OpenArm

1. Welcome to OpenArm

An open source implementation of the Open Groups' ARM (Application Response Measurement) 4.0 Java interfaces. OpenArm can be configured to send ARM transaction data to a log, to a messaging queue, a datastore, a JMX notification or to an SNMP trap.

What is ARM? The OpenGroup defines it as follows:

"The Application Response Measurement (ARM) standard describes a common method for integrating enterprise applications as manageable entities. The ARM standard allows users to extend their enterprise management tools directly to applications creating a comprehensive end-to-end management capability that includes measuring application availability, application performance, application usage, and end-to-end transaction response time."

"The ... Application Response Measurement (ARM) API function calls ... can be used to instrument an application or other software for transaction monitoring. The ARM API provides a way to monitor business transactions, by embedding simple calls in the software which can be captured by an agent supporting the ARM API. The calls are used to capture data that allows software to be monitored for availability, service levels, and capacity."

ARM is a standard, owned and managed by the OpenGroup. It defines essentially two elements:

1. the abstract API, used for instrumenting your application
2. the concept of, and very vague guidelines for implementing, an ARM agent

The first of these, the abstract API, is represented in ARM 4.0 for Java as a collection of Java interfaces. The interfaces themselves are only distributed by the Open Group in source code format. OpenArm includes these interfaces in compiled format as part of the binary distribution. OpenArm also contains concrete implementations of these interfaces, although it would be unusual, and potentially an error, for an application programmer to be aware of these classes or need to use them directly. Typically, one programs to the interfaces provided by the OpenGroup.

The second element, the ARM agent, is also provided by OpenArm. In effect, OpenArm turns your instrumented application into a lightweight agent. Via configuration, you direct OpenArm's agent functionality to delegate handling of ARM transaction events to one of

several possible "mediators" -- log4j, JMX, SNMP traps, a database, or JMS.