



# Through the JMX Window

Matt Brasier  
Head of Consulting  
C2B2 Consulting



# Who am I?

- Head of consulting at C2B2
- Application server consultant
  - WebLogic 5.1->Present
  - JBoss 3.x -> Present
  - Tomcat, etc
- Performance tuning and critical situation support
- Author



# Agenda

- What is JMX?
- JMX tools
- What information does it expose?
- What operations can JMX expose?
- Exposing your own information and operations

# WHAT IS JMX?



# What is JMX

- Java Management eXtensions
- Currently at version 1.4
- [http://docs.oracle.com/javase/7/docs/technote\\_s/guides/jmx/JMX\\_1\\_4\\_specification.pdf](http://docs.oracle.com/javase/7/docs/technote_s/guides/jmx/JMX_1_4_specification.pdf)

“The Java Management extensions (also called the JMX specification) define an architecture, the design patterns, the APIs, and the services for application and network management and monitoring in the Java programming language.”

# JMX History

- **1998** – JMX starts as JSR 3. EG composed of most J2EE vendors, based on JDMK
- **2003** – J2EE 1.4 incorporates JMX in the J2EE Spec
- **2004** – With Java 5 release, JMX becomes part of the JDK/Java SE
- JMAPI – Java Management API, failed Sun project
- JDMK – Java Dynamic Management Kit, another failed Sun project
- JMX



# What is management?

- Management Information
  - Any information that is useful for the runtime management of the application.
- Management Operations
  - Any operations that can assist in the runtime management of the application.

# Why JMX?

- Application Management is important!
  - If often forgotten.
- Standardise the APIs and patterns for exposing management operations and tooling.
  - Tools can be sure information is in an expected format.
  - Applications can take advantage of existing tooling.



# Core Concepts

- MBean
  - Management Bean
    - Operations
    - Attributes
- MBean Server
  - Registry for MBeans
- API
  - Remote and local access to MBean Server and MBeans

# Core Concepts

- Applications/Frameworks/Containers expose MBeans
- The JVM contains an MBeanServer which registers all MBeans
- Clients use the JMX API to connect to the MBeanServer and find relevant MBeans
- Clients read/change values of attributes and invoke operations to manage the service

# JMX TOOLS



# JMX Tools

- JConsole
- JVisualVM
- Monitoring tools
  - Hyperic HQ
  - Nagios
  - Etc



# Demo





# WHAT INFORMATION DOES IT EXPOSE?



# Default JMX info

- Heap Use
  - java.lang:type=Memory
  - java.lang:type=MemoryPool,name=Eden Space
  - java.lang:type=MemoryPool,name=Tenured Gen
  - java.lang:type=MemoryPool,name=Perm Gen
- Garbage collection
  - java.lang:type=GarbageCollector,name=MarkSweepCompact
  - java.lang:type=GarbageCollector,name=Copy
- Operating system info
  - java.lang:type=OperatingSystem

# Default JMX info

- Runtime JVM parameters
  - `java.lang:type=Runtime`
- NIO Buffers
  - `java.nio:type=BufferPool,name=direct`
- JDK Logger
  - `java.util.logging:type=Logging`
- Classloading
  - `java.lang:type=ClassLoading`
- And more!





# In an application server

- Connection pool info
  - Sizes
  - Free connections
- Servlet/JSP info
  - Number of requests
  - Number of errors
- Thread pools
  - Sizes
- JMS
  - Queue sizes
  - Pending messages
- EJBs
  - Pool sizes
  - Cache hit rates
- Connectors
  - In coming request volumes

# WHAT OPERATIONS DOES IT EXPOSE?



# Default operations

- `com.sun.management:type=HotSpotDiagnostic`
  - `dumpHeap`
  - `setVMOption`
  - `getVMOption`
- `java.lang:type=Memory`
  - `Gc`
- `java.lang:type=Threading`
  - `dumpAllThreads`
  - `findDeadLockedThreads`

# In an application server

- Datasources
  - Reset all datasources
- JMS
  - Delete all messages
- Server
  - Restart or stop a server
- Deployment
  - Deploy or redeploy applications

# In an application server

- Any operation you could normally perform to manage an application server!
- It is like an X-ray of your application



