

Oracle WebLogic Server

The #1 Application Server for
Enterprise Java and SOA Developers



ORACLE®

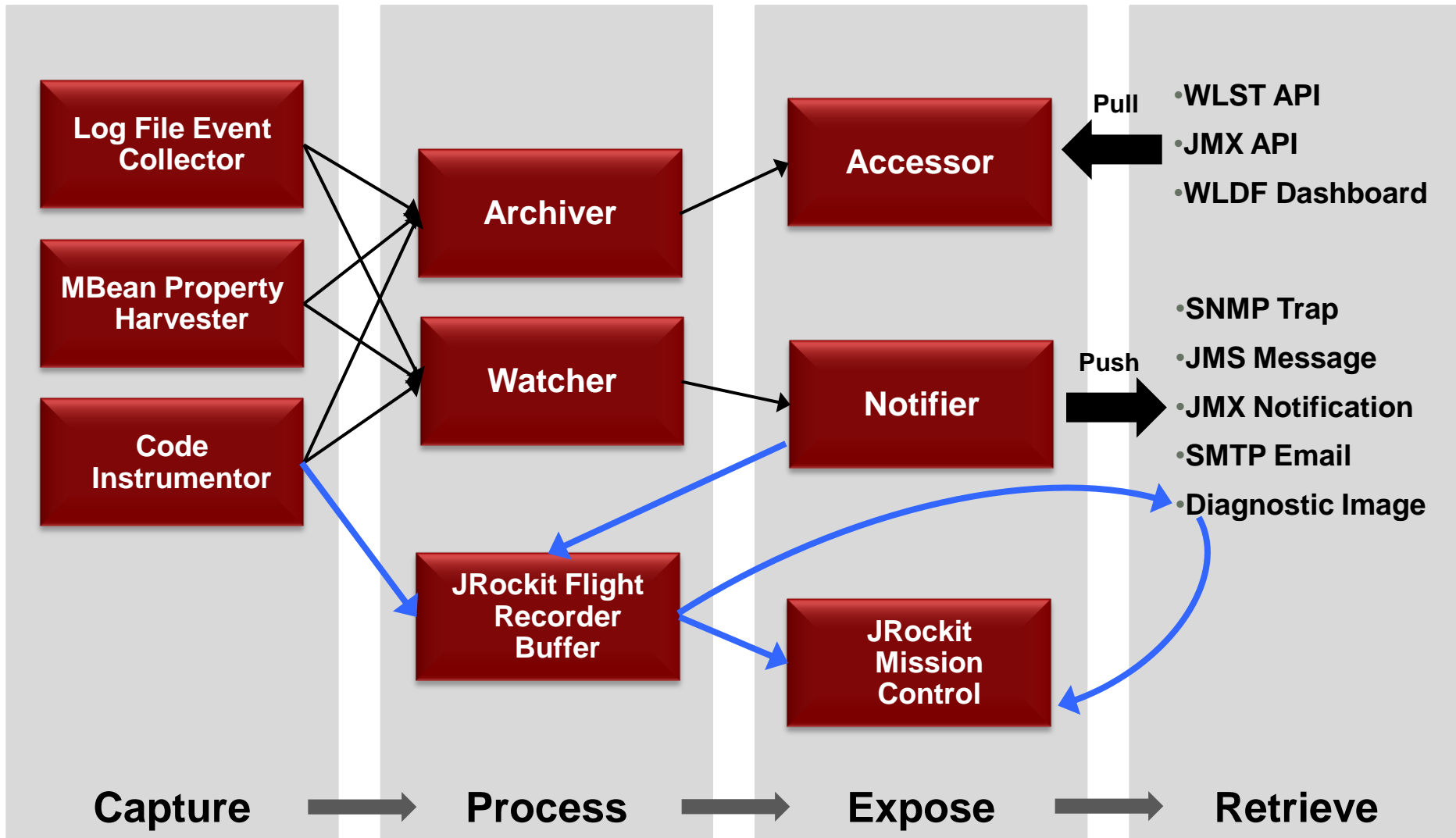
JRokit Flight Recorder & WebLogic Diagnostic Framework (WLDF) Integration

