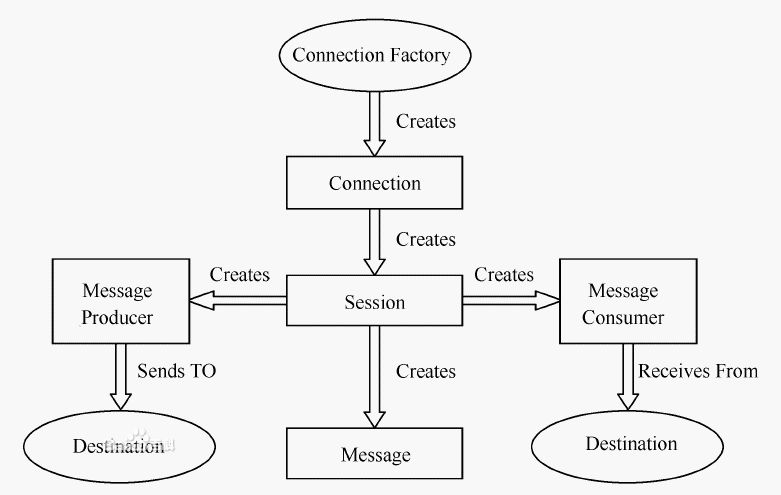
参考：<http://blog.csdn.net/caiwenfeng_for_23/article/details/8606521>

[JMS简介](http://blog.csdn.net/caiwenfeng_for_23/article/details/8606521)

**jms**即**[Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "_blank)**消息服务（Java Message Service）应用程序接口，是一个Java平台中关于面向消息中间件（**MOM**）的API，用于在两个应用程序之间，或分布式系统中发送消息，进行异步通信（关键词：应用程序接口 应用程序之间通信的接口 面向消息中间件-MOM的结束规范）。Java消息服务是一个与具体平台无关的API，绝大多数**MOM提供商**都对JMS提供支持。

JMS（Java Messaging Service）是Java平台上有关面向消息中间件(MOM)的**技术规范**，它便于消息系统中的Java应用程序进行消息交换,并且通过提供标准的产生、发送、接收消息的接口简化企业应用的开发，翻译为Java消息服务。



**JMS有以下元素组成。**

**JMS提供者**

连接面向消息中间件的，JMS接口的一个实现。提供者可以是Java平台的JMS实现，也可以是非Java平台的面向消息中间件的适配器。

**JMS客户（是生产者和消费者的统称）**

生产或消费基于消息的Java的应用程序或对象。

**JMS生产者**

创建并发送消息的JMS客户。

**JMS消费者**

接收消息的JMS客户。

**JMS消息**

包括可以在JMS客户之间传递的数据的对象

**JMS队列**

一个容纳那些被发送的等待阅读的消息的区域。队列暗示，这些消息将按照顺序发送。一旦一个消息被阅读，该消息将被从队列中移走。

**JMS主题 （Topics）**

一种支持发送消息给多个订阅者的机制。

**JMS对象模型**

JMS对象模型包含如下几个要素：[1]

**1）连接工厂。**连接工厂（ConnectionFactory）是由管理员创建，并绑定到JNDI树中。客户端使用JNDI查找连接工厂，然后利用连接工厂创建一个JMS连接。

**2）JMS连接。**JMS连接（Connection）表示JMS客户端和服务器端之间的一个活动的连接，是由客户端通过调用连接工厂的方法建立的。

**3）JMS会话。**JMS会话（Session）表示JMS客户与JMS服务器之间的会话状态。JMS会话建立在JMS连接上，表示客户与服务器之间的一个会话线程。

**4）JMS目的。**JMS目的（Destination），又称为消息队列，是实际的消息源。

**5）JMS生产者和消费者。**生产者（Message Producer）和消费者（Message Consumer）对象由Session对象创建，用于发送和接收消息。

6）**JMS消息通常有两种类型（P2P和Pub/Sub）：**

① 点对点（Point-to-Point）。在点对点的消息系统中，消息分发给一个单独（p->p）的使用者。点对点消息往往与队列（javax.jms.Queue）相关联。

② 发布/订阅（Publish/Subscribe）。发布/订阅消息系统支持一个事件驱动模型，消息生产者和消费者都参与消息的传递。生产者发布事件，而使用者订阅感兴趣的事件，并使用事件。该类型消息一般与特定的主题（javax.jms.Topic）关联。

**JMS模型**

Java消息服务应用程序结构支持两种模型：

**点对点或队列模型**

**发布者/订阅者模型**

在**点对点或队列模型**下，一个生产者向一个特定的队列发布消息，一个消费者从该队列中读取消息。这里，生产者知道消费者的队列，并直接将消息发送到消费者的队列。这种模式被概括为：

只有一个消费者将获得消息

生产者不需要在接收者消费该消息期间处于运行状态，接收者也同样不需要在消息发送时处于运行状态。

每一个成功处理的消息都由接收者签收

**发布者/订阅者模型**支持向一个特定的消息主题发布消息。0或多个订阅者可能对接收来自特定消息主题的消息感兴趣。在这种模型下，发布者和订阅者彼此不知道对方。这种模式好比是匿名公告板。

这种模式被概括为：

1、多个消费者可以获得消息（对比于p2p的点对点，一个生产者和一个消费者）

2、在发布者和订阅者之间存在时间依赖性（对比于生产者和消费者的无时间依赖性，生产者不需要等待消费者上线，消费者也不需要等待生产者上线）。发布者需要建立一个订阅（subscription），以便客户能够购订阅。订阅者必须保持持续的活动状态以接收消息，除非订阅者建立了持久的订阅。在那种情况下，在订阅者未连接时发布的消息将在订阅者重新连接时重新发布。

