# 队列

一个安全的存储消息的地方，消息的存储一般是顺序的，队列是消息分阶段地传送和接收。因为消息存放在队列中，所以应用程序可以相互独立的运行，以不同的速度，在不同的时间，在不同的地点。

## 本地队列（local queue）

一个本地队列是一个物理上位于本地队列管理器中的队列。本地队列实际上存在与本地系统的内存或磁盘存储终。本地队列管理器控制队列的访问。应用程序可以“PUT”消息到本地队列，也可以从本地队列“GET”消息，另外程序还可以查询或修改这些队列的某些属性。对队列属性的修改需要相应的权限。

## 远程队列（remote queue）

     一个远程队列属于一个不与该应用程序直接相连的队列管理器。对这类队列的访问包含有本地队列管理器和远程队列管理器的通信过程。这种通信涉及到通道。 应用程序可对远程队列进行某些操作，比如程序可以向一个远程队列放一条消息，但程序不能从远程队列中去消息。应用程序只能从本地队列读取消息。

     应用程序有两种不同的方法可用来访问远程队列。第一种是当程序打开一个远程队列时同时提供队列管理器名和队列名两个参数。这要求程序知道目的队列属于哪个队列管理器。第二种方法是在本地队列管理器上存在一个远程队列的定义，这个定义包含有足够的信息让本地队列管理器确定该远程队列所在的队列管理器。远程队列定义中的目的队列不一定是远程队列管理器的本地队列，它也可以是一个远程队列定义。(这就话和之后的网关队列管理器的创建有关系)

    应用程序放一条消息到Q1，Q1是本地队列管理器QM1上的一个远程队列定义：Q2at QM2。QM2是远程队列管理器。Q2是QM2上的一个远程队列定义Q3 at QM3。Q3是QM3的一个本地队列，经过两次传送，消息最终到达Q3这个QM3的本地队列。

    有多种原因使这种多跳（Multihop）传送变得有意义。在一个TCP/IP网络内部，所有机器都有IP地址，IP协议本身处理节点间的路由选择。但假设消息需要穿过不同类型的网络，这就需要中间件参加路由选择。在图中，QM2位于一台连接TCP/IP网和SNA网的机器上，只需在QM2上提供一个远程队列定义Q2：Q3 at QM3，就可以实现消息的跨网络传输。

    因为对远程队列的访问总是涉及到队列管理器之间的通信，因而我们需要定义其他一些资源，比如通道、传输队列（Transmission queue）。

## 传输队列(Transmission queue)

    传输队列是临时存储目标为远程队列管理器的消息的队列。队列管理器利用传输队列把消息分阶段地发向远程队列。队列管理器和消息移动程序一起负责把数据传送到远程队列。当队列管理器收到把一条消息发往远程队列的要求后，它把消息发送到一个与目的队列管理器相关联的传输队列，传输队列位于本地队列管理器上。目的队列管理器的名称可能由应用程序提供，也可以从远程队列定义中得到。

     一个传输队列是两个队列管理器之间的连接的一端。所有直接目的地是同一队列管理器的消息都可放在同一个传输队列上，这些消息的最终目的可能不一样。把消息从一个队列管理器传送到另一个队列管理器只需要一个传输队列，然而也有可能在两个队列管理器之间存在着多个连接以提供不同的传输服务，每个连接都带有一个不同的传输队列。

     传输队列是由MCA处理的，MCA负责在队列管理器之间可靠地传送消息。MCA实际上是处理传输队列上消息的MQI应用程序。