SendMode Flags

You specify how your application should wait for a reply by using one of these flags in the sendMode parameter of the **AESend** function.

Flag	Description
kAENoReply	Your application does not want a reply Apple event; the server processes your Apple event as soon as it has the opportunity.
kAEQueueReply	Your application wants a reply Apple event; the reply appears in your event queue as soon as the server has the opportunity to process and respond to your Apple event.
kAEWaitReply	Your application wants a reply Apple event and is willing to give up the processor while waiting for the reply; for example, if the server application is on the same computer as your application, your application yields the processor to allow the server to respond to your Apple event.

If you specify the kAEWaitReply flag, you may provide an idle function. This function should process any events that occur while your application is waiting for a reply. You supply a pointer to your idle function as a parameter to the **AESend** function. So that your application can process other Apple events while it is waiting for a reply, you can also specify an optional filter function to the **AESend** function that filters Apple events.

If your Apple event may require the user to interact with the server application (for example, to specify print or file options), you can communicate your user-interaction preferences to the server by specifying one of the following flags in the sendMode parameter of the **AESend** function.

Flag	Description
kAENeverInteract	The server application should never interact with the user in response to this Apple event. If this flag is set, **AEInteractWithUser** does not bring the server application to the foreground (this is the default when an Apple event is sent to a remote application).
kAECanInteract	The server application can interact with the user in response to this Apple event-by convention, if the user needs to supply information to the server. If this flag is set and the server allows interaction, **AEInteractWithUser** brings the server application to the foreground (this is the default when an Apple event is sent to a local application).
kAEAlwaysInteract	The server application can interact with the user in response to this Apple event-by convention, even if no information is needed from the user. If this flag is set and the

server allows interaction, **AEInteractWithUser** brings the server application to the foreground. The **Apple Event Manager** does not distinguish between this flag and the kAECanInteract flag-distinguishing between them is the responsibility of the server application.

kAECanSwitchLayer — If both the client and server allow interaction and this flag is set, **AEInteractWithUser** brings the server directly to the foreground if adherence to the principle of user control allows. If the action would be contrary to this principle, **AEInteractWithUser** uses the **Notification Manager** to request that the user bring the server application to the foreground. If both the client and server allow interaction and this flag is not set, **AEInteractWithUser** always uses the **Notification Manager** to request that the user bring the server application to the foreground.

The server can set its own interaction preferences.