# JMX快速入门学习指南

## 1、简介

JMX（Java Management Extensions）从JDK1.5开始引入，它提供了一种可以在Java程序运行过程中动态管理程序资源的机制。JMX通常用于企业级系统中，让系统运行时也可配置，或者在任意时间点获得系统关键的状态数据。

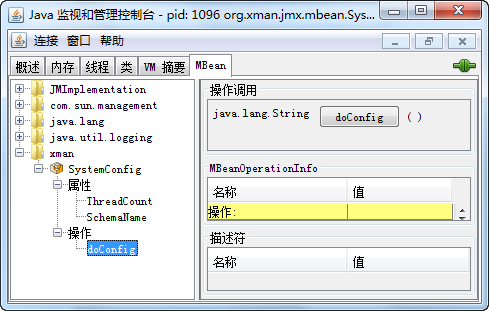
要想使用JMX去管理资源，我们需要创建Mbeans（Managed Beans），然后把它注册到一个Mbean Server中，这个Mbean Server就是注册的Mbeans的管理引擎。

## 2、MBean接口类

使用JMX的第一步就是创建一个接口，且这个**接口的名称必须以MBean结尾**。MBean接口中的方法，就是我们要暴露给外部去操作或控制内部资源的方法。如果想让数据只读，不提供对应的setter方法即可。

|  |
| --- |
| **package** org.xman.jmx.mbean;  **public** **interface** SystemConfigMBean {  **public** **void** **setThreadCount**(**int** noOfThreads);  **public** **int** **getThreadCount**();  **public** **void** **setSchemaName**(String schemaName);  **public** String **getSchemaName**();  // any method starting with get and set are considered  // as attributes getter and setter methods, so I am  // using do\* for operation.  **public** String **doConfig**();  } |

该MBean暴露后，在jconsole中将看到下面的结果。



## 3、MBean实现类

对MBean接口实现类的命名，一种比较方便易行的方法就是把原接口中的MBean几个字母去掉。

下面这个实现类中定义了两个属性threadCount，schemaName，即可配置的线程数与模式名称。

|  |
| --- |
| **package** org.xman.jmx.mbean;  **public** **class** SystemConfig **implements** SystemConfigMBean {  **private** **int** threadCount;  **private** String schemaName;  **public** **SystemConfig**(**int** numThreads, String schema) {  **this**.threadCount = numThreads;  **this**.schemaName = schema;  }  @Override  **public** **void** **setThreadCount**(**int** noOfThreads) {  **this**.threadCount = noOfThreads;  }  @Override  **public** **int** **getThreadCount**() {  **return** **this**.threadCount;  }  @Override  **public** **void** **setSchemaName**(String schemaName) {  **this**.schemaName = schemaName;  }  @Override  **public** String **getSchemaName**() {  **return** **this**.schemaName;  }  @Override  **public** String **doConfig**() {  **return** "No of Threads=" + **this**.threadCount + " and DB Schema Name=" + **this**.schemaName;  }  } |

