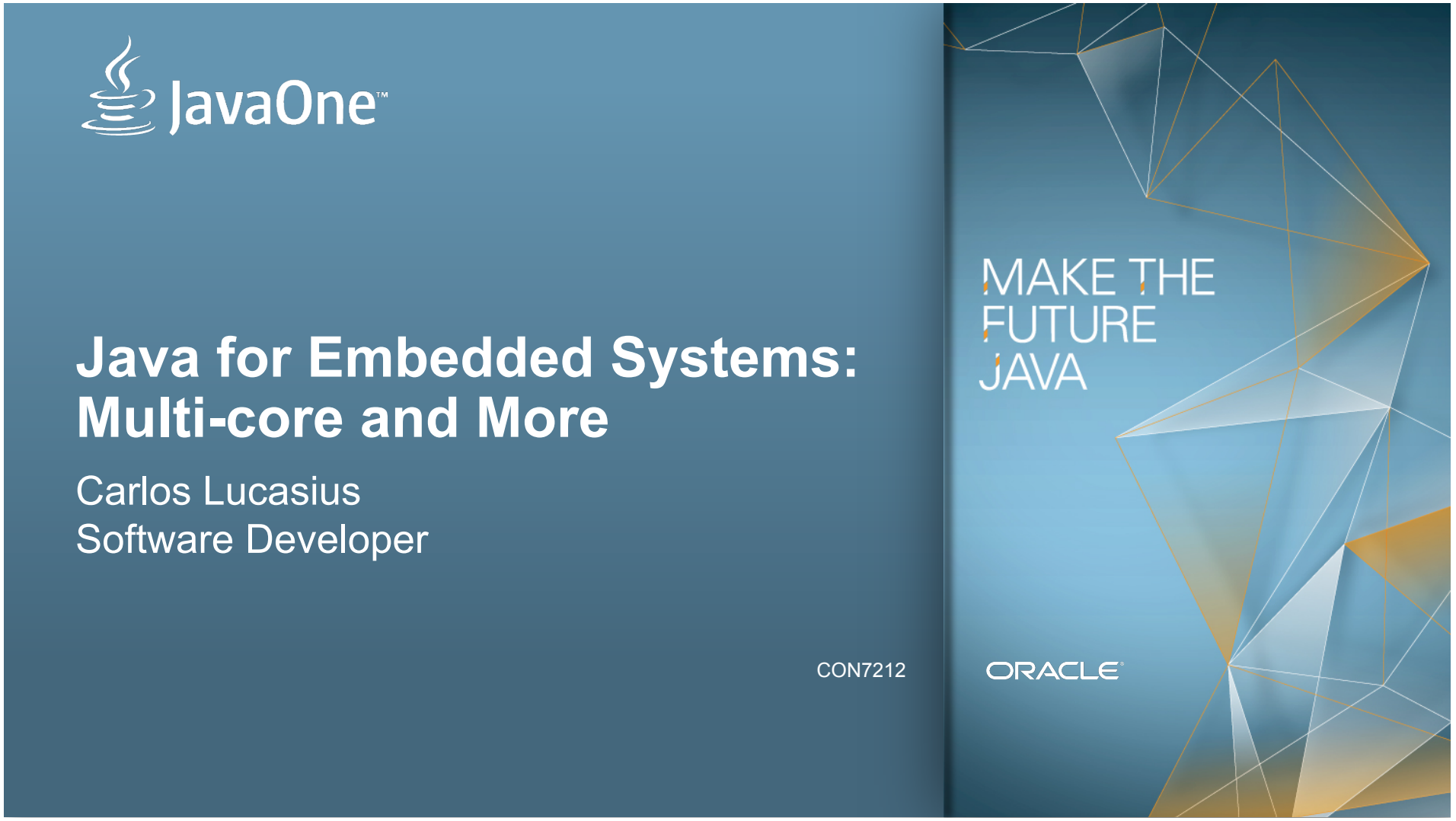

JavaOne™


JavaOne™
Java for Embedded Systems:
Multi-core and More
Carlos Lucasius
Software Developer
CON7212
MAKE THE
FUTURE
JAVA
ORACLE

The following 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.

