quartz

[编辑](javascript:;)

Quartz是OpenSymphony开源组织在Job scheduling领域又一个开源项目，它可以与J2EE与J2SE应用程序相结合也可以单独使用。Quartz可以用来创建简单或为运行十个，百个，甚至是好几万个Jobs这样复杂的程序。Jobs可以做成标准的Java组件或 EJBs。Quartz的最新版本为Quartz 2.2.1。

调度简单[编辑](javascript:;)

Quartz是一个完全由java编写的开源作业调度框架。不要让作业调度这个术语吓着你。尽管Quartz框架整合了许多额外功能， 但就其简易形式看，你会发现它易用得简直让人受不了！。简单地创建一个实现org.quartz.Job接口的java类。Job接口包含唯一的方法：

public void execute(JobExecutionContext context)

throws JobExecutionException;

在你的Job接口实现类里面，添加一些逻辑到execute()方法。一旦你配置好Job实现类并设定好调度时间表，Quartz将密切注意剩余时间。当调度程序确定该是通知你的作业的时候，Quartz框架将调用你Job实现类（作业类）上的execute()方法并允许做它该做的事情。无需报告任何东西给调度器或调用任何特定的东西。仅仅执行任务和结束任务即可。如果配置你的作业在随后再次被调用，Quartz框架将在恰当的时间再次调用它。

内部架构[编辑](javascript:;)

在规模方面，Quartz跟大多数开源框架类似。大约有300个java类和接口，并被组织到12个包中。这可以和Apache Struts把大约325个类和接口以及组织到11个包中相比。尽管规模几乎不会用来作为衡量框架质量的一个特性，但这里的关键是quartz内含很多功能，这些功能和特性集是否成为、或者应该成为评判一个开源或非开源框架质量的因素。

调度器[编辑](javascript:;)

Quartz框架的核心是调度器。调度器负责管理Quartz应用运行时环境。调度器不是靠自己做所有的工作，而是依赖框架内一些非常重要的部件。Quartz不仅仅是线程和线程管理。为确保可伸缩性，Quartz采用了基于多线程的架构。启动时，框架初始化一套worker线程，这套线程被调度器用来执行预定的作业。这就是Quartz怎样能[并发运行](http://baike.baidu.com/view/3048188.htm)多个作业的原理。Quartz依赖一套松耦合的[线程池](http://baike.baidu.com/view/262150.htm)管理部件来管理线程环境。本篇文章中，我们会多次提到线程池管理，但Quartz里面的每个对象是可配置的或者是可定制的。所以，例如，如果你想要插进自己线程池管理设施，我猜你一定能！

框架特征[编辑](javascript:;)

Quartz框架有一个丰富的特征集。事实上，Quartz有太多特性以致不能在一种情况中全部领会，下面列出了一些有意思的特征，但没时间在此详细讨论。

**监听器和插件**

每个人都喜欢监听和[插件](http://baike.baidu.com/view/18979.htm)。今天，几乎下载任何开源框架，你必定会发现支持这两个概念。监听是你创建的java类，当关键事件发生时会收到框架的回调。例如，当一个作业被调度、没有调度或触发器终止和不再触发时，这些都可以通过设置来通知你的监听器。Quartz框架包含了调度器监听、作业和触发器监听。你可以配置作业和触发器监听为全局监听或者是特定于作业和触发器的监听。

一旦你的一个具体监听被调用，你就能使用这个技术来做一些你想要在监听类里面做的事情。例如，你如果想要在每次作业完成时发送一个电子邮件，你可以将这个逻辑写进作业里面，也可以写进JobListener里面。写进JobListener的方式强制使用[松耦合](http://baike.baidu.com/view/1343493.htm)有利于设计上做到更好。

Quartz插件是一个新的功能特性，无须修改Quartz源码便可被创建和添加进Quartz框架。他为想要扩展Quartz框架又没有时间提交改变给Quartz开发团队和等待新版本的开发人员而设计。如果你熟悉Struts[插件](http://baike.baidu.com/view/18979.htm)的话，那么完全可以理解Quartz插件的使用。

与其Quartz提供一个不能满足你需要的有限扩展点，还不如通过使用插件来拥有可修整的扩展点。

**集群Quartz应用**

Quartz应用能被[集群](http://baike.baidu.com/view/302477.htm)，是水平集群还是垂直集群取决于你自己的需要。集群提供以下好处：

·伸缩性

·高可用性

·负载均衡

Quartz可以借助关系数据库和JDBC作业存储支持集群。

Terracotta扩展quartz提供集群功能而不需要数据库支持

**相关工具**

Quartz经常会用到cron表达式，可以使用国外网站cronmaker辅助生成cron表达式。