ORACLE®


JavaOne™
ORACLE®

Java Mission Control & Java Flight Recorder in JDK 9

A Sneak Peek

Marcus Hirt
Consulting Member of Technical Staff
Java Product Group
February, 2017

@hirt

Java Your Next (Cloud)

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Safe Harbor Statement

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.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Program Agenda

1. Introduction to JFR/JMC
2. Java Flight Recorder in JDK 9
3. Java Mission Control 6
4. Summary
5. Q&A

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Introduction to JFR/JMC

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Production Time Profiling and Diagnostics

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Production Time Profiling and Diagnostics

"The big challenge is no longer really performance. The big challenge is profiling, and especially profiling in production."

- Tony Printezis, JVM engineer, Twitter
(Devoxx 2015, "Life of a Twitter JVM Engineer", 49:49)

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

“Once more @javamissionctrl is saving my day! "
- Michael Nitschinger, SDK Engineer, Couchbase

"Java Mission Control is the best profiler by far.“
- T Jake Luciani, PMC Cassandra

“Java Mission Control is my current favourite profiler for the Oracle JVM."
- Nitsan Wakart, Azul

"JMC not only saves time trying to resolve performance issues and bugs, it can give you a detailed view on your application you cannot get with other commercial profilers"
- Peter Lawrey, CEO, Chronicle Software

“… Our real-time messaging products can publish millions of messages a second to many thousands of connections - only JMC can keep up with this level of load."
- Phil Aston, Product Architect, Push Technology

"For the record: Java Mission Control is the best profiler ever, I use it daily, and so should you"
- Marcus Lagergren, Lead Architect, Nasdaq

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

"I am ACS engineer since 2008, delivering local Middleware support to several customers. Since I started to work with Java/JRockit Mission Control, it became a key tool for my work, helping me to troubleshooting, identifying root causes and bottlenecks, and also for doing proactive follow up services to customers. Without it, I would be blind."

- Iratxe Etxebarria, Oracle (ACS)

"In Fusion we create hundreds of thousands of Flight Recordings, and we can figure out 95% of the issues using nothing but the recordings."

- Joe Albowicz, Oracle (Fusion Application Development)

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

“when someone sends me a screenshot of Java Mission Control I can't help but feel the need to respond w/ "in what year was this taken...90s? "”
- @autoletics

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Definitions

- Java Flight Recorder (JFR)
  - Highly Efficient Data Recorder, built into the Java Runtime
- Java Mission Control (JMC)
  - The client application
  - Visualization of Flight Recordings
  - JMX console
  - Heap dump analyzer
  - …

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Flight Recorder Helps You To...

- Resolve problems faster
- Find bottlenecks in applications
- Do post mortem analysis, even from crash dumps

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Flight Recorder Performance

Extremely Low Overhead

- Built into the JVM/JDK, by the people developing the JVM
- High performance flight recording engine and high performance data collection
  - Access to data already collected in the runtime
  - Thread local native buffers
  - Invariant TSC for time stamping
  - Method profiling data even from outside safe-points
  - Faster and more accurate allocation profiling (scalarization not undone by profiler)

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

JMC & JFR Demo

JDK 8, JMC 5.5

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Program Agenda

1. Introduction to JFR/JMC
2. Java Flight Recorder in JDK 9
3. Java Mission Control 6
4. Summary
5. Q&A

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Improvements

- New **Supported** APIs
  - Easier to use
  - Moved namespace from oracle.jrockit.* to jdk.jfr.*
    - Not compatible with old unsupported APIs
  - Modularized
- Performance enhancements
  - Compressed Integers
  - Smarter Event Classes
    - Event reference does not escape into the generated code
    - No event object reuse required

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Improvements

Continued

- Can emit data to disk even in bad situations
  - Useful in fatal situation, e.g. OOM or crash
- New Events
  - More detailed safe point information
  - More detailed code cache information
  - New PLAB events
  - New compiler events for detailed inlining information
  - New G1 specific information for better visualization of region states
  - Module events (loaded module, full transitive closure)
  - NativeLibrary (load, periodic event, by default each chunk)

