- 3) Use `noneMatch()` instead of the inner `filter()`.
- 4) Replace both `filter()` functions using the iteration of the Java Collections Framework.

GOT THE ANSWER?

ART BY I-HUA CHEN

Look for the answer in the next issue. Or submit your own code challenge!



Hint: Find the short circuit.