Jeffrey West

Application Grid Product Management



Overview of the WebLogic Diagnostics Framework

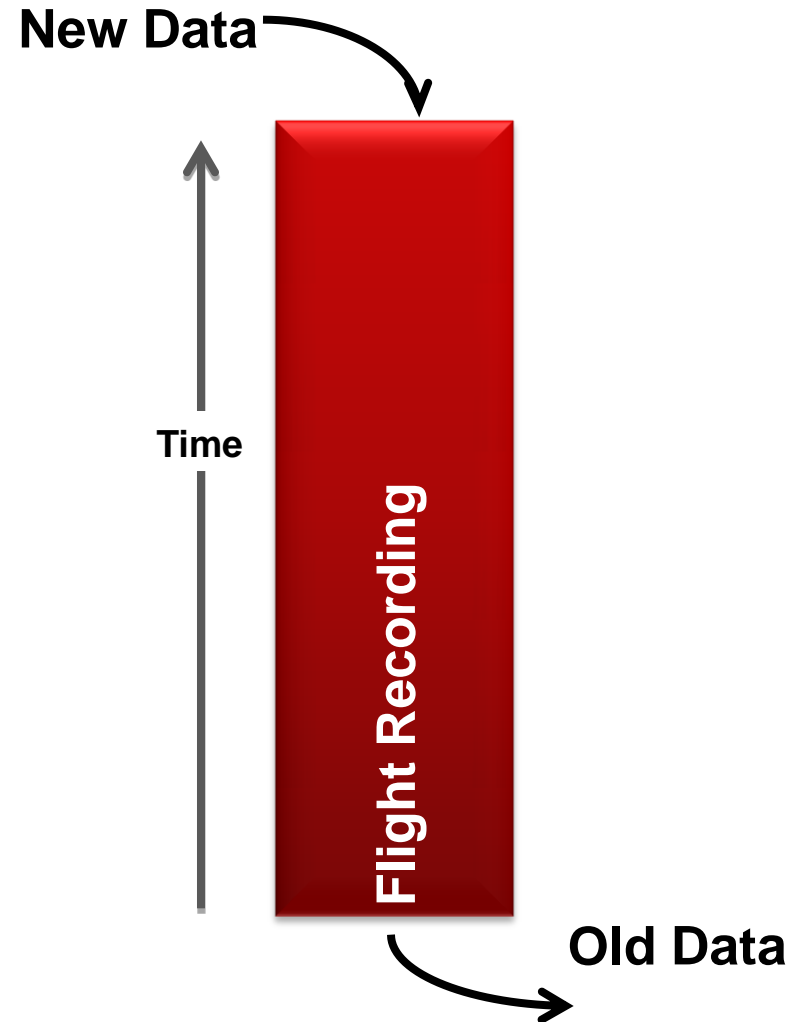


JRokit Flight Recorder Overview



What is the JRockit Flight Recorder?

- New in JRockit R28
- “Circular buffer” in JRockit JVM that stores diagnostic data
 - Always on
 - New data comes in and is stored, old data dropped off
- Built-in integration with JRMC
 - Replaces JRMC Runtime Analyzer and Latency Analyzer
- Very low/near zero overhead
 - Uses data already used by JVM
- Data can include events from the JVM and from any other event producer
 - WebLogic Server (WLDF)
 - Fusion Middleware (DMS)



JRokit Flight Recorder Use Cases

- What it is designed for?
 - Provide diagnostic information in running production systems
 - Look back in time to see what happened after a crash
 - Capture most recent activity to enable analysis leading up to an issue
 - Capture data from all levels JVM, WLS, DMS, etc...
 - Offline/offsite analysis can be done using the JRMC GUI
 - JRokit dumps capture information to assist in crash-analysis
- What it is not designed for?
 - Large event payloads or very high volumes of events
 - Long history
 - Not a replacement for Debug logging or the server logging

JRokit Flight Recorder + WLDF Integration



WebLogic/Flight Recorder Integration Goals and Points

3

- Leverage diagnostics in WebLogic Server
- Enhance usefulness of Flight Recorder