Program Agenda

- Introduction to Java SE Embedded
- Java Embedded Value-Added Platforms
- Multi-core Support
- Troubleshooting Support
- Demo

Introduction to Java SE Embedded

Motivation for Java Embedded

- The Internet of Things
- Ecosystem with Embedded devices at the Edge
- Huge platform diversity across Embedded devices
  - Java has what it takes to address this challenge: portability
  - Java is a must in this space
- Java Availability, Reliability and Performance are key!

Introduction to Java SE Embedded

Java SE optimized for embedded use

- Java SE implementation optimized for deployment on high-end embedded devices:
  - Memory optimizations
  - Runtime optimizations
  - Power optimizations
- SE compliant: fully implements Java SE specification
  - Promotes portability across different Java SE platforms
- Multi-core enabled

Introduction to Java SE Embedded

Example devices










- ATMs
- Parking Meters
- POS Systems
- Lottery/Gaming Systems
- Multi Function Printers
- Intelligent Power Module
- Netbooks

- Routers & Switches
- Storage Appliances
- Network Management Systems
- Medical Imaging Systems
- Radar Systems
- Industrial PCs
- Factory Automation Systems
- Geo-Imaging Devices

- Smart Meters
- RFID Readers
- Video Conferencing Systems
- In-Flight Entertainment Systems
- Video Streaming Systems
- Electronic Voting Systems
- Voice Messaging Systems
- Security Systems

Introduction to Java SE Embedded

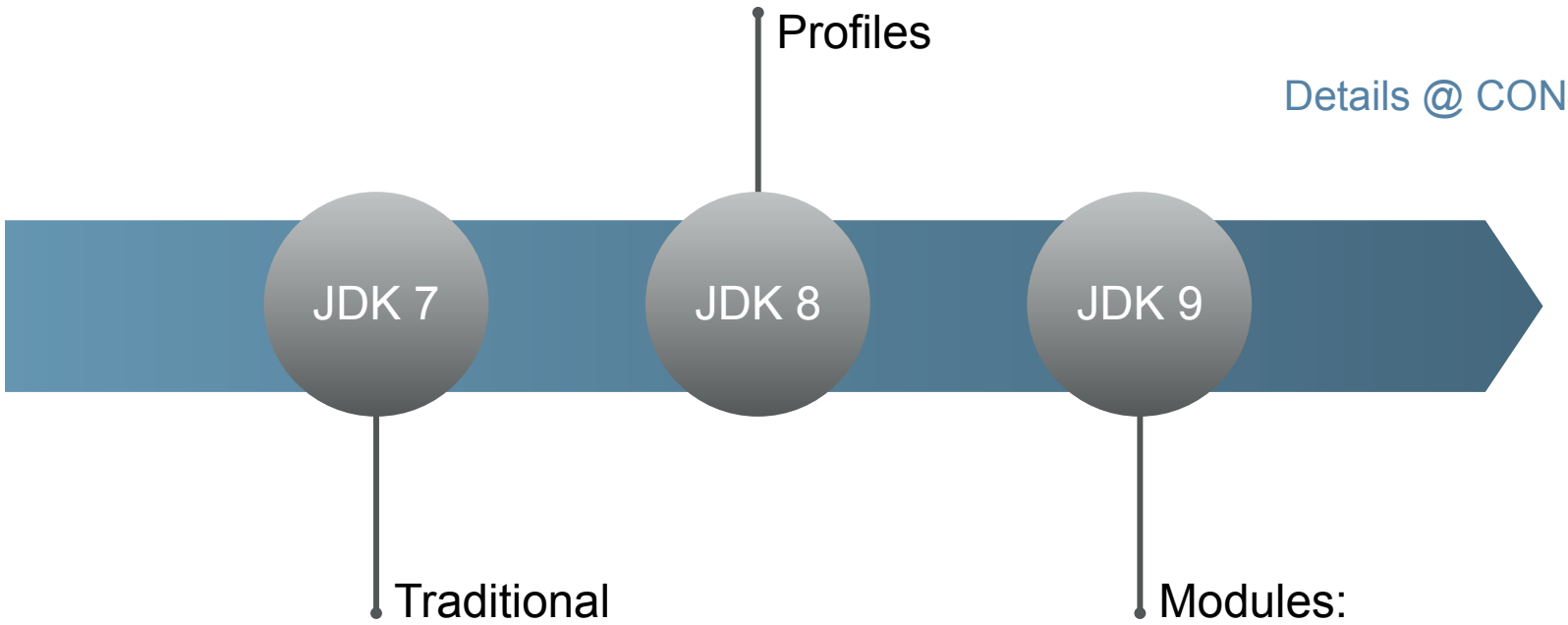
Currently Supported Platforms and Releases