使用Java语言，JMS提供了将应用与提供数据的传输层相分离的方式。同一组Java类可以通过JNDI中关于提供者的信息，连接不同的JMS提供者。这一组类首先使用一个连接工厂以连接到队列或主题，然后发送或发布消息。在接收端，客户接收或订阅这些消息。

**传递消息方式**

JMS现在有两种传递消息的方式。标记为**NON\_PERSISTENT**的消息最多投递一次，而标记为**PERSISTENT**的消息将使用暂存后再转送的机理投递。如果一个JMS服务离线，那么持久性消息不会丢失但是得等到这个服务恢复联机时才会被传递。所以默认的消息传递方式是非持久性的。即使使用非持久性消息可能降低内存和需要的存储器，并且这种传递方式只有当你不需要接收所有的消息时才使用。

虽然JMS规范并不需要JMS供应商实现消息的**优先级**路线，但是它需要递送加快的消息优先于普通级别的消息。JMS定义了从0到9的优先级路线级别，0是最低的优先级而9则是最高的。更特殊的是0到4是正常优先级的变化幅度，而5到9是加快的优先级的变化幅度。举例来说： topicPublisher.publish (message, DeliveryMode.PERSISTENT, 8, 10000); //Pub-Sub 或 queueSender.send(message,DeliveryMode.PERSISTENT, 8, 10000);//P2P 　这个代码片断，有两种消息模型，映射递送方式是持久的，优先级为加快型，生存周期是10000 (以毫秒度量)。如果生存周期设置为零，这则消息将永远不会过期。当消息需要时间限制否则将使其无效时，设置生存周期是有用的。

JMS定义了五种不同的消息正文格式，以及调用的消息类型，允许你发送并接收以一些不同形式的数据，提供现有消息格式的一些级别的兼容性。

· StreamMessage -- Java原始值的数据流

· MapMessage--一套名称-值对

· TextMessage--一个字符串对象

· ObjectMessage--一个序列化的 Java对象

· BytesMessage--一个未解释字节的数据流

**JMS应用程序接口**

**ConnectionFactory 接口（连接工厂）**

用户用来创建到JMS提供者的连接的被管对象。JMS客户通过可移植的接口访问连接，这样当下层的实现改变时，代码不需要进行修改。 管理员在JNDI名字空间中配置连接工厂，这样，JMS客户才能够查找到它们。根据消息类型的不同，用户将使用队列（p2p）连接工厂，或者主题(pub/sub)连接工厂。

**Connection 接口（连接）**

连接代表了应用程序和消息服务器之间的通信链路。在获得了连接工厂后，就可以创建一个与JMS提供者的连接。根据不同的连接类型，连接允许用户创建会话，以发送和接收队列和主题到目标。

**Destination 接口（目标）**

目标是一个包装了消息目标标识符的被管对象，消息目标是指消息发布和接收的地点，或者是队列，或者是主题。JMS管理员创建这些对象，然后用户通过JNDI发现它们。和连接工厂一样，管理员可以创建两种类型的目标，点对点模型的队列，以及发布者/订阅者模型的主题。

**MessageConsumer 接口（消息消费者）**

由会话创建的对象，用于接收发送到目标的消息。消费者可以同步地（阻塞模式），或异步（非阻塞）接收队列和主题类型的消息。

**MessageProducer 接口（消息生产者）**

由会话创建的对象，用于发送消息到目标。用户可以创建某个目标的发送者，也可以创建一个通用的发送者，在发送消息时指定目标。

**Message 接口（消息）**

是在消费者和生产者之间传送的对象，也就是说从一个应用程序传送到另一个应用程序。一个消息有三个主要部分：

消息头（必须）：包含用于识别和为消息寻找路由的操作设置。

一组消息属性（可选）：包含额外的属性，支持其他提供者和用户的兼容。可以创建定制的字段和过滤器（消息选择器）。

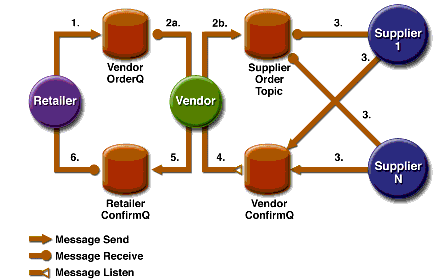
一个消息体（可选）：允许用户创建五种类型的消息（文本消息，映射消息，字节消息，流消息和对象消息）。

消息接口非常灵活，并提供了许多方式来定制消息的内容。

**Session 接口（会话）**

表示一个单线程的上下文，用于发送和接收消息。由于会话是单线程的，所以消息是连续的，就是说消息是按照发送的顺序一个一个接收的。会话的好处是它支持事务。如果用户选择了事务支持，会话上下文将保存一组消息，直到事务被提交才发送这些消息。在提交事务之前，用户可以使用回滚操作取消这些消息。一个会话允许用户创建消息生产者来发送消息，创建消息消费者来接收消息。

**JMS提供者实现**

****

  JMS流程图

要使用Java消息服务，你必须要有一个JMS提供者，管理会话和队列。现在既有开源的提供者也有专有的提供者。

开源的提供者包括：

**Apache ActiveMQ**

JBoss 社区所研发的 HornetQ

Joram

Coridan的MantaRay

The OpenJMS Group的OpenJMS

专有的提供者包括：

BEA的BEA WebLogic Server JMS

TIBCO Software的EMS

GigaSpaces Technologies的GigaSpaces

Softwired 2006的iBus

IONA Technologies的IONA JMS

SeeBeyond的IQManager（2005年8月被Sun Microsystems并购）

webMethods的JMS+ -

my-channels的Nirvana

Sonic Software的SonicMQ

SwiftMQ的SwiftMQ

IBM的WebSphere MQ