

## Creating a Publisher

You need to support a **Create Publisher** menu command in the **Edit** menu. When a user selects a portion of a document and chooses **Create Publisher** from this menu, you should display the publisher dialog box on the user's screen. The **Create Publisher** menu command should remain dimmed until the user selects a portion of a document.

Use the **NewPublisherDialog** function to display the publisher dialog box on the user's screen. This function is similar to the **CustomPutFile** procedure described in the **Standard File Package**.

```
err = NewPublisherDialog (reply);
```

The dialog box contains space for a preview (a thumbnail sketch) of the edition and a space for the user to type in the name of the edition in which to write the publisher data.

The **NewPublisherDialog** function displays the preview (provided by your application), a text box with the default name of the edition (provided by your application), and handles all user input until the user clicks **Publish** or **Cancel**.

You pass a new publisher reply record as a parameter to the **NewPublisherDialog** function. This is the NewPublisherReply structure.

You fill in the usePart, preview, previewFormat, and container fields of the NewPublisherReply record.

Always set the usePart field to FALSE. The preview field contains either NIL or the data to display in the preview. The previewFormat field should contain 'PICT', 'TEXT', or 'prvw'.

Set the container field to be the default name and folder for the edition. The default name should reflect the data contained in the publisher. For example, if a user publishes a bar chart of sales information entitled 'sales data,' then the default name for the edition could also be 'sales data.' Otherwise, you should use the document name followed by a hyphen (-) and a number to establish uniqueness. For example, your default name could be "January Totals - 3."

If the document has not been saved, the default name should be "untitled edition <*n*>" where *n* is a number to establish uniqueness. The default folder should be the same as the edition for the last publisher created in the same document. If this is the first publisher in the document, the default folder should be the same folder that the document is in.

The canceled field of the new publisher reply record indicates whether the user canceled from the dialog box. The replacing field indicates that the user chose to replace an existing edition file. If replacing returns FALSE, call the **CreateEditionContainerFile** function to create an edition file.

The container field is of structure type EditionContainerSpec.

The field theFile on the EditionContainerSpec structure is of type FSSpec.

You identify the edition using a volume reference number, directory ID, and filename. When specifying an edition, follow the standard conventions

described in the **File Manager** description.

After filling in the fields of the new publisher reply record, pass it as a parameter to the **NewPublisherDialog** function, which displays the publisher dialog box.

```
err = NewPublisherDialog (reply);
```

After displaying the publisher dialog box, use the **CreateEditionContainerFile** function to create the edition container, and then use **NewSection** function to create the section record and the alias record.

In response to the user selecting the **Create Publisher** menu item, this code illustrates how your application might set up the preview for the edition, set the default name for the edition container, and call an application-defined function (DoNewPublisher function) to display the publisher dialog box on the user's screen. 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.

```
#include <Editions.h>
// Assuming inclusion of <MacHeaders>

typedef Ptr MyDocumentInfoPtr;

void MyFunction (void);
Handle MyGetPreviewForSelection (MyDocumentInfoPtr thisDocument);
EditionContainerSpec MyGetDefaultEditionSpec (MyDocumentInfoPtr
    thisDocument);
OSErr DoNewPublisher (MyDocumentInfoPtr thisDocument, Boolean
    promptForDialog, Handle preview, FormatType
    previewFormat, EditionContainerSpec defaultLocation);

void MyFunction()
{
    MyDocumentInfoPtr  thisDocument;
    Boolean            promptForDialog;
    Handle             preview;
    FormatType         previewFormat;
    EditionContainerSpec defaultLocation;
    OSErr              myErr;

    // Get a preview to show the user. The MyGetPreviewForSelection
    // function returns a handle to the preview.
    preview = MyGetPreviewForSelection(thisDocument);
    previewFormat = 'TEXT';
    defaultLocation = MyGetDefaultEditionSpec(thisDocument);
    promptForDialog = TRUE;
    myErr = DoNewPublisher(thisDocument, promptForDialog, preview,
        previewFormat, defaultLocation);
}
```