Publishing a New Edition While Saving or Manually

By default, your application should write publisher data to an edition each time the user saves the document and the contents of the publisher differ from the latest edition. In the publisher options dialog box, the user can choose to write new data to an edition each time the document is saved (by clicking **On Save**) or only when the user specifically requests it (by clicking **Manually**).

When the update mode is set to manual, a user must click the **Send Edition Now** button within the **Publisher options** dialog box to write publisher data to an edition. When a user clicks **Send Edition Now**, the section options reply record contains the action code 'writ'. Write out the new edition beginning with the **OpenNewEdition** function. Writing to an edition manually is useful when a user tends to save a document numerous times while revising it.

Each time the user saves the document, check the update mode of the publisher section. If the publisher section sends its data to an edition on save, check to see whether the publisher data has changed since it was last written out to the edition. If so, write out the new edition.

In addition, you may also support a **Stop All Editions** menu command to provide a method for temporarily suspending all update activity. See **Publishers, Subscribers, and Editions** for additional information.

Subscribing to an Edition Automatically or Manually

By default, your application should subscribe to an edition each time new edition data becomes available. In the subscriber options dialog box, the user can choose to read new data from an edition as the data is available (by clicking **Automatically**) or only when the user specifically requests it (by clicking **Manually**).

When the update mode is set to manual, the user must click the Get Edition Now button within the subscriber options dialog box to receive new editions. When a user clicks this button, the section options reply record contains the action code 'read'. Read in the new edition beginning with the **OpenEdition** function. See **Opening an Edition Container to Read Data** for detailed information.

When the update mode is set to automatic, your application receives a Section Read event each time a new edition becomes available. In response, you should read the new edition data beginning with the **OpenNewEdition** function.

Your application does not receive Section Read events for subscribers that receive new editions manually.

You may also support a **Stop All Editions** menu command to provide a method for temporarily suspending all update activity. See the section entitled **Publishers, Subscribers, and Editions** for additional information.

Canceling Sections Within Documents

The option of canceling publishers and subscribers is available to the user through the **Cancel Publisher** and **Cancel Subscriber** buttons in the corresponding options dialog boxes. When the user wants to cancel the publisher or cancel the subscriber within a document, the action code of the section options reply record is 'cncl'. See "Relocating an Edition" for additional information on canceling a section.

When a user cancels a section (either a publisher or subscriber) and then saves the document, or when a user closes an untitled document (which contains newly created sections) without saving, you must unregister each corresponding section record and alias record using the **UnRegisterSection** function. In addition, you should also delete the section record and alias record using the DisposHandle procedure.

When a user cancels a publisher section and then saves the document, or when a user closes an untitled document (which contains newly created publishers) without saving, you must also delete any corresponding edition containers (in addition to deleting section records and alias records).

Do not delete an edition container file, section record, or alias record until the user saves the document--the user may decide to undo changes before saving the document.

To locate the appropriate edition container to be deleted (before you use the **UnRegisterSection** function), use the **GetEditionInfo** function.

err = **GetEditionInfo** (*sectionH, editionInfo*);

The *editionInfo* parameter is a record of data type **EditionInfoRecord**.

The **GetEditionInfo** function returns the edition container as part of the edition information returned on the EditionInfoRecord structure.

The crDate field of the EditionInfoRecord structure contains the creation date of the edition. The mdDate field contains the modification date of the edition.

The fdType and the fdCreator fields are the type and creator of the edition file. The container field includes a volume reference number, directory ID, filename, script, and part number for the edition.

To remove the edition container, use the **DeleteEditionContainerFile** function.

err = **DeleteEditionContainerFile** (editionFile);

Locating a Publisher Through a Subscriber

The user can locate a publisher from a subscriber within a document by clicking the **Open Publisher** button in the subscriber options dialog box. As a shortcut, Apple suggests that you also allow the user to locate a publisher when the user selects a subscriber within a document and presses Option-double-click.

When the action code of the SectionOptionsReply record is 'goto', use the **GoToPublisherSection** function.

err = **GoToPublisherSection** (*container*);

The **GoToPublisherSection** function locates the correct document by resolving the alias in the edition, and it launches the document's application if necessary (the **Edition Manager** sends an Open Documents event). The **Edition Manager** then sends the publishing application a Section Scroll event. If the document containing the requested publisher is located on the same computer as its subscriber, the document opens and scrolls to the location of the publisher. If the document containing the requested publisher is located on a shared volume (using file sharing), the document opens and scrolls to the location of the publisher only if the user has privileges to open the document from the Finder.

You need to provide the **GoToPublisherSection** function with the edition container. To accomplish this, use the **GetEditionInfo** function. See the previous section, "Canceling Sections Within Documents," for information on the **GetEditionInfo** function.

Renaming a Document Containing Sections

If a user renames a document that contains sections by choosing **Save As** from the **File** menu, or if a user pastes a portion of a document that contains a section into another document, use the **AssociateSection** function.

Use the **AssociateSection** function to update the alias record of a registered section.

err = **AssociateSection** **(***sectionH, newSectionDocument***);**

The **AssociateSection** function internally calls the **UpdateAlias** function. It is also possible to update the alias record using the **Alias Manager**.