Processor	Operating System	Headless or Headful	FPU	Java SE version
ARMv6/v7	Linux	Headful	VFP	JDK 7u6
ARMv5	Linux	Headless	Soft	6u34, 7u6
ARMv6/v7	Linux	Headless & Headful (v7)	VFP	6u34, 7u6
ARMv7 (Server JIT)	Linux	Headless	VFP	7u6
PowerPC e600 core	Linux	Headless	Classic HW FP	6u34, 7u6
PowerPC e500v2 core	Linux	Headless	Embedded FP	6u34, 7u6
x86	Linux	Headless	x86	6u34, 7u6

http://www.oracle.com/technetwork/java/embedded/downloads/javase/index.html

Introduction to Java SE Embedded

Platform Evolution: more flexibility in feature selection to meet requirements



 | |

JavaOne ORACLE

Java Embedded Value-Added Platforms

Java Embedded + More: Integrated Solutions

- Java Embedded Suite (JES):
  - Java SE Embedded + Embedded Middleware:
    - Glassfish
    - JavaDB
    - Jersey
    - More
  - Released September 25, 2012
- More Java Embedded Value-Added Platforms may be expected

Multi-core Support

Virtual Processors

- Clock rates have hit a plateau
- Moore's transistor count now needs more cores, not faster ones
- Multi-Processor (#sockets: *p*)
  - Multi-Core (#cores / processor: *c*)
    - Multi-Thread (#HW threads / core: *h*)
- #Virtual Processors == total #HW threads: $p \cdot c \cdot h$

Multi-core Support

JVM request #virtual processors: Runtime.availableProcessors()

```
int os::active_processor_count() {
  int online_cpus = sysconf(_SC_NPROCESSORS_ONLN);
  pid_t pid = getpid();
  psetid_t pset = PS_NONE;
  // Are we running in a processor set?
  if (pset_bind(PS_QUERY, P_PID, pid, &pset) == 0) {
    if (pset != PS_NONE) {
      uint_t pset_cpus;
      // Query number of cpus in processor set
      if (pset_info(pset, NULL, &pset_cpus, NULL) == 0) {
    assert(pset_cpus > 0 && pset_cpus <= online_cpus, "sanity check");
    _processors_online = pset_cpus;
    return pset_cpus;
      }
    }
  }
  // Otherwise return number of online cpus
  return online_cpus;
}
```


Multi-core Support

Observations

- Multi-core has been in Java for a long time (server apps), introduced into Java SE Embedded for ARM and PPC a few years ago
- Take advantage of inherent parallellism, both in Java platform (parallel GC, JIT compilation, etc.) and Java application (Java Concurrency, Fork-Join, Lambda)
- Insulates developer from intricacies of having to deal directly with thread affinities on multi-core, can instead focus on actual programming task

