CloseDeskAcc Close a desk accessory

#include <Desk.h> **Desk Manager**

```
void           CloseDeskAcc(daRefNum );
short          daRefNum ;     reference number of DA to close
```

Call **CloseDeskAcc** when the user selects the Close item of your File menu and the frontmost window is that of a DA. The DA window is removed from the screen and the next-to-frontmost window is reactivated.

daRefNum identifies the DA to close. The value to use is stored in the windowKind field of the DA's window (see the example, below).

Returns: none

Notes: There is no need to use this call when the user closes a DA by clicking its Close box; in that case the Desk Manager takes care of it. Use **CloseDeskAcc** only when a system window is frontmost and the user picks Close from your File menu.

It is incorrect to use the *daRefNum* returned from a previous call to **OpenDeskAcc**. The DA reference number is stored in the DA's WindowRecord. It is a negative number in the windowKind field. This has ramifications for DAs (see **IsDialogEvent**). For non-DA applications, a typical sequence might include:

```
Boolean isMyWindow (WindowPtr theWindow);

WindowPeek    wPeek;
long mr;
EventRecord theEvent;
WindowRecord whichWindow;

if(WaitNextEvent(everyEvent, &theEvent, 0, nil)) {
    if ( theEvent.what == mouseDown ) {
        switch ( FindWindow( theEvent.where,  &whichWindow ) ) {

        case inMenuBar:
            mr = MenuSelect( theEvent.where );   /* user interaction*/
            if ( HiWord( mr ) == FILE_MENU ) {  /* in File menu? */
                if ( LoWord( mr ) == CLOSE_ITM ) {/* Close Item ? */
                    if ( isMyWindow( FrontWindow()) ) {
                                /* if it's mine */

                        /*  ... close one of my application's windows ..*/

                    }
                    else {        /* must be a DA window */
                        wPeek = (WindowPeek)FrontWindow();
                        CloseDeskAcc( wPeek->windowKind );
                    }
                }
            }
        case inContent:

            /* .. etc ... */
        }
    }
}
```