- WLDF Watch Rules and Notifications
- Remote Diagnostics Agent

2

- Enable easy configuration



- Available OOTB
- On/off/volume control
- Automatic JFR recording control

1

- View and correlate WebLogic and JVM events
- Make problem identification easy



- WebLogic JFR events
- Custom WebLogic event viewer in JRMC

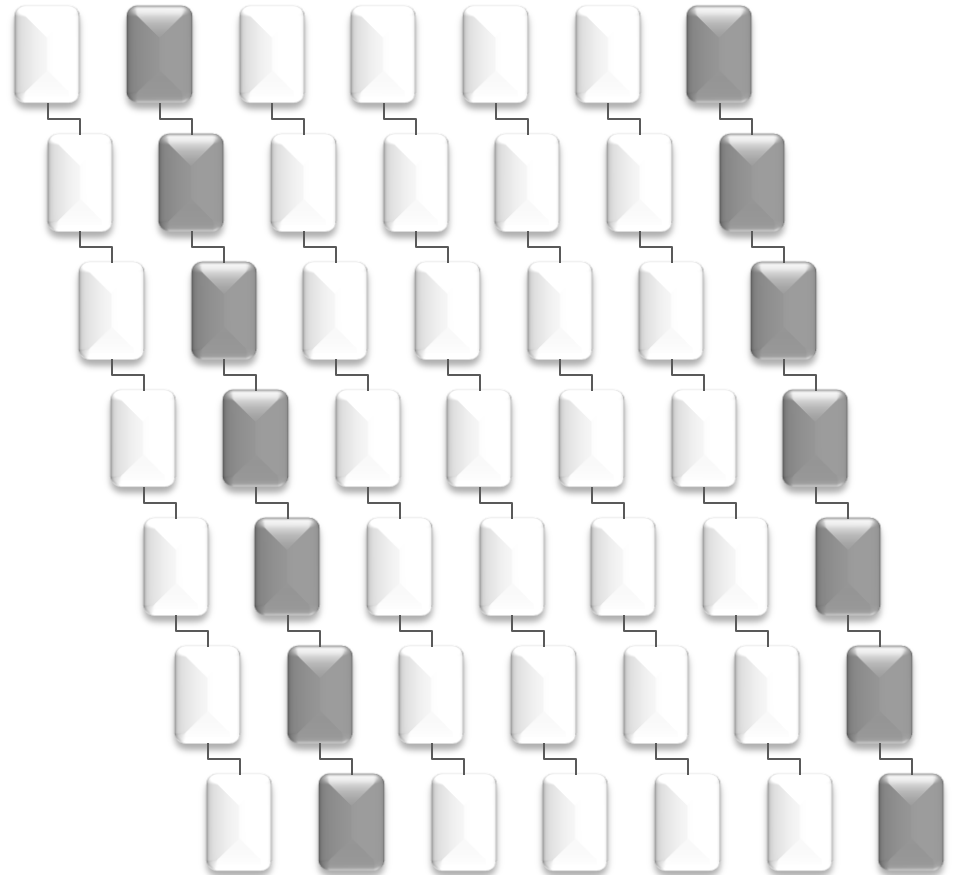
Integration Goals

Integration Points

WebLogic Diagnostic Events for Flight Recorder Integration

1

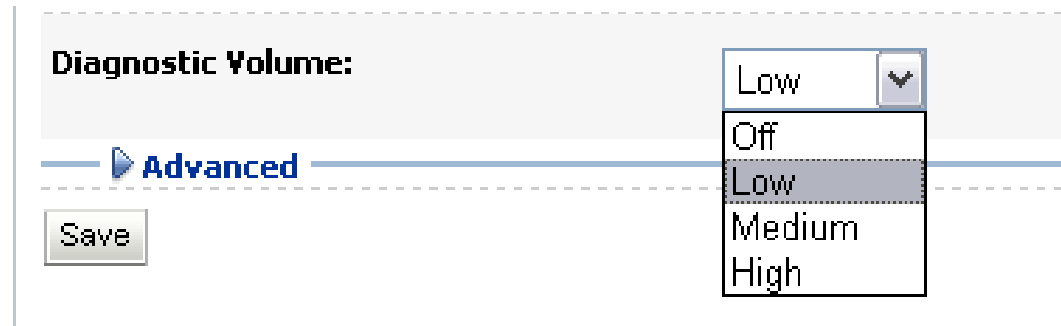
- Point-cuts woven into WebLogic to generate diagnostic events
 - Events are propagated to the Flight Recorder
- Logical event generators: low, medium, and high
- Very low performance impact
 - Low – 0.81 decrease in throughput
 - High – 3.83 decrease in throughput
- Events are throttled to avoid high overhead
 - Request is tagged for event generation
 - All events for a specific context ID are captured regardless of throttling
 - Target number of reqs/events per second, checked every 2 seconds, throttling readjusted



Easy Configuration with WLDF Volume

2

- Simple configuration: enabled and configured using a volume control per server: Off (default), Low, Medium, High
- Each level has a predetermined data set
 - Determined through consultation with dev groups and Support
 - Data set was adjusted based on performance testing
 - High volume includes user info and stack traces, which are expensive data gathering operations
- Volume also controls “WLDF Recording” in JFR
 - WLDF has it’s own recording independent of other “recordings” in JFR
 - At startup (WLDF volume = Off), WLDF recording created, but all events turned off