Troubleshooting Support

Responding to Features requested by Embedded Developers

- Profiling support added as of 6u25 (HPROF): http://www.oracle.com/technetwork/java/javase/6u25embeddedrelnotes-356542.html
- Limited Serviceability support added as of 7u2 (and more to come): http://www.oracle.com/technetwork/java/javase/javase-emb7u2-relnotes-1395056.html
- Continued “HotRockit” (HotSpot + Java ME + JRockit) convergence also leading to more troubleshooting tooling convergence
  - Java Flight Recorder
- NetBeans Profiler support on ARM available in NetBeans Dev/EA builds since mid-Summer 2012: http://bits.netbeans.org/netbeans/trunk/nightly/

Troubleshooting Support

Interfaces exposed by the JVM for Debugging, Profiling, Monitoring

- Java VM Tools Interface: **JVMTI**
- Java Monitoring and Management Interface: **JMX**
- Both interfaces fully implemented in Java SE Embedded
- An agent thread is typically started inside the Java VM process – the agent interacts directly with the Java VM interface and provides a front-end interface for tools, hence acts like a proxy
- Many agents networking-enabled, thus allowing remote troubleshooting

Troubleshooting Support

HPROF

- One of various tools available to generate heap dumps
- Heaps contain a wealth of useful information on Java execution: snapshot at dump time has details on current state, also some history
  - Great in helping find root causes of Java performance issues
    - Example: excessive GC caused by Java memory leak
- How to use: java -agentlib:hprof[=option1,option2,...] Main
  - Example: java -agentlib:hprof=heap=dump,format=b
  - HPROF data format platform independent

Java Monitoring, Profiling, Tuning

NetBeans Profiler

- Powerful profiler, originates from Research Labs
- Implementation on ARM currently being developed and already accessible:
  - Availability since mid-Summer 2012:

    NetBeans 7.3 Dev/EA version, check nightly builds:

    http://bits.netbeans.org/dev/nightly/

Java Monitoring, Profiling, Tuning

Java Management and Monitoring

- How to use:

```
java  -Dcom.sun.management.jmxremote.port=9999 \
      -Dcom.sun.management.jmxremote.ssl=false \
      -Dcom.sun.management.jmxremote.authenticate=false \
      Main
```

- com.sun.management.HotSpotDiagnosticMXBean supported

  Represents yet another way to dump heap on demand (HPROF-Format)

Demo

Java Remote Profiling on ARM

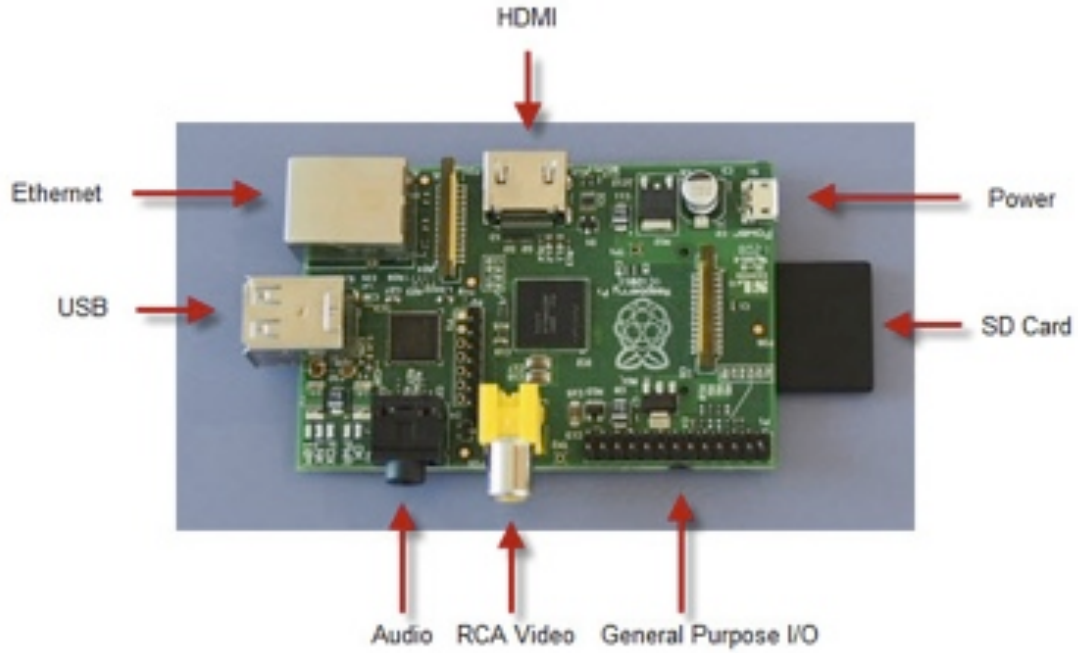
- On Raspberry Pi (ARMv6) device
- HPROF
- NetBeans Profiler
- Management & Monitoring