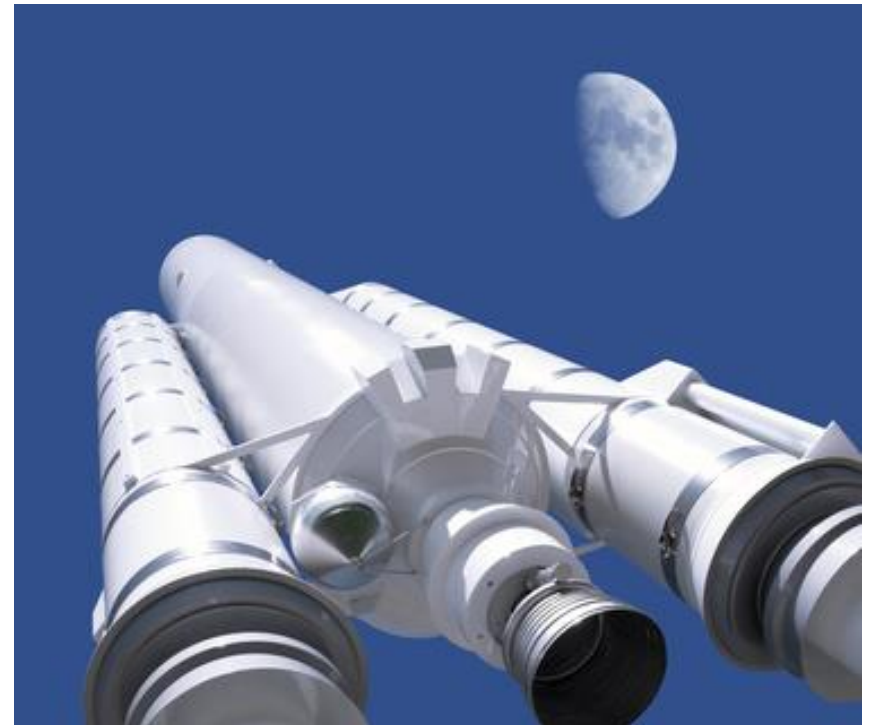


# EXPOSING YOUR OWN MANAGEMENT INFORMATION AND OPERATIONS



# Two step process

- Create MBeans
- Register with MBean server
- Profit



# Create MBeans

- Identify where in your application you have relevant management information
- Create an MBean interface
- Implement the MBean interface



# MBean Interface

```
public Interface LocationMonitorMBean
```

```
{
```

```
    public void getNumberOfFiles();
```

```
    public void setScanPeriod();
```

```
    public void getScanPeriod();
```

```
    public void resetFileCounter();
```

```
    public void disable();
```

```
    public void enable();
```

```
    public void getEnabled();
```

```
}
```

Getters and Setters represent  
Attributes

Other methods represent  
operations

# Implementation class

- Implement the MBean interface
  - You can do this with an existing class
- Try and avoid doing intensive work in the MBean implementation
  - Pre-compute values
  - Periodically check and cache values

# Implementation class

```
public class LocationMonitor implements LocationMonitorMBean
{
    private boolean enabled;
    private long scanInterval;
    private long numFiles;

    public void getNumberOfFiles(
    {
        return numFiles;
    }
    ...
}
```

Attributes are accessed via getters/setters

# Register with MBeanServer

- Create (or get hold of) an instance of your MBean implementation
- Get hold of an MBean Server
  - PlatformMBeanServer
- Create an ObjectName for your MBean
- Register the MBean

# Register with MBeanServer

```
public static void main(String[] args) throws Exception
{
    MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();
    ObjectName name = new
        ObjectName("uk.co.c2b2.management:type=LocationMonitor");
    LocationMonitor mbean = new LocationMonitor();
    mbs.registerMBean(mbean, name);
}
```

Get the MBean Server

Create the ObjectName

Create the MBean instance

Register it

# What to monitor

- Resources that your application uses
  - Including data (flow rates etc)
- Expose management operations
  - Restarting components
  - Re-loading configuration files

# What is through the JMX window?

- A future with more Information
- A future with more control





# QUESTIONS?



© C2B2 Consulting Limited 2013  
All Rights Reserved