 - When WLDF volume set to Low or higher, WLDF events and default JVM events are generated
 - DMS leverages the same JFR recording



WLDF Low Volume Events

- WLDF Logging Snapshot
- WLDF LogRecord Snapshot
- WLDF WLogRecord Snapshot
- Connector Activate Endpoint
- Connector Deactivate Endpoint
- Connector Inbound Transaction Rollback
- Connector Outbound Connection Error
- Connector Outbound Destroy Connection
- Connector Outbound Register Resource
- Connector Outbound Release Connection
- Connector Outbound Reserve Connection
- Connector Outbound Transaction Rollback
- Connector Outbound Unregister Resource
- EJB Business Method Post Invoke
- EJB Business Method Pre Invoke
- EJB Pool Manager Post Invoke
- EJB Pool Manager Pre Invoke
- JDBC Connection Rollback
- JDBC Statement Execute
- Servlet Invocation
- Web Application Load
- Web Application Unload
- Webservices JAXRPC Client Request
- Webservices JAXRPC Client Response
- Webservices JAXRPC Dispatch
- Webservices JAXRPC Request
- Webservices JAXRPC Response
- Webservices JAXWS Endpoint
- Webservices JAXWS Request
- Webservices JAXWS Resource

WLDF Medium Volume Events

- EJB Home Create
- EJB Home Remove
- EJB PoolManager Create
- JDBC Connection Close
- JDBC Connection Commit
- JDBC Connection Create Statement
- JDBC Connection Prepare
- JDBC Connection Release
- JDBC Connection Reserve
- JDBC Data Source Get Connection
- JDBC Driver Connect
- JDBC Statement Creation
- Servlet Execute
- Servlet Request Run
- Servlet Request Dispatch
- Servlet Request
- Servlet Filter
- Servlet Async Action
- Servlet Context Execute
- Servlet Response Write Headers
- Servlet Response Send
- Servlet Stale Resource
- Servlet Check Access
- JMS BE Consumer Log

