Opening an Edition Container to Write Data

Several routines are required to write (publish) data from a publisher to an edition container. Before writing data to an edition, you must use the **OpenNewEdition** function. This function should be used only for a publisher within a document. Use this function to initiate the writing of data to an edition.

err = **OpenNewEdition** (*publisherSectionH, fdCreator,*
publisherSectionDocument, refNum)**;**

A user may try to save a document containing a publisher that is unable to write its data to an edition--because another publisher (that shares the same edition) is writing, another subscriber (that shares the same edition) is reading, or a publisher located on another computer is registered to the section. In such a case, you may decide to refrain from writing to the edition so that the user does not have to wait. You should also refrain from displaying an error to the user. The contents of the publisher are saved to disk with the document. The next time that the user saves, you can write the publisher data to the edition. You should discourage users from making multiple copies of the same publisher and pasting them in the same or other documents by displaying an alert box (see **Duplicating Publishers and Subscribers**.

If a user clicks **Send Edition Now** within the publisher options dialog box (to write publisher data to an edition manually), and the publisher is unable to write its data to its edition, you should display an error message.

After you are finished writing data to an edition, use the **CloseEdition** function to close the edition.

The Listing below illustrates how to write data to an edition. You must open the edition, write each format using the **WriteEdition** function, and close the edition using the **CloseEdition** function. This listing shows how to write text only. If the edition is written successfully, subscribers receive Section Read events.

```
// Listing. Writing data to an edition
// Assuming inclusion of Macheaders
#include <Editions.h>

// This is a sample declaration for the pointer to your document information.
typedef struct {
      short resForkRefNum;
      FSSpec fileSpec;
      SectionHandle  sectionH;
      short nextSectionID;
      } *MyDocumentInfoPtr;

// Example app signature
#define kAppSignature 'MINE'

// Prototype your function like this prior to calling
void DoWriteEdition(SectionHandle);
void DoWriteEdition(SectionHandle thePublisher)
{
    EditionRefNum  eRefNum;
    OSErr openErr;
    OSErr writeErr;
    OSErr closeErr;
    MyDocumentInfoPtr thisDocument;
    Handle textHandle;

    // User defined prototypes
    MyDocumentInfoPtr FindDocument(SectionHandle);
    void MyErrHandler(OSErr);
     // Handle any errors that occur i.e. put up alert
    Handle GetTextInSection(SectionHandle,MyDocumentInfoPtr);

     // Find out which document this section belongs to.
     //  The FindDocument function accomplishes this.
     thisDocument = FindDocument(thePublisher);
     // Open edition for writing.
     openErr = OpenNewEdition(thePublisher, kAppSignature,
                         &thisDocument->fileSpec, &eRefNum);
     if ( openErr != noErr )
        // if the open failed, then you can't write,
        //  so don't continue with this operation.
        MyErrHandler(openErr);

     // Get the text data to write. The GetTextInSection
     // function accomplishes this.
     textHandle = GetTextInSection(thePublisher, thisDocument);

     // Write out text data.
     HLock(textHandle);
     writeErr = WriteEdition(eRefNum, 'TEXT', *textHandle,
                                             GetHandleSize(textHandle));
     HUnlock(textHandle);

     if (writeErr != noErr)
          // There were problems writing; simply close the edition.
          //  When successful = FALSE, the edition data != section data.
          //  Note: this isn't fatal or bad; it just means that the
          //  data wasn't written and no Section Read events will be
          //  generated.
          closeErr = CloseEdition(eRefNum, FALSE);

     else
          // The write was successful; now close the edition.
          //  When successful = TRUE, the edition data = section data.
          //  This edition is now available to any subscibers.
          //   Section Read events will be sent to current subscribers.
          closeErr = CloseEdition(eRefNum, TRUE);

}
```