Creating an Apple Event

You create an Apple event by using the **AECreateAppleEvent** function. You specify the event class and event ID, the target address, the return ID, and the transaction ID to the function. The **AECreateAppleEvent** function creates and returns an Apple event with the attributes set as your application requested. You should not directly manipulate the contents of the Apple event; rather, use **Apple Event Manager** functions to add additional attributes or parameters to it.

This example creates a Multiply event using the **AECreateAppleEvent** function. You specify the event class, the event ID, the address of the server application, a return ID, a transaction ID, and a buffer to store the returned Apple event as parameters to **AECreateAppleEvent**.

```
myErr = AECreateAppleEvent(kArithmeticClass, kMultEventID,
                           &targetAddress, kAutoGenerateReturnID,
                           kAnyTransactionID, &theAppleEvent);
```

The event class here is identified by the kArithmeticClass constant and specifies that this event belongs to a specific class of Apple events for arithmetic operations. The event ID specifies the particular Apple event within the class-in this case, an Apple event to perform multiplication.

You specify the target of the Apple event in the third parameter to **AECreateAppleEvent**. The target address can identify an application on the local computer or another computer on the network. You can specify the address using a target ID record or session ID. For processes on the local computer, you can also use a process serial number or application signature to specify the address.

You specify the return ID of the Apple event in the fourth parameter. The return ID provides a way to associate this Apple event with the server's reply. The **AECreateAppleEvent** function assigns the specified return ID value to the keyReturnIDAttr attribute of the Apple event. If a server returns an Apple event in response to this event, the server should use the same return ID. When you receive an Apple event, you can check the keyReturnIDAttr attribute to determine whether the event is a response to an outstanding Apple event. You can use the kAutoGenerateReturnID constant to request that the **Apple Event Manager** generate a return ID that is unique to this session for the Apple event.

The fifth parameter specifies the transaction ID attribute of the Apple event. A **transaction** refers to a sequence of Apple events that are sent back and forth between the client and server applications, beginning with the client's initial request for a service. All Apple events that are part of one transaction must have the same transaction ID.

You can use a transaction ID to indicate that an Apple event is one of a sequence of Apple events related to a single transaction. The kAnyTransactionID constant indicates that the Apple event is not part of a transaction.

The **AECreateAppleEvent** function creates an Apple Event with only the specified attributes and no parameters. To add parameters or additional attributes, use other **Apple Event Manager** functions.