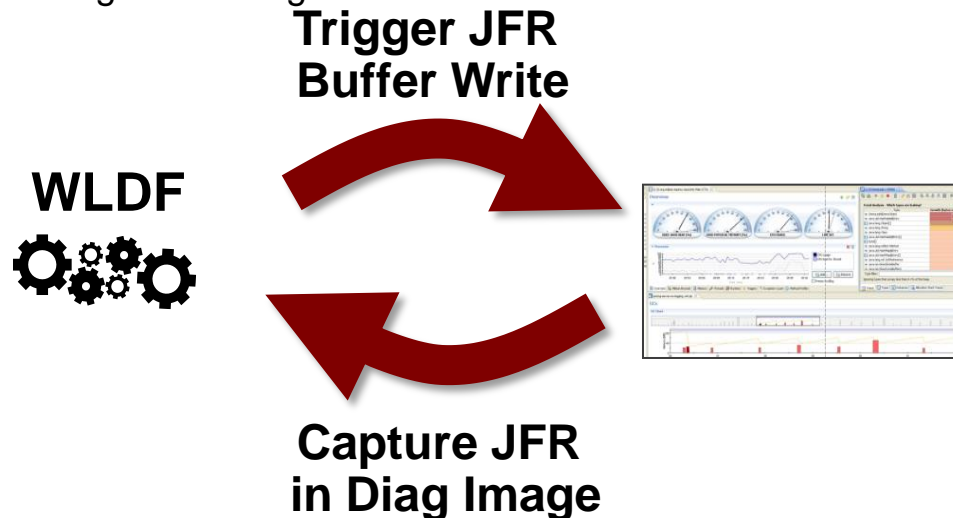
WLDF High Volume Events

- EJB Database Access
- EJB Business Method Post Invoke Cleanup
- EJB Pool Manager Remove
- EJB Replicated Session Manager
- EJB Timer Manager
- JDBC Transaction Commit
- JDBC Transaction End
- JDBC Transaction Get XA Resource
- JDBC Transaction Is Same RM
- JDBC Transaction Prepare
- JDBC Transaction Rollback
- JDBC Transaction Start
- JTA Transaction Commit
- JTA Transaction End
- JTA Transaction Prepared
- JTA Transaction Prepare
- JTA Transaction Start
- Servlet Request Overload
- Servlet Request Cancel
- Servlet Context Handle Throwable
- Plus stack traces and User IDs

