JDK Mission Control - What is JMC?

A **tool suite** to monitor, manage, and profile Java apps

Main client for **Java Flight Recorder (JFR)** recordings

Built into the JDK — low overhead, production safe

Also includes:

- JMX Console
- HPROF dump analyzer
- Extensible with Eclipse plug-ins

JDK Mission Control - JDK Flight Recorder (JFR)

- **Always-on flight recorder** inside the JVM

- Captures:
  - Performance bottlenecks
  - Allocation & method profiling
  - GC pauses, I/O, thread states

- 🎥 Works like a **time machine**:
  - See what happened **before, during, after** an issue
  - Even from crash dumps

JDK Mission Control - Why JFR + JMC?

🔥 **Extremely low overhead** (no measurable impact)

✅ Safe & reliable in production

⚡ Built by JVM engineers → high accuracy

🛠️ Used internally at Oracle (Fusion apps)

"We can solve 95% of issues using nothing but recordings." – Oracle

JDK Mission Control - Example Workflow

1️⃣ App shows performance issue

2️⃣ Trigger a **JFR recording**

3️⃣ Load it in **Mission Control**

4️⃣ Inspect:

- Hot methods

- Allocation hotspots

- Threads & locks

- Latency outliers

  5️⃣ Fix bottlenecks, re-run

JDK Mission Control - Always-On Profiling

Keep JFR **running in production** (low cost)

Dump data **on demand** when an issue occurs

Even if JVM crashes → last minutes are persisted

Speeds up **post-mortem analysis** & Oracle support

JDK Mission Control - Installing JMC

Bundled with OpenJDK distributions (Red Hat, Oracle, Adoptium)

On Linux (RHEL): via CodeReady Builder repo

On Windows: included in OpenJDK archive

Can run standalone or inside Eclipse (plug-in)

JDK Mission Control - The JMC Agent

Add **custom JFR events** to running apps

Configure via XML or presets in JMC UI

Control what to record, when to record

Low memory overhead

JDK Mission Control - Typical Use Cases

🔍 Debugging performance regressions

⏱️ Finding latency spikes in production

🛠️ Supporting customers with JFR files

📊 Historical analysis with flight recordings

👨‍💻 Extend with plug-ins for team-specific metrics

JDK Mission Control - Key Takeaways 💡

JFR + JMC = production-safe observability

Lightweight, always-on profiling

Essential for diagnosing **hard-to-reproduce issues**

Works out of the box with modern JDKs

Used by Oracle at massive scale → proven tool