;K_ToDo_Timers.rtfd;linkMarkername ;¬ Previous Section

4. To Do Tutorial

Build, Run, and Extend the Application

Although you probably have been building the ToDo project frequently now, as it's been taking shape, build it one more time and check out what you have wrought. Go through the following sequence and observe To Do's behavior.

1. When you choose New from the Document menu, the application creates a new To Do document and selects the current day.

2. Enter a few items. Click a new day on the calendar and enter a few more items. Click the previous day and notice how the items you entered reappear.

3. Choose Inspector from the main menu. When the inspector appears, click an item and notice how the name and date of the item appears in the top part of the inspector. Enter due times for a couple items, and some associated notes. Note how the times, as you enter them, appear in the Status/Due column of the To Do document. Click among a few items again and note how the Notifications and Notes displays change.

4. Click a Status/Due button; the image toggles among the three states. Then, with an item that has a due time, select a notification time that has already passed. The application immediately displays an attention panel with a notification message. When you dismiss this panel, To Do sets the notification option to ªDo not notify.º

5. Click the document window and respond to the attention panel by clicking Save. In the Save panel, give the document a location and name. When the window has closed, chose Open from the Document menu and open the same document. Observe how the items you entered are redisplayed.

Optional Exercises

You should be able now to supplement the To Do application with other features and behaviors. Try some of the following suggestions.

Make Your Own Info Panel

Make your own Info panel. Define a method that responds to a click on the Info panel button by loading a nib file containing the panel. The owner of the panel can be the application controller. You can customize this panel however you wish. For instance, put the application icon in a toggled button (the main image) and make the alternate image a photo (yourself, your significant other, your dog). When users click the button, the image changes between the two.

Implement Application Preferences

Make a Preferences panel for the application, with a new controller object (or the application controller) as the owner of the nib file containing the panel. Follow what you've done for ToDoInspector, especially if the panel has multiple displays. Some ideas for Preferences: how long to keep expired ToDoItems before logging and purging them (see below); the default document to open upon launch; the default rescheduling interval (see below). Store and retrieve specified preferences as user defaults; for more information, see the NSUserDefaults specification.

Implement Rescheduling

ToDo's Inspector pane has a Rescheduling display that does almost nothing now. Implement the capability for rescheduling items by the period specified.

Implement Logging and Purging

After certain period (set via Preferences), append expired ToDoItems (as formatted text) to a log, and expunge the ToDoItems from the application.