

WHAT IS A MESSAGING SYSTEM?

Messaging capabilities are typically provided by a separate software system called a *messaging system* or *message-oriented middleware* (MOM). A messaging system manages messaging the way a database system manages data persistence. Just as an administrator must populate the database with the schema for an application's data, an administrator must configure the messaging system with the channels that define the paths of communication between the applications. The messaging system then coordinates and manages the sending and receiving of messages. The primary purpose of a database is to make sure each data record is safely persisted, and likewise the main task of a messaging system is to move messages from the sender's computer to the receiver's computer in a reliable fashion.

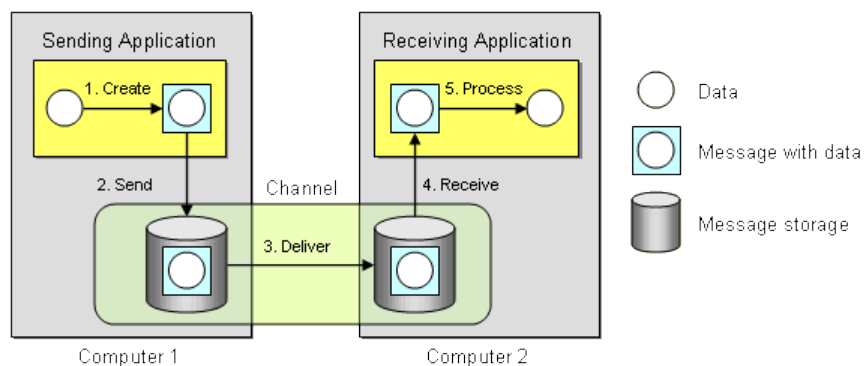
The reason a messaging system is needed to move messages from one computer to another is that computers and the networks that connect them are inherently unreliable. Just because one application is ready to send a communication does not mean that the other application is ready to receive it. Even if both applications are ready, the network may not be working, or may fail to transmit the data properly. A messaging system overcomes these limitations by repeatedly trying to transmit

the message until it succeeds. Under ideal circumstances, the message is transmitted successfully on the first try, but circumstances are often not ideal.

In essence, a message is transmitted in five steps:

1. *Create* — The sender creates the message and populates it with data.
2. *Send* — The sender adds the message to a channel.
3. *Deliver* — The messaging system moves the message from the sender's computer to the receiver's computer, making it available to the receiver.
4. *Receive* — The receiver reads the message from the channel.
5. *Process* — The receiver extracts the data from the message.

This diagram illustrates these five transmission steps, which computer performs each, and which steps involve the messaging system:



Message Transmission Step-by-step

This diagram also illustrates two important messaging concepts:

1. *Send and forget* — In step 2, the sending application sends the message to the message channel. Once that send is complete, the sender can go on to other work while the messaging system transmits the message in the background. The sender can be confident that the receiver will eventually receive the message and does not have to wait until that happens.
2. *Store and forward* — In step 2, when the sending application sends the message to the message channel, the messaging system stores the message on the sender's computer, either in memory or on disk. In step 3, the messaging system delivers the message by forwarding it from the sender's computer to the receiver's computer, and then stores the message once again on the receiver's computer. This store-and-forward process may be repeated many times, as the message is moved from one computer to another, until it reaches the receiver's computer.

The create, send, receive, and process steps may seem like unnecessary overhead.

Why not simply deliver the data to the receiver? By wrapping the data as a message and storing it in the messaging system, the applications delegate to the messaging system the responsibility of delivering the data. Because the data is wrapped as an atomic message, delivery can be retried until it succeeds and the receiver can be assured of reliably receiving exactly one copy of the data.

Source: <http://www.enterpriseintegrationpatterns.com/patterns/messaging/Introduction.html>