<http://blog.csdn.net/zhongguoren666/article/details/8267495>

MQ的基本概念：

1) 队列管理器

队列管理器是MQ系统中最上层的一个概念，由它为我们提供基于队列的消息服务。

2) 消息

在MQ中，我们把应用程序交由MQ传输的数据定义为消息，我们可以定义消息的内容并对消息进行广义的理解，比如：用户的各种类型的数据文件，某个应用向其它应用发出的处理请求等都可以作为消息。消息有两部分组成：

消息描述符(Message Discription或Message Header)，描述消息的特征，如：消息的优先级、生命周期、消息Id等；

消息体(Message Body)，即用户数据部分。在MQ中，消息分为两种类型，非永久性(non-persistent)消息和永久性(persistent)消息，非永久性消息是存储在内存中的，它是为了提高性能而设计的，当系统掉电或MQ队列管理器重新启动时，将不可恢复。当用户对消息的可靠性要求不高，而侧重系统的性能表现时，可以采用该种类型的消息，如：当发布股票信息时，由于股票信息是不断更新的，我们可能每若干秒就会发布一次，新的消息会不断覆盖旧的消息。永久性消息是存储在硬盘上，并且纪录数据日志的，它具有高可靠性，在网络和系统发生故障等情况下都能确保消息不丢、不重。

此外，在MQ中，还有逻辑消息和物理消息的概念。利用逻辑消息和物理消息，我们可以将大消息进行分段处理，也可以将若干个本身完整的消息在应用逻辑上归为一组进行处理。

3) 队列

队列是消息的安全存放地，队列存储消息直到它被应用程序处理。

消息队列以下述方式工作：

a) 程序A形成对消息队列系统的调用，此调用告知消息队列系统，消息准备好了投向程序B；

b) 消息队列系统发送此消息到程序B驻留处的系统，并将它放到程序B的队列中；

c) 适当时间后，程序B从它的队列中读此消息，并处理此信息。

由于采用了先进的程序设计思想以及内部工作机制，MQ能够在各种网络条件下保证消息的可靠传递，可以克服网络线路质量差或不稳定的现状，在传输过程中，如果通信线路出现故障或远端的主机发生故障，本地的应用程序都不会受到影响，可以继续发送数据，而无需等待网络故障恢复或远端主机正常后再重新运行。

在MQ中，队列分为很多种类型，其中包括：本地队列、远程队列、模板队列、动态队列、别名队列等。

本地队列又分为普通本地队列和传输队列，普通本地队列是应用程序通过API对其进行读写操作的队列；传输队列可以理解为存储-转发队列，比如：我们将某个消息交给MQ系统发送到远程主机，而此时网络发生故障，MQ将把消息放在传输队列中暂存，当网络恢复时，再发往远端目的地。

远程队列是目的队列在本地的定义，它类似一个地址指针，指向远程主机上的某个目的队列，它仅仅是个定义，不真正占用磁盘存储空间。

模板队列和动态队列是MQ的一个特色，它的一个典型用途是用作系统的可扩展性考虑。我们可以创建一个模板队列，当今后需要新增队列时，每打开一个模板队列，MQ便会自动生成一个动态队列，我们还可以指定该动态队列为临时队列或者是永久队列，若为临时队列我们可以在关闭它的同时将它删除，相反，若为永久队列，我们可以将它永久保留，为我所用。

4) 通道

通道是MQ系统中队列管理器之间传递消息的管道，它是建立在物理的网络连接之上的一个逻辑概念，也是MQ产品的精华。

在MQ中，主要有三大类通道类型，即消息通道，MQI通道和Cluster通道。消息通道是用于在MQ的服务器和服务器之间传输消息的，需要强调指出的是，该通道是单向的，它又有发送(sender), 接收(receive), 请求者(requestor), 服务者(server)等不同类型，供用户在不同情况下使用。MQI通道是MQ Client和MQ Server之间通讯和传输消息用的，与消息通道不同，它的传输是双向的。群集(Cluster)通道是位于同一个MQ 群集内部的队列管理器之间通讯使用的。

--------------------------------------------------------------------------------  
MQ的工作原理

如图所示：

首先来看本地通讯的情况，应用程序A和应用程序B运行于同一系统A，它们之间可以借助消息队列技术进行彼此的通讯：应用程序A向队列1发送一条信息，而当应用程序B需要时就可以得到该信息。

其次是远程通讯的情况，如果信息传输的目标改为在系统B上的应用程序C，这种变化不会对应用程序A产生影响，应用程序A向队列2发送一条信息，系统A的MQ发现Q2所指向的目的队列实际上位于系统B，它将信息放到本地的一个特殊队列－传输队列(Transmission Queue)。我们建立一条从系统A到系统B的消息通道，消息通道代理将从传输队列中读取消息，并传递这条信息到系统B，然后等待确认。只有MQ接到系统B成功收到信息的确认之后，它才从传输队列中真正将该信息删除。如果通讯线路不通，或系统B不在运行，信息会留在传输队列中，直到被成功地传送到目的地。这是MQ最基本而最重要的技术--确保信息传输，并且是一次且仅一次(once-and-only-once)的传递。