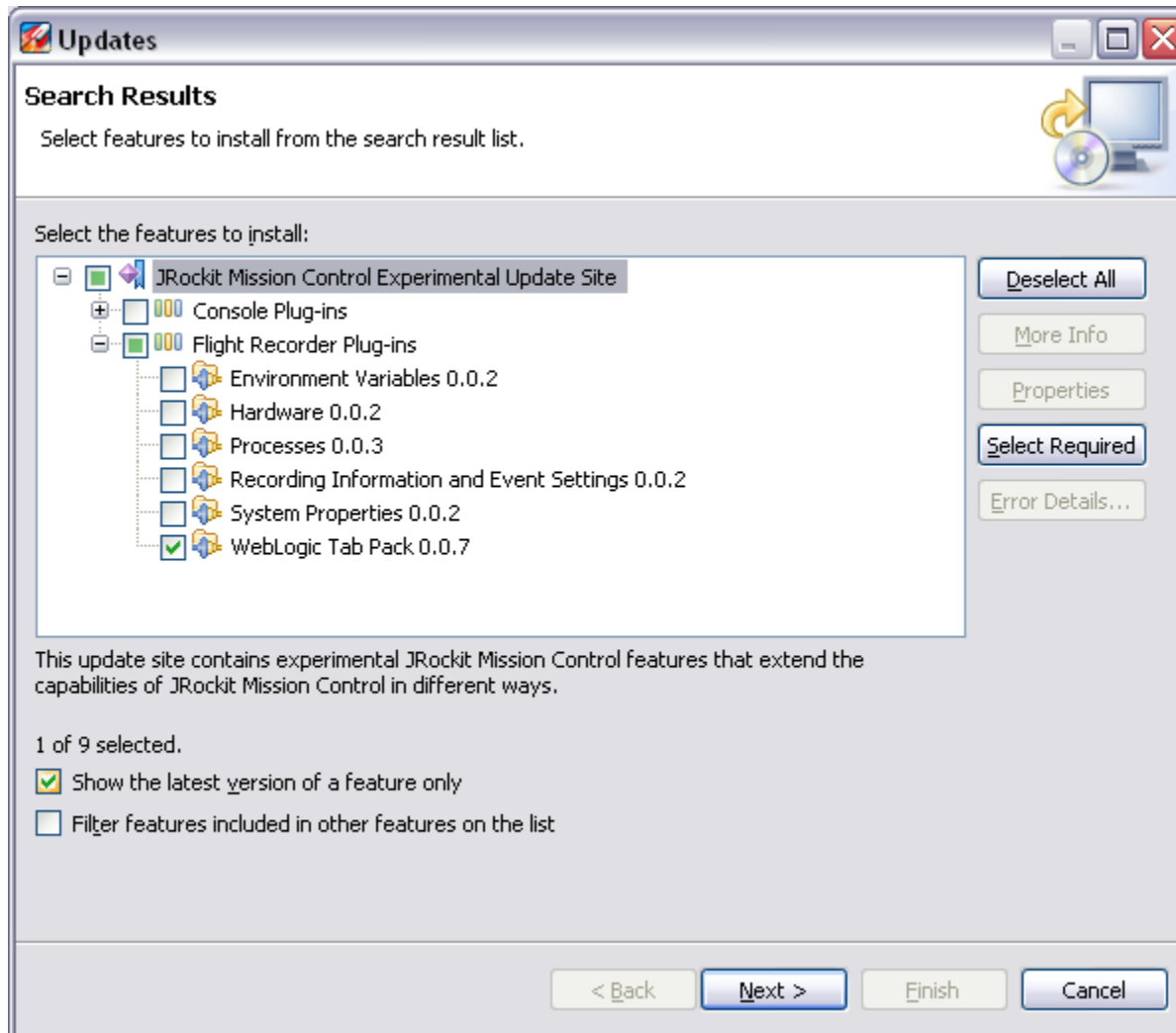
Automated Recording Capture

3

- Integrated with the WebLogic Diagnostic Image and Watch and Notification system
 - Enables capture based on system state, event – capture during event; no need to replicate
 - Watch for stuck thread count, heap size increase, available memory, etc.
 - Set up notifications: capture WLDF image
 - Diagnostic image capture spurs JFR file generation; JFR file included in diagnostic image
 - Includes full JFR data from all event generators
- Integration with Oracle Support's Remote Diagnostics Agent
 - Generates WL Diagnostic Image



WebLogic Experimental Plug-in for Mission Control 3



WebLogic Experimental Plugin

The screenshot displays the Oracle WebLogic Flight Recorder interface. The left sidebar contains navigation tabs: General, Memory, Code, CPU/Threads (highlighted with a red box), and Events. The main content area shows a recording session for 'flightRecording1133234013691345737.jfr'. A red dashed arrow points from the 'CPU/Threads' tab to the 'Diagnostic Volume' field in the 'General' section.

WebLogic Flight Recorder Tab Pack for Mission Control (Unsupported)
The events produced by the WebLogic Diagnostics Framework (WLDF) are throttled per ECID (Execution Context Identifier). If an ECID is picked out by the throttler you will see all WLDF-events generated for a request flow with that ECID. You can control the number of ECIDs that are collected by setting the Diagnostic Volume to Low, Medium or High. See [documentation](#) for how to configure and use WLDF with Oracle JRockit Flight Recorder.

General
Information about from where the data was collected and in what quantity.

Diagnostic Volume: N/A
Machine Name: N/A
Server Name: N/A
Domain Name: N/A

Throttling Information
A throttle rate below or equal to 1.0 indicates no throttling.

Current Period Throttle Rate	2
Last Period Throttle Rate	1
Last Period Events Generated per Request	24.071
Last Period Requests Seen	453.625
Last Period Requests Selected	180.625

Events by Type
Shows events reported by the WebLogic Diagnostics Framework (WLDF).

Filter Column: Event Type

Event Type	Count
JDBC Statement Execute	12,787
JDBC Connection Prepare	8,706
JDBC Statement Creation	8,706
EJB Business Method Pre Invoke	1,896
EJB Pool Manager Pre Invoke	1,896

Overview | ECID ECIDs | Users | Connector | Database | EJBs | JMS | JTA | Servlets | Web Services | HTTP Requests

Hardware and Software

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font with a registered trademark symbol, is centered within a solid red horizontal bar.

ORACLE®

Engineered to Work Together