## 4、MBean注册

下面将演示如果将MBean注册到MBeanServer中去。

注册的操作通常都是在主线程去做的。下面的注册程序将线程数设置为10、模式名称设置为default，并每3秒循环输出这两个数据，直到检测到线程为0时退出程序。

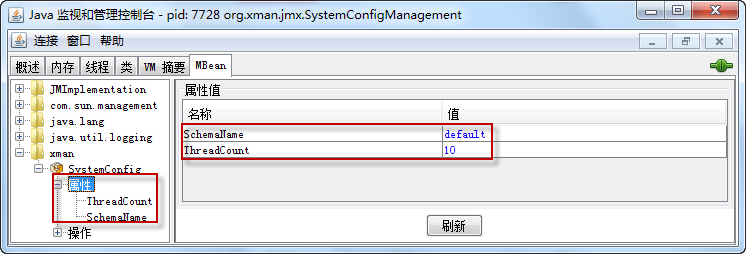
|  |
| --- |
| **package** org.xman.jmx.mbean;  **import** java.lang.management.ManagementFactory;  **import** javax.management.MBeanServer;  **import** javax.management.ObjectName;  **public** **class** SystemConfigManagement {  **private** **static** **final** **int** DEFAULT\_NO\_THREADS = 10;  **private** **static** **final** String DEFAULT\_SCHEMA = "default";  **public** **static** **void** **main**(String[] args) **throws** Exception {  // Get the MBean server  MBeanServer mbs = ManagementFactory.*getPlatformMBeanServer*();  // register the MBean  SystemConfig mBean = **new** SystemConfig(DEFAULT\_NO\_THREADS, DEFAULT\_SCHEMA);  ObjectName name = **new** ObjectName("xman:type=SystemConfig");  mbs.registerMBean(mBean, name);  **do** {  Thread.*sleep*(3000);  System.out.println("Thread Count=" + mBean.getThreadCount() + ":::Schema Name=" + mBean.getSchemaName());  } **while** (mBean.getThreadCount() != 0);  }  } |

## 5、JConsole测试

主程序SystemConfigManagement运行后，我们打开jconsole，将看到类似下面的界面：



我们选择jmx字样并连接，然后在MBean标签页找到我们注册的类。



可以直接修改蓝色的值，修改后主程序中的输出将进行改变。

|  |
| --- |
| Thread Count=100:::Schema Name=default  Thread Count=100:::Schema Name=jmx  Thread Count=100:::Schema Name=jmx  Thread Count=100:::Schema Name=jmx |

## 6、远程连接

如果想远程连接上主程序，**前提是必须在主程序启动时加上jmx的可远程连接配置**，包括主机IP、端口、授权等，示例如下所示：

|  |
| --- |
| -Djava.rmi.server.hostname=192.168.0.88  -Dcom.sun.management.jmxremote  -Dcom.sun.management.jmxremote.port=8888  -Dcom.sun.management.jmxremote.authenticate=false  -Dcom.sun.management.jmxremote.ssl=false |

### 6.1、Jconsole中远程连接



### 6.2、Java代码中远程连接

通过Java代码来远程连接，则需要用到JMXServiceURL、JMXConnector、MBeanServerConnection等对象，具体用法看下面的代码一目了然。

|  |
| --- |
| **package** org.xman.jmx.mbean;  **import** javax.management.JMX;  **import** javax.management.MBeanServerConnection;  **import** javax.management.ObjectName;  **import** javax.management.remote.JMXConnector;  **import** javax.management.remote.JMXConnectorFactory;  **import** javax.management.remote.JMXServiceURL;  **public** **class** Client {  /\*\*  \* **@param** args  \* **@throws** Exception  \*/  **public** **static** **void** **main**(String[] args) **throws** Exception {  JMXServiceURL url = **new** JMXServiceURL("service:jmx:rmi:///jndi/rmi://127.0.0.1:8888/jmxrmi");  JMXConnector jmxc = JMXConnectorFactory.*connect*(url, **null**);    MBeanServerConnection mbsc = jmxc.getMBeanServerConnection();    ObjectName mbeanName = **new** ObjectName("xman:type=SystemConfig");    SystemConfigMBean mbeanProxy = JMX.*newMBeanProxy*(mbsc, mbeanName, SystemConfigMBean.**class**, **true**);  **int** threadCount = mbeanProxy.getThreadCount();  System.out.println("Current ThreadCount: " + threadCount);  mbeanProxy.setThreadCount(100);  threadCount = mbeanProxy.getThreadCount();  System.out.println("Changed ThreadCount: " + threadCount);    Thread.*sleep*(3);  jmxc.close();  }  } |

客户端程序运行后，可看到主程序输出数据已变化。

## 7、问题

### 7.1、一个MBean接口只能有一个实现？

是的，一个Mbean接口只会有一个实现才有效，这个实现类的类名为接口名去掉“Mbean”。