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

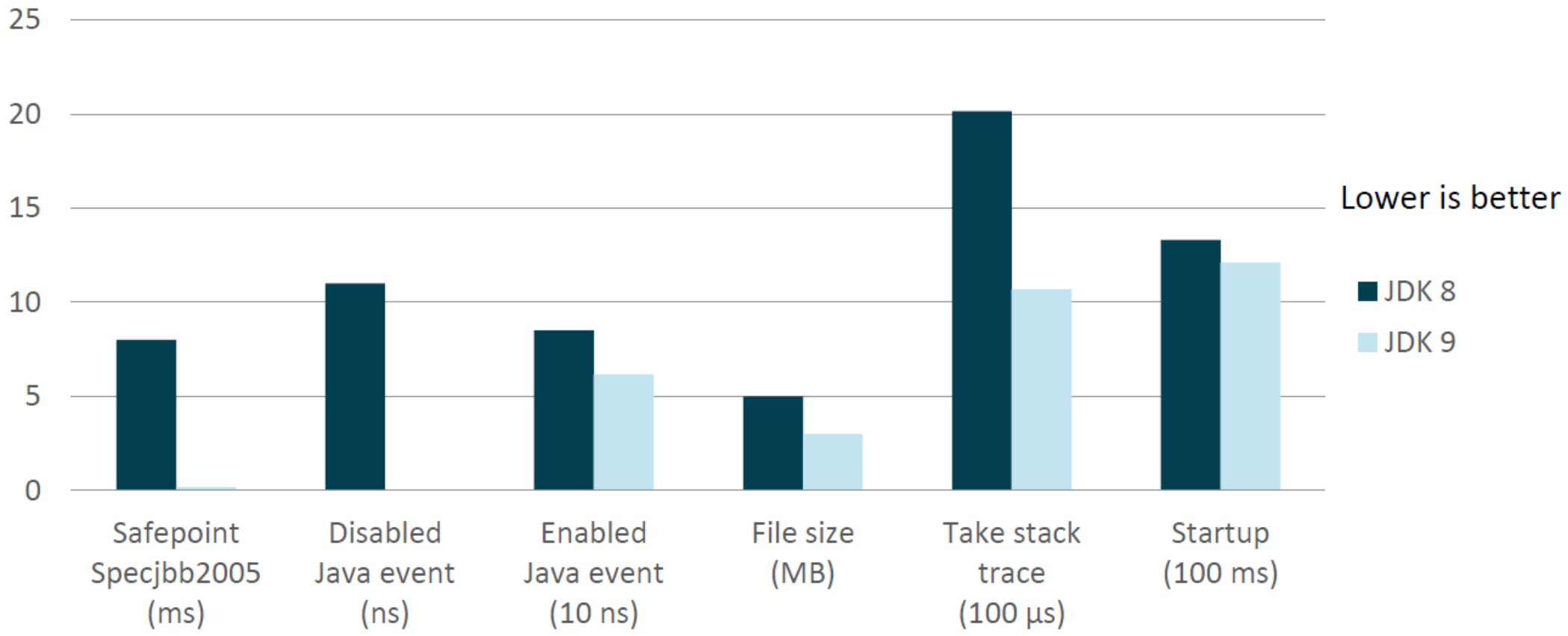
Supported API for Custom Events

- Supported!
- Easy to correlate with events from the runtime
- Piggy back on the whole JFR infrastructure (jcmd, jmc, commandline)
- High performance
  - High precision, cheap timestamping
  - Cheap stack traces
  - Binary, compact data
- Self describing, easy to consume

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

API Improvements, Performance



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Supported API for Custom Events

Continued

- Extensible
  - Custom settings can be defined for event types
  - User definable metadata using annotations
- Simplified Programming Model
  - EventTypes are self registering (no more Producers)
  - Life cycle follow the event type class
  - New event types are enabled by default
    - As a matter of fact event types can be fully configured with annotations
  - No need to be careful with event object creation

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

New API for Custom Events

Example

```
import jdk.jfr.Event;
import jdk.jfr.Label;

public class Hello {

    @Label("Hello World")
    static class HelloWorldEvent extends Event {
       @Label("Message")
       String message;
    }

    public static void main(String... args) {
        HelloWorldEvent event = new HelloWorldEvent();
        event.message = "Hello World event message!";
        event.commit();
   }
}
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Example from jfr4junit

JFR plug-in for the upcoming JUnit 5

```
/**
 * Event for a test run and completed normally (without exception).
 *
 * @author Marcus Hirt
 */
@Label("Test Run")
@Description("JUnit test executed and completed normally without exception")
@Category("JUnit")
public class TestEvent extends Event {
    @Label("Display Name")
    @Description("The display name for the test")
    private String displayName;

    public void setDisplayName(String displayName) {
        this.displayName = displayName;
    }

    public String getDisplayName() {
        return displayName;
    }
}
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Example from jfr4junit

