The Wayback Machine - https://web.archive.org/web/20060904234759/http://java.sun.com:80/j2se/1.5.0/docs/…

Chapter 1

Introduction to JMX Technology

Java Management Extensions Technology is a new feature in version 5.0 of the Java Platform, Standard Edition (J2SE). If you are already familiar with JMX technology, see Appendix A, "JMX Technology Versions" for version information.

JMX technology provides a simple, standard way of managing resources such as applications, devices, and services. Because JMX technology is dynamic, you can use it to monitor and manage resources as they are created, installed and implemented. You can also use JMX technology to monitor and manage the Java Virtual Machine[1] (JVM™).

JMX technology was developed through the Java Community Process℠ (JCP℠) as two closely related Java Specification Requests (JSRs):

- *JSR 3, Java Management Extensions Instrumentation and Agent Specification*
- *JSR 160, Java Management Extensions Remote API*

Each of the JSRs is defined by the API documentation generated by the JavadocTM tool, in conjunction with the PDF specification documents for each JSR. The Specifications and Reference Implementations can be downloaded at:

```
http://java.sun.com/products/JavaManagement/download.html
```

As its name indicates, the JMX Remote API adds remoting capability to the JMX specification, allowing you to monitor and manage applications, systems and networks remotely. In this document, the term “JMX technology” is used to cover both the JMX specification and the JMX Remote API.

This chapter introduces JMX technology in the following sections:

- "What Is JMX Technology?"
- "Why Use JMX Technology?"

What Is JMX Technology?

The JMX specification defines an architecture, the design patterns, the APIs, and the services for application and network management and monitoring in the Java programming language.

Using JMX technology, a given resource is instrumented by one or more Java objects known as Managed Beans, or MBeans. These MBeans are registered in a core managed object server, known as an MBean server, that acts as a management agent and can run on most devices enabled for the Java programming language.

The specifications define JMX agents that you use to manage resources instrumented in compliance with the specifications. A JMX agent consists of an MBean server, in which MBeans are registered, and a set of services for handling MBeans. In this way, JMX agents directly control resources and make them available to remote management applications.

The way in which resources are instrumented is completely independent from the management infrastructure. Resources can therefore be rendered manageable regardless of how their management applications are implemented.

JMX technology defines standard connectors (JMX connectors) that allow you to access JMX agents from remote management applications. JMX connectors using different protocols provide the same management interface. Hence a management application can manage resources transparently, regardless of the communication protocol used. JMX agents can also be used by systems or applications that are not compliant with the JMX specification but which support JMX agents.

Why Use JMX Technology?

JMX technology provides Java developers across all industries with a flexible means to instrument Java code, create smart Java agents, implement distributed management middleware and managers, and smoothly integrate these solutions into existing management and monitoring systems.

- *JMX technology enables Java applications to be managed without heavy investment:* A JMX technology agent can run on most Java technology-enabled devices, thus Java applications can become manageable with little impact on their design. A Java application simply needs to embed a managed object server and make some of its functionality available as one or several managed beans (MBeans) registered in the object server; that is all it takes to benefit from the management infrastructure.
- *JMX technology provides a standard way to manage Java technology-based applications, systems, and networks.* For example, the Java 2 Platform, Enterprise Edition (J2EE) 1.4 Application Server conforms to the JMX architecture and consequently can be managed using JMX technology.
- *JMX technology can be used for out-of-the-box management of the JVM.* The Java Virtual Machine (JVM) is highly instrumented using JMX technology. You can easily start a JMX agent to access the built-in JVM instrumentation, and thereby monitor and manage the JVM remotely.
- *JMX technology provides a scalable, dynamic management architecture.* Every JMX agent service is an independent module that can be plugged into the management agent, depending on the requirements. This component-based approach means that JMX solutions can scale from small-footprint devices to large telecommunications switches and beyond. The JMX specification provides a set of core agent services. Additional services can be developed and dynamically loaded, unloaded, or updated in the management infrastructure.
- *JMX technology leverages existing standard Java technologies.* Whenever needed, the JMX specification references existing Java specifications, for example, the Java Naming and Directory Interface (JNDI).
- *JMX technology integrates easily with existing management solutions and emerging technologies.* For example, JMX agents could be managed through an HTML browser. The JMX APIs are open interfaces that any management system vendor can leverage. JMX solutions can use lookup and discovery services and protocols such as Jini™ network technology and the Service Location Protocol (SLP).

[1] The terms “Java Virtual Machine” and “JVM” mean a virtual machine for the Java platform.

Copyright © 2004 Sun Microsystems, Inc. All rights reserved.