Open Documents Event

To handle the **Open Documents event**, your application should open the documents specified in the Apple event. The **Open Documents event** contains a list of documents to open in its direct parameter. Your application extracts this information and then opens the specified documents.

The following program shows a handler for the **Open Documents event**. The handler illustrates how to open the documents referred to in the direct parameter.

```
// A handler for the Open Documents event

// Assuming inclusion of <MacHeaders>

#include <AppleEvents.h>
pascal  OSErr  MyHandleODOC (AppleEvent *theAppleEvent,
                    AppleEvent *reply, long handlerRefcon);
void    DoError (OSErr myErr);
OSErr   MyGotRequiredParams (AppleEvent *theAppleEvent);
OSErr   MyOpenFile (FSSpec *myFSS);

pascal  OSErr  MyHandleODOC (AppleEvent *theAppleEvent,
                    AppleEvent *reply, long handlerRefcon)
{
    FSSpec      myFSS;
    AEDescList  docList;
    OSErr       myErr;
    long        index, itemsInList;
    Size        actualSize;
    AEKeyword   keywd;
    DescType    returnedType;

    // get the direct parameter--a descriptor list--and put
    // it into docList
    myErr = AEGetParamDesc(theAppleEvent, keyDirectObject,
                        typeAEList, &docList);
    if (myErr)
        DoError(myErr);

    // check for missing required parameters
    myErr = MyGotRequiredParams(theAppleEvent);
    if (myErr) {
        // an error occurred:  do the necessary error handling
        myErr = AEDisposeDesc(&docList);
        return myErr;
    }

    // count the number of descriptor records in the list
    myErr = AECountItems (&docList,&itemsInList);

    // now get each descriptor record from the list, coerce
    // the returned data to an FSSpec record, and open the
    // associated file
    for (index=1; index<=itemsInList; index++) {
        myErr = AEGetNthPtr(&docList, index, typeFSS, &keywd,
                            &returnedType, (Ptr)&myFSS,
                            sizeof(myFSS), &actualSize);
        if (myErr)
            DoError(myErr);
        myErr = MyOpenFile(&myFSS);
        if (myErr)
            DoError(myErr);
    }

    myErr = AEDisposeDesc(&docList);

    return noErr;
}
```

The handler in this program first uses the **AEGetParamDesc** function to get the direct parameter (specified by the keyDirectObject keyword) out of the Apple event. The handler requests that **AEGetParamDesc** return a descriptor list in the *docList* variable. The handler then checks to make sure that it has retrieved all of the required parameters by calling the **MyGotRequiredParams** function. A listing of the **MyGotRequiredParams** function can be found under **Writing Apple Event Handlers**.

Once the handler has retrieved the descriptor list from the Apple event, it uses **AECountItems** to count the number of descriptors in the list. Using the returned number as an *index*, the handler can get the data of each descriptor record in the list. This handler requests that the **AEGetNthPtr** function coerce the data in the descriptor record to a file system specification record (FSSpec). The handler can then use the file system specification record as a parameter to its own routine for opening files.

After extracting the file system specification record (FSSpec) that describes the document to open, your application can use this record to open the file. For example, in the program above, the code passes the file system specification record to its routine for opening files, the **MyOpenFile** function.

The **MyOpenFile** function is designed so that it can be called both in response to the **Open Documents event** and to events generated by the user. For example, when the user chooses **Open** from the **File** menu, the code that handles the mouse-down event uses the **StandardGetFile** procedure to let the user choose a file; it then calls **MyOpenFile**, passing the file system specification record returned by **StandardGetFile**. By isolating code that performs a requested action from code that interacts with the user, you can easily adapt your application to handle Apple events that request the same action.

Note that your handler should use the **AEDisposeDesc** function to dispose of the descriptor list when your handler no longer requires the data in it. Your handler should also return a result code.