Demo

Raspberry Pi Embedded board Model B

Raspberry Pi

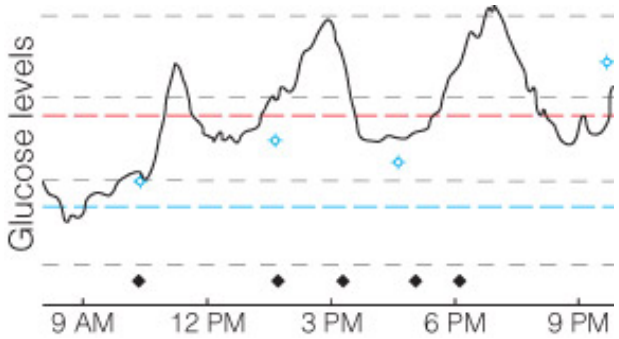
Embedded board Connectors



Demo

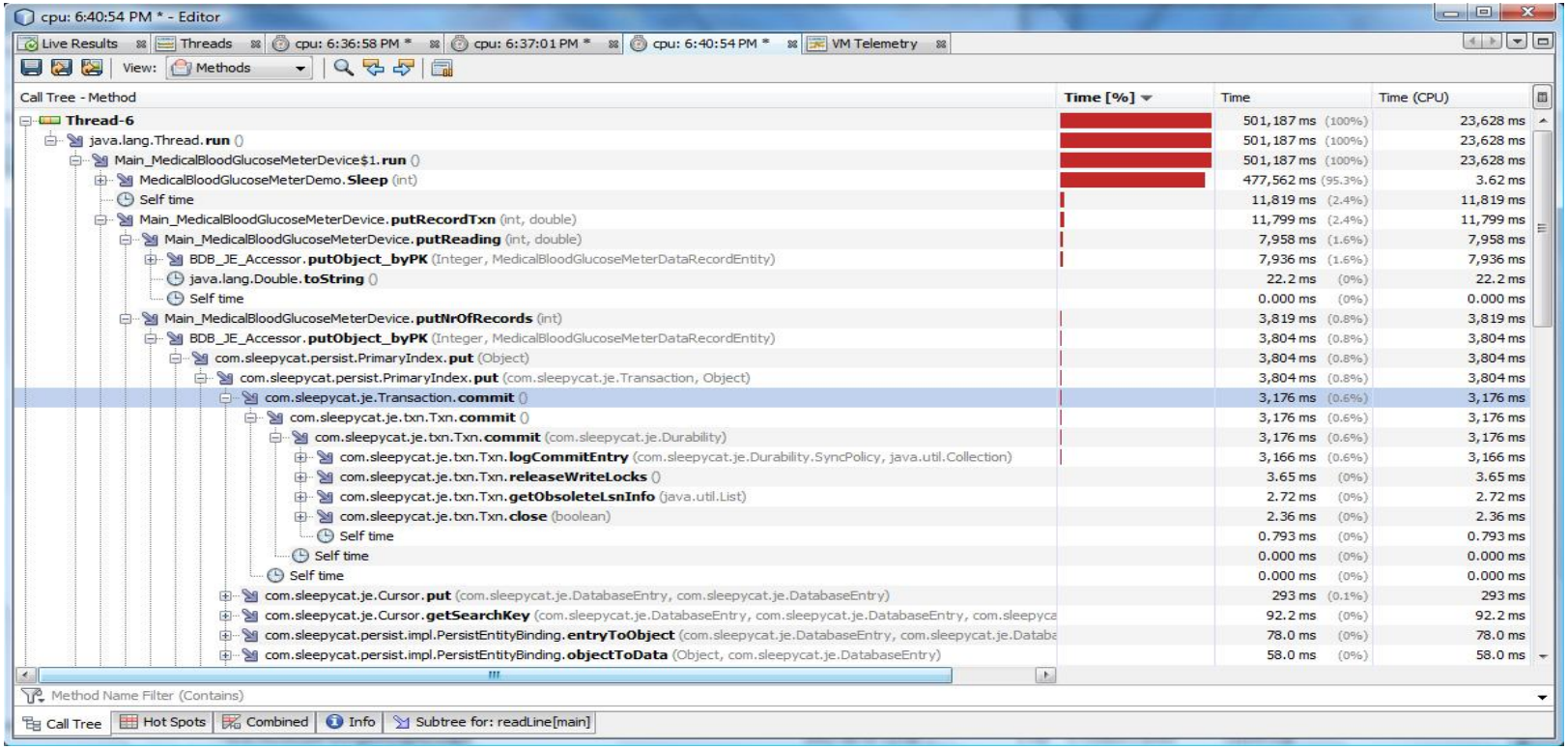
Java in Medical Blood Glucose Meter – Proof-of-Concept