MQ提供了用于应用集成的松耦合的连接方法，因为共享信息的应用不需要知道彼此物理位置（网络地址）；不需要知道彼此间怎样建立通信；不需要同时处于运行状态；不需要在同样的操作系统或网络环境下运行。

--------------------------------------------------------------------------------  
MQ的基本配置举例

在上图中，要实现网络上两台主机上的通讯，若采用点对点的通讯方式，我们至少要建立如下MQ的对象：

在发送方A:

1） 建立队列管理器QMA: crtmqm -q QMA

2） 定义本地传输队列: define qlocal (QMB) usage (xmitq) defpsist(yes)

3） 创建远程队列: define qremote (QR.TOB) rname (LQB) rqmname (QMB) xmitq (QMB)

4） 定义发送通道: define channel (A.TO.B) chltype (sdr) conname ('IP of B') xmitq (QMB) + trptype (tcp)

在接收方B：

1） 建立队列管理器QMB: crtmqm -q QMB

2） 定义本地队列QLB: define qlocal (LQB)

3） 创建接收通道: define channel (A.TO.B) chltype (rcvr) trptype (tcp)

经过上述配置，我们就可以实现从主机A到B的单向通讯，若要实现二者之间的双向通讯，可参考此例创建所需要的MQ对象。

--------------------------------------------------------------------------------  
MQ的通讯模式

1) 点对点通讯：点对点方式是最为传统和常见的通讯方式，它支持一对一、一对多、多对多、多对一等多种配置方式，支持树状、网状等多种拓扑结构。

2) 多点广播：MQ适用于不同类型的应用。其中重要的，也是正在发展中的是"多点广播"应用，即能够将消息发送到多个目标站点(Destination List)。可以使用一条MQ指令将单一消息发送到多个目标站点，并确保为每一站点可靠地提供信息。MQ不仅提供了多点广播的功能，而且还拥有智能消息分发功能，在将一条消息发送到同一系统上的多个用户时，MQ将消息的一个复制版本和该系统上接收者的名单发送到目标MQ系统。目标MQ系统在本地复制这些消息，并将它们发送到名单上的队列，从而尽可能减少网络的传输量。

3) 发布/订阅(Publish/Subscribe)模式：发布/订阅功能使消息的分发可以突破目的队列地理指向的限制，使消息按照特定的主题甚至内容进行分发，用户或应用程序可以根据主题或内容接收到所需要的消息。发布/订阅功能使得发送者和接收者之间的耦合关系变得更为松散，发送者不必关心接收者的目的地址，而接收者也不必关心消息的发送地址，而只是根据消息的主题进行消息的收发。在MQ家族产品中，MQ Event Broker是专门用于使用发布/订阅技术进行数据通讯的产品，它支持基于队列和直接基于TCP/IP两种方式的发布和订阅。

4) 群集(Cluster)：为了简化点对点通讯模式中的系统配置，MQ提供Cluster(群集)的解决方案。群集类似于一个域(Domain)，群集内部的队列管理器之间通讯时，不需要两两之间建立消息通道，而是采用群集(Cluster)通道与其它成员通讯，从而大大简化了系统配置。此外，群集中的队列管理器之间能够自动进行负载均衡，当某一队列管理器出现故障时，其它队列管理器可以接管它的工作，从而大大提高系统的高可靠性。

MQ Server和MQ Client

MQ产品分为Server和Client 两种版本，在MQ服务器的运行环境下，有队列管理器、队列、消息通道等对象，它提供全面的消息服务；MQ Client为我们提供了一个MQ应用程序的开发和运行环境，它是MQ API的Client实现。在客户端环境下，没有队列管理器、队列等对象，它通过MQI通道与服务器之间建立通讯，并将消息从客户端发往服务器端的队列，或从Server端的队列中取得消息，它比较适合于网络条件较好或实时通讯的情况。同时要指出的是：采用MQ Client并不会导致数据的丢失或不完整性。MQ Client提供下列好处：适合同步处理的工作模式；减少系统负担；减少系统管理开销；减少磁盘空间要求等。

--------------------------------------------------------------------------------  
MQ的API

MQ支持多种编程语言，其中包括：C、C++、Java、VisualBasic、COBOL、PL/1、RPG等，同时也支持多种流行的开发工具，如：WebSphere Studio Application Developer, PowerBuiler、Microsoft Visual C++、Visual Basic、Delphi等。并且，MQ在不同平台上提供统一的编程接口，仅需重新编译就可完成不同平台间程序的移植。MQ的API接口十分简单易学，用户仅需利用MQ的13个常用而又功能强大的函数调用，便可以以最快的速度，写出各种复杂的应用程序。用户可以将主要精力集中于应用业务逻辑的实现，而不是底层通讯、例外处理等方面。