Another example

```
/**
 * Event for when a test ended with an exception being thrown.
 *
 * @author Marcus Hirt
 */
@Label("Test Exception")
@Description("Test ended with an exception being thrown")
@Category("JUnit")
public class ExceptionEvent extends Event {
	@Label("Display Name")
	@Description("The display name for the test")
	private String displayName;

	@Label("Exception Message")
	@Description("The exception message for the exception")
	private String exceptionMessage;

	@Label("Exception class")
	@Description("The class of the exception")
	private Class<?> exceptionClass;

	public String getDisplayName() {
		return displayName;
	}

	public void setDisplayName(String displayName) {
		this.displayName = displayName;
	}
.
.
.
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

New APIs for Controlling the Flight Recorder

- New Java POJO API
  - Supported!
  - Makes it possible to use JFR on JMX-less profiles
  - Cleaner and more expressive for in-process use
- New JMX API
  - Supported!
  - In a separate module
  - Cleaner and easier to use
    - Uses MXBeans and type mapping instead of CompositeData directly

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

New API for Parsing Flight Recorder Files

```
public class ParserExample {
    public static void main(String[] args) throws IOException {
        long maxNanos = Long.MIN_VALUE;
        RecordedEvent maxEvent = null;
        for (RecordedEvent event : RecordingFile.readAllEvents(Paths.get(args[0]))) {
            if (event.getEventType().getName().equals("se.hirt.jfr4junit.TestEvent")) {
                long nanos = event.getDuration();
                if (nanos > maxNanos) {
                    maxNanos = nanos;
                    maxEvent = event;
                }
            }
        }
        System.out.printf("Longest running test was: %s for %ds\n",
             maxEvent.getValue("displayName"), maxNanos / 1_000_000);
        System.out.println("Event was:\n" + maxEvent);
    }
}
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Control API and Parsing Example

```
public class RecordAndConsume {
    public static void main(String[] args) throws IOException {
        Path path = Paths.get(args[0]);
        try (Recording recording = new Recording()) {
            recording.setName("Fibonacci Recording");
            recording.start();
            recording.enable(FibonacciEvent.class);
            for (int n = 0; n < 50; n++) {
                FibonacciEvent event = new FibonacciEvent();
                event.number = n;
                event.begin();
                event.value = Fibonacci.fibonacciIterative(n);
                event.commit();
            }
            recording.stop();
            recording.dump(path);
            for (RecordedEvent event : RecordingFile.readAllEvents(path)) {
                int number = event.getValue("number");
                long value = event.getValue("value");
                System.out.printf("fibonacci(%d) = %d (time: %dns)\n", number, value, event.getDuration());
            }
        }
    }
}
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Changed Command Line Flags

In-memory example

JDK 8

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:FlightRecorderOptions=defaultrecording=true,
dumponexit=true,dumponexitpath=/home/hirt/myrecording.jfr
```

JDK 9

```
-XX:+UnlockCommercialFeatures -XX:StartFlightRecording=
dumponexit=true,filename=/home/hirt/myrecording.jfr
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Changed Command Line Flags

Disk Repository Example

JDK 8

```
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:FlightRecorderOptions=defaultrecording=true,
dumponexit=true,dumponexitpath=/home/hirt/myrecording.jfr,disk=true,
maxsize=500M,maxage=20m
```

JDK 9

```
-XX:+UnlockCommercialFeatures -XX:StartFlightRecording=
dumponexit=true,filename=/home/hirt/myrecording.jfr,maxsize=500M,maxage=20m
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

JCMD

JCMD will continue to work the same, for example:

```
>jcmd 4711 VM.unlock_commercial_features
>jcmd 4711 JFR.start duration=2m,filename=/home/hirt/myrecording.jfr
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Performance

JFR enabled, event disabled

```
private static long calculateFibonacci(int n) {
    FibonacciEvent event = new FibonacciEvent();
    event.begin();
    event.number = n;
    long fibValue = Fibonacci.fibonacciIterative(n);
    event.value = fibValue;
    event.commit();
    return fibValue;
}
```

```
0x000001524f68c8a0: sub     rsp,18h
0x000001524f68c8a7: mov     qword ptr [rsp+10h],rbp  ;*synchronization entry
                    ; - se.hirt.jdk9.enabledisable.EnableDisableTesterFibonacci::calculateFibonacci@-1 (line 54)

