

Requesting Return Receipts

When you post a high-level event, you can request a return receipt by including the constant `nReturnReceipt` as one of the posting options. This requests that the **Event Manager** send your application a high-level event that tells you whether the other application accepted your event. Note that this does not necessarily mean that the other application performed any action you might have requested from it.

A return receipt is a high-level event having an `EventClass` and an `eventID` indicated by the two constants:

```
HighLevelEventMsgClass = 'jaym';  
rtrnReceiptMsgID       = 'rtrn';
```

Return receipts are posted by the **Event Manager** on the machine of the receiving application (and not by the receiving application itself). No data buffer is associated with a return receipt. However, the posting **Event Manager** sets the `modifiers` field of the high-level `EventRecord` to one of the following values:

```
msgWasNotAccepted      = 0;  
msgWasFullyAccepted    = 1;  
msgWasPartiallyAccepted = 2;
```

The constant `msgWasNotAccepted` indicates that your event was not accepted by the receiving application. This means that the receiving application was notified of the arrival of your event (through `WaitNextEvent`) but did not call `AcceptHighLevelEvent` to accept the event. The constant `msgWasFullyAccepted` indicates that the receiving application did call `AcceptHighLevelEvent` and retrieved all the data in the optional data buffer. The constant `msgWasPartiallyAccepted` indicates that the receiving application called `AcceptHighLevelEvent`, but that the application's data buffer was too small to hold the data sent with your application and that the receiving application called `WaitNextEvent` before retrieving the rest of the buffer.

Note that a return receipt does not indicate the identity of the receiving application. To determine on whose behalf the **Event Manager** has sent you a particular return receipt, you need to call `AcceptHighLevelEvent`. When `AcceptHighLevelEvent` returns successfully, the `sender` parameter contains a `targetID` record with the fields filled in for the receiving application. With return receipts, the `msgLen` parameter is 0, the `msgBuff` parameter is NIL, and the `msgRefCon` field contains the unique number of the `refCon` parameter of the original high-level event sender (that is, your application).