- Simulated sensor: blood glucose-level data measurements realistically simulated
- Collected data stored in Berkeley Database (Java Edition)
- Remote access to meter using RMI



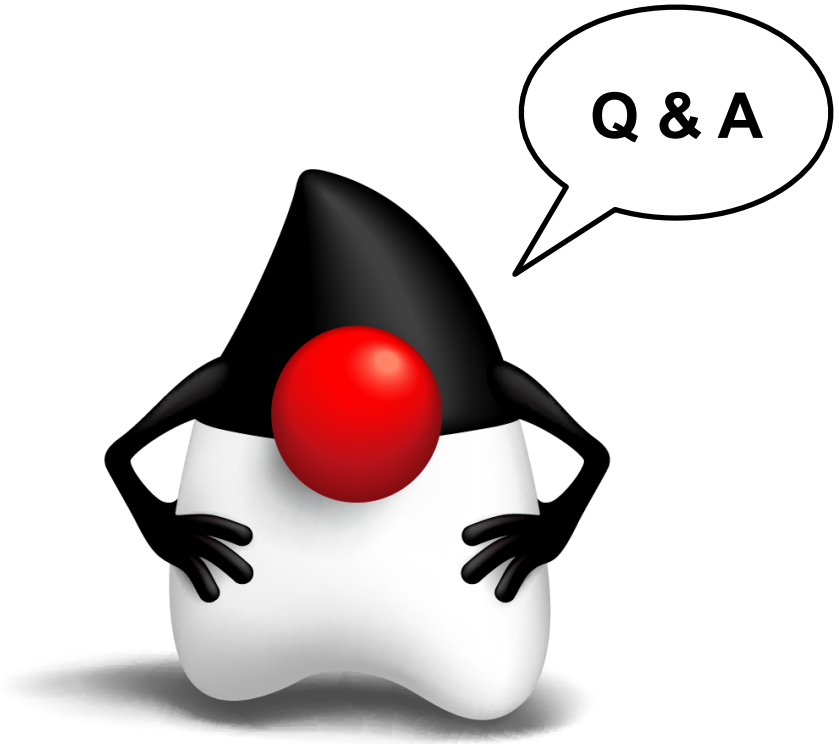
Troubleshooting Support(Dev/EA version)

NetBeans Profiling of Java SE Embedded (ARM) running Blood Glucose Meter



 | |


MAKE THE
FUTURE
JAVA
JavaOne
ORACLE



- Home page:
  - http://www.oracle.com/technetwork/java/embedded/overview/getstarted/index.html
- Downloads:
  - http://www.oracle.com/technetwork/java/embedded/downloads/javase/index.html


JavaOne™