0x000001524f68c8ac: movsxd  r10,edx          ;*i2l {reexecute=0 rethrow=0 return_oop=0}
                    ; - se.hirt.jdk9.enabledisable.EnableDisableTesterFibonacci::calculateFibonacci@18 (line 57)

0x000001524f68c8af: test    r10,r10
0x000001524f68c8b2: jle     1524f68c8fah     ;*iflt {reexecute=0 rethrow=0 return_oop=0}
                    ; - se.hirt.jdk9.enabledisable.Fibonacci::fibonacciIterative@37 (line 14)
                    ; - se.hirt.jdk9.enabledisable.EnableDisableTesterFibonacci::calculateFibonacci@19 (line 57)

0x000001524f68c8b4: mov     eax,1h
0x000001524f68c8b9: mov     r11d,2h
0x000001524f68c8bf: mov     r8d,1h
0x000001524f68c8c5: mov     r9d,1h
0x000001524f68c8cb: jmp     1524f68c8e3h
.
.
.
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Performance

JFR enabled, event enabled

```
0x00000194572ba0c0: mov     dword ptr [rsp+0ffffffffffffa000h],eax
0x00000194572ba0c7: push    rbp
0x00000194572ba0c8: sub     rsp,20h          ;*synchronization entry
                                  ; - se.hirt.jdk9.enabledisable.EnableDisableTesterFibonacci::calculateFibonacci@-1 (line 54)
.
.
.

0x00000194572ba125: mov     rbp,rax          ;*invokespecial <init> {reexecute=0 rethrow=0 return_oop=0}
                                  ; - se.hirt.jdk9.enabledisable.EnableDisableTesterFibonacci::calculateFibonacci@4 (line 54)

0x00000194572ba128: mov     rdx,rbp
0x00000194572ba12b: call    194503f8060h      ; ImmutableOopMap{rbp=Oop }
                                  ;*invokevirtual begin {reexecute=0 rethrow=0 return_oop=0}
                                  ; - se.hirt.jdk9.enabledisable.EnableDisableTesterFibonacci::calculateFibonacci@9 (line 55)
                                  ;   {optimized virtual_call}

.
.
.
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Summary of JDK 9 JFR Features

- Easy to use supported APIs for all things Flight Recorder
  - Allows for custom events
  - Programmatic access for reading Flight Recordings
  - Programmatic access for controlling the Flight Recorder
  - Modularized, works on smaller profiles
- New events
- Improved command line ergonomics
- Can dump on crashes and OOM

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Program Agenda

1. Introduction to JFR/JMC
2. Java Flight Recorder in JDK 9
3. Java Mission Control 6
4. Summary
5. Q&A

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Java Mission Control 6

Highlights

- Major changes to JFR
  - Support for both the JDK 7/8 and JDK 9 JFR file formats
  - Support for connecting to JDK 7/8 and JDK 9
- Automated analysis of Flight Recordings
- Only released with JDK 9
- Only minor changes to the rest of JMC

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Automated Analysis of Flight Recordings

- Recordings contain quite a bit of runtime specific data
- Java promises an abstraction that hide implementation specifics
- Attempts to:
  - Flag information that is relevant
  - Provide links to documentation to explain relevant concepts (e.g. safe points, code caches, PLABs etc)
- Individual rules can be disabled

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Automated Analysis of Flight Recordings

Components and API

- POJO components usable outside of JMC
  - Parser
  - Rules engine
  - JDK 7
- Extensible through a Java API
  - Using the standard Java Service Loader mechanism
  - Experimental PDE plug-in which generates the boilerplate
  - Unsupported but documented in JMC 6
  - If there is enough interest -> supported in a future version

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Automated Analysis of Flight Recordings

Components and API, continued

- Can be associated with existing pages
- API features help simplify the development of rules
  - Filters
  - Aggregators
- The rules engine can be run outside JMC
  - Oracle internal PoC cloud service – upload JFR, get JSON back
  - Oracle internal analysis web application
    (API testing, rules testing and also great fun)
  - Used in Oracle Enterprise Manager for automated analysis of recordings

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Example Rule

```
public class EnvironmentVariableRule implements IRule {
    private static final TypedPreference<String> PREFERENCE_ENVIRONMENT_VARIABLE_NAME = new TypedPreference<>(
        "environmentVariable", "Environment Variable",
        "The name of the environment variable to check for a floating point score",
        UnitLookup.PLAIN_TEXT.getPersister(), "JFR_RULE_TEST");

