Creating the Edition Container

Use the **CreateEditionContainerFile** function to create an edition container to hold the publisher data.

```
err = CreateEditionContainerFile (editionFile, fdCreator,
        editionFileNameScript );
```

This function creates an edition container. The edition container is empty (that is, it does not contain any formats) at this time.

To create a customized icon for the edition container, put the creator signature of your application with the icon in your application's bundle. See the **Finder Interface** for additional information. Depending on the contents of the edition, the file type will be 'edtp' (for graphics), 'edtt' (for text), or 'edts' (for sound).

After creating the edition container, use the **NewSection** function to create the section record and alias record for the section.

The Listing below illustrates how to create a publisher. The DoNewPublisher function shown in the listing is a function provided by an application. Note that an application might call the DoNewPublisher function as a result of the user making a menu selection to create a publisher or in response to handling the Create Publisher event. See the **Apple Event Manager** description for an example handler that handles the Create Publisher event.

The parameters to the DoNewPublisher function include a pointer to information about the document, a Boolean value that indicates if the function should display the new publisher dialog box, the preview for the edition, the preview format, and an edition container.

The function displays the publisher dialog box if requested, letting the user accept or change the name of the edition and the location where the edition should reside. Use the **CreateEditionContainerFile** function to create the edition with the given name and location. Use the **NewSection** function to create a new section for the publisher.

After the section is created, you must write out the edition data. Be sure to add the newly created section to your list of sections for this document. There are several different techniques for creating publishers and unique IDs; this listing displays one technique.

```
// Listing.  Creating a publisher
// Assuming inclusion of MacHeaders
#include <Editions.h>
#include <AppleEvents.h>

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

```
// Used to create Edition Container file, is signature of your application
#define kAppSignature 'MINE'

// Prototype your function like this prior to calling it
OSErr DoNewPublisher(MyDocumentInfoPtr, Boolean, Handle, FormatType,
            EditionContainerSpec);

OSErr  DoNewPublisher(MyDocumentInfoPtr thisDocument,
                      Boolean promptForDialog, Handle preview,
                      FormatType    previewFormat,
                      EditionContainerSpec editionSpec)
{
    OSErr getLastErr, dialogErr;
    OSErr createErr, sectionErr;
    short resID;
    SectionHandle thisSectionH;
    NewPublisherReply   reply;

    // User defined prototypes
    void MyErrHandler(OSErr);
    void AddSectionAliasPair(MyDocumentInfoPtr, SectionHandle, short);
    void DoWriteEdition(SectionHandle, MyDocumentInfoPtr);
    OSErr MyGetLastError(void);


    // Set up info for new publisher reply record
    reply.replacing = FALSE;
    reply.usePart = FALSE;
    reply.preview = preview;
    reply.previewFormat = previewFormat;
    reply.container = editionSpec;

    if ( promptForDialog ) {
        // user interaction is allowed

        // Display dialog box and let user select.
        dialogErr = NewPublisherDialog(&reply);

        //Dispose of preview data handle.
        DisposHandle(reply.preview);

        //There's usually no error returned here, but if there is,
        // then it makes no sense to continue with this operation.
        if ( dialogErr != noErr )
            MyErrHandler(dialogErr);

        //Do nothing if user canceled.
        if (reply.canceled)
            return userCanceledErr;

    }

    // If user wants to replace an existing file, don't create one.
```

```
    if ( ! reply.replacing ) {
        createErr =
        CreateEditionContainerFile( &reply.container.theFile,
            kAppSignature,reply.container.theFileScript);
        // If the create failed, then this operation can't be completed
        if ( createErr != noErr )
            return errAEEventNotHandled;

    }

    // Advance counter to make a new unique sectionID for this
    // document. It is not required that you equate section IDs with
    // resources.
    thisDocument->nextSectionID++;

    // Create a publisher section.
    sectionErr = NewSection(&reply.container, &thisDocument->fileSpec,
            stPublisher, thisDocument->nextSectionID,
            pumOnSave, &thisSectionH);
    if ((sectionErr != noErr) && (sectionErr != multiplePublisherWrn) &&
        (sectionErr != notThePublisherWrn))
        // If a new section could not be created, don't continue with this
        //  operation.
        MyErrHandler(sectionErr);

    resID = thisDocument->nextSectionID;

    // Add this section/alias pair to my internal bookkeeping.
    //  The AddSectionAliasPair is a routine to accomplish this.
    AddSectionAliasPair(thisDocument, thisSectionH, resID);

    // Write out first edition.
    DoWriteEdition(thisSectionH, thisDocument);

    // Remember that the section and alias records need to be
    //  saved as resources when the user saves the document.
    // Set the function result appropriately
    return MyGetLastError();
}
```