Modifying a Subscriber

When the user selects data or clicks in the data area of a subscriber, you should highlight the entire contents of the subscriber using reverse video. You can allow users to globally adorn subscribers. For example, a user might select a subscriber within a document and change all text from plain to bold. However, you should discourage users from modifying the individual elements contained within a subscriber--for example, by editing a sentence or rotating an individual graphical object.

Remember that each time a new edition arrives for a subscriber, any modifications that the user has introduced are overwritten. Global adornment of a subscriber is much easier for your application to regenerate.

If you do allow a user to edit a subscriber section, provide an enable/disable editing option within the subscriber options dialog box using the **SectionOptionsExpDialog** function. When you allow a user to edit a subscriber, you should change the subscriber from a selected state to editable data.

In addition to global adornment, your application may also need to support partial selection of subscribers to enable spell checking and search operations.

Because a user can modify a publisher just like any other portion of a document, its subscriber may change in size as well as content. For example, a user may modify a publisher by adding two additional columns to a spreadsheet.

Relocating an Edition

In the Finder, users cannot move an edition across volumes. To relocate an edition, the user must first select its publisher and cancel the section (remember to remove the border). The user needs to republish and then select a new volume location for the edition. As a convenience for the user, you should retain the selection of all the publisher data after the user cancels the section to make it easy to republish the section.

Customizing Dialog Boxes

The expandable dialog box functions allow you to add items to the bottom of the dialog boxes, apply alternate mapping of events to item hits, apply alternate meanings to the item hits, and choose the location of the dialog boxes. See the **Dialog Manager** and the **Standard File Package** for additional information.

The expandable versions of these dialog boxes require five additional parameters. Use the **NewPublisherExpDialog** function to expand the publisher dialog box.

err = **NewPublisherExpDialog** (*reply, where, expansionDITLresID, dlgHook, filterProc, yourDataPtr*);

Use the **NewSubscriberExpDialog** function to expand the subscriber dialog box.

err = **NewSubscriberExpDialog** (*reply, where, expansionDITLresID, dlgHook, filterProc, yourDataPtr*);

Use the **SectionOptionsExpDialog** function to expand the publisher options and the subscriber options dialog boxes.

err = **SectionOptionsExpDialog** **(***reply, where, expansionDITLresID, dlgHook, filterProc, yourDataPtr***);**

The *reply* parameter is a pointer to a **NewPublisherReply**, **NewSubscriberReply**, or **SectionOptionsReply** record, respectively.

You can automatically center the dialog box by passing (-1, -1) in the *where* parameter.

The *expansionDITLresID* parameter should be 0 or a valid dialog item list **(**'DITL') resource ID. This integer is the ID of a dialog item list whose items are appended to the end of the standard dialog item list. The dialog items keep their relative positions, but they are moved as a group to the bottom of the dialog box. See the **Dialog Manager** for additional information on dialog item lists.

The *filterProc* parameter should be a valid, expandable modal filter procedure pointer or NIL. This procedure is called by the **ModalDialog** function. The *filterProc* function enables you to map real events (such as a mouse-down event) to an item hit (such as clicking the **Cancel** button). For instance, you may want to map a keyboard equivalent to an item hit. See the **Dialog Manager** for information on the **ModalDialog** function.

The *dlgHook* parameter should be a valid, expandable dialog hook procedure pointer or NIL. This procedure is called after each call to the **ModalDialog** filter function. The *dlgHook* parameter takes the appropriate action, such as filling in a check box. The *itemOffset* parameter to the procedure is the number of items in the dialog item list before the expansion dialog items. You need to subtract the item offset from the item hit to get the relative item number in the expansion dialog item list. The return value from the *dlgHook* parameter is the absolute item number.

When the **Edition Manager** displays subsidiary dialog boxes in front of another dialog box on the user's screen, your *dlgHook* and *filterProc* parameters should check the refCon field in the **WindowRecord** data type (from the window field in the **DialogRecord**) to determine which window is currently in the foreground. The main dialog box for the **NewPublisherExpDialog** and the **NewSubscriberExpDialog** functions contains the following constant:

sfMainDialogRefCon	new publisher and new subscriber

The main dialog box for the **SectionOptionsExpDialog** function contains the following constant:

emOptionsDialogRefCon	options dialog

The *yourDataPtr* parameter is reserved for your use. It is passed back to your hook and modal filter procedure. This parameter does not have to be of type Ptr--it can be any 32-bit quantity that you want. In Pascal, you can pass in register A6 for *yourDataPtr*, and make *dlgHook* and *filterProc* local functions without the last parameter. The stack frame is set up properly for these

functions to access their parent local variables. See the Standard File Package for detailed information.

For the **NewPublisherExpDialog** and **NewSubscriberExpDialog** functions, all the pseudo-items for the **Standard File Package**--such as

sfHookFirstCall

sfHookNullEvent

sfHookRebuildList

sfHookLastCall

emHookRedrawPreview

For the **SectionOptionsExpDialog** function, the only valid pseudo-items are

sfHookFirstCall	emHookGoToPublisher
sfHookNullEvent	emHookGetEditionNow
sfHookLastCall	emHookSendEditionNow
emHookRedrawPreview	emHookManualUpdateMode
emHookCancelSection	emHookAutoUpdateMode

See **Customizing Your Interface** for more information on **Standard File Package** pseudo-items.