    @Override
    public Collection<TypedPreference<?>> getConfigurationAttributes() {
        return Arrays.<TypedPreference<?>> asList(PREFERENCE_ENVIRONMENT_VARIABLE_NAME);
    }

   @Override
   public String getId() {
       return "se.hirt.envrule.EnvironmentVariableRule";
   }

   @Override
   public String getName() {
       return "Configurable Rule Demo";
   }

   @Override
   public String getTopic() {
       return JfrRuleTopics.ENVIRONMENT_VARIABLES_TOPIC;
   }

.
.
.
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Example Rule

Cont'd

```
.
.
.

    @Override
    public Result evaluate(IItemCollection items, IPreferenceValueProvider valueProvider) {
        String variableName = valueProvider.getPreferenceValue(PREFERENCE_ENVIRONMENT_VARIABLE_NAME);
        String environmentVariableValue = getEnvironmentVariable(variableName, items);
        if (environmentVariableValue == null) {
            return new Result(this, 100, "Could not find the environment variable named " + variableName);
        }
        double score = 0;
        try {
            score = Double.parseDouble(environmentVariableValue);
        } catch (NumberFormatException e) {
            return new Result(this, 100, "Could not parse the environment variable named " + variableName);
        }
        return new Result(this, score, "The score in " + variableName + " was " + score);
    }

    private String getEnvironmentVariable(String variableName, IItemCollection items) {
        IItemCollection envItems = items.apply(ItemFilters.and(JfrFilters.ENVIRONMENT_VARIABLE,
            ItemFilters.equals(JfrAttributes.ENVIRONMENT_KEY, variableName)));
        return envItems.getAggregate(JfrAggregators.first(JfrAttributes.ENVIRONMENT_VALUE));
    }
}
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Updated User Interface

- All new, separate and configurable Stack Trace View
  - By default showing the most common path (a full stacktrace)
  - Can be reconfigured to have the old tree representation
  - May take a bit of getting used to, but much easier to work with
- New configurable ”Java Application” page
  - Shows several key metrics together with per thread event lanes
  - Easier to focus on points of interest
- Requires JDK 8
  - UI uses streams and other JDK 8 language and API additions
  - If running in Eclipse – must be running on a JDK 8 JVM

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Updated User Interface

Cont'd

- New navigation
  - Uses the outline view
  - No tabs-in-tabs
  - Cleaner and more intuitive
- Advanced - Filters
  - Can build advanced filters through boolean operations (intersection, union)
  - Can be recording independent
  - Operative Set is no more -> Selections & Aspects

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Updated User Interface

Cont'd

- No more GUI builder
  - Embryo for easily adding custom event pages instead
  - Something similar, but supported, may resurge if there is enough interest
- Instead easy to create custom pages directly in the UI

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

JavaYourNext
(Cloud)

JMC UI Demo

JMC 6 / JDK 9

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Running the Rules

Simple example

```
public class RateFile {
    public static void main(String[] args) throws IOException, CouldNotLoadRecordingException {
        IItemCollection events = JfrLoaderToolkit.loadEvents(new File(args[0]));

        for (IRule rule : RuleRegistry.getRules()) {
            Result result = rule.evaluate(events, IPreferenceValueProvider.DEFAULT_VALUES);
            if (result.getScore() > 50) {
                System.out.printf("(%.0f) [%s]: %s\n", result.getScore(), result.getRule().getId(),
                    result.getShortDescription());
            }
        }
    }
}
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

JavaYourNext (Cloud)

jfrscore

Oracle Internal Example

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Confidential – Oracle

Summary

- JFR APIs are ***SUPPORTED*** in JDK 9
  - Controlling the recorder
  - Creating custom events
  - Parsing the recordings
- JMC 6 JFR part greatly revised
  - Automated analysis of recordings
  - Plug-in for easily adding custom rules
  - Will work on JDK 7, 8 and 9 recordings
  - Can be used stand alone for scripting (initially unsupported)

JavaOne ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Additional Resources (JMC 5.5 and JDK 7 & 8)

- JMC Homepage
  http://oracle.com/missioncontrol
- Hirt's blog
  http://hirt.se/blog
- Twitter
  @javamissionctrl, @hirt
- JMC Tutorial
  http://hirt.se/blog/?p=611

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Q&A

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Confidential – Oracle