

CS193E: Assignment 2

FavoriteThings I

Due Date

This assignment is due by 11:59 PM, January 23.

Assignment

Over the next two weeks you will build a small application that lets a user keep track of their favorite things. The application will make use of and give you practice with many of the central design patterns used in Cocoa.

This week you will get the basics up and running - creating and connecting the model, view, and controller. The favorite things you'll be tracking are a favorite word, number, and date. You'll also enable the model to be saved out to disk in an archive. Finally, you'll make some tweaks to the standard AppKit behavior by doing additional setup after nib loading, and by using delegation.

Here's a rough outline of what you'll need to do to complete the assignment successfully:

Basic Model-View-Controller definition and hookup

1. Create a new 'Cocoa application' project named FavoriteThings.
2. Lay out the user interface.
3. Create the application controller and wire it up to the UI elements.
4. Create and implement the FavoritesRepository class, including keyed archiving.
5. Implement an action method that is triggered by the various editing controls. The method should take the values from the controls and set them in the model.

Object persistence

6. Implement a saveChanges: action to archive the favorites repository to the file system in the user's Application Support folder. When an application uses the Application Support folder, it usually creates a folder (directory) within it with the same name as the application. For grading purposes, please also include your Stanford user name in the name of the folder. For instance "FavoriteThings-dempsey1".
7. Implement awakeFromNib to unarchive the saved favorites if they exist or create a new FavoritesRepository if there is none already.

Other tweaks

8. Before the window is displayed change its title to include the user's full name. Also center the window before it is displayed.
9. Handle setting window document edited state when changes occur and when changes are saved
10. Make the application controller the delegate of the window. Only allow the window to close if the window is not edited.
11. Add a checkbox that controls if the application quits when the last window is closed. Make the application controller the delegate of the application as well. Use the state of the checkbox to determine whether the application should terminate after the last window is closed.

Testing

Make sure the following work in your application (these are the things we will test to determine your grade):

0. Your project should build without errors or warnings.
1. Your project should run without crashing.
2. When launched the title of the window should have the user's full name as well as 'Favorite Things'. For instance, for the user James Dempsey, the title of the window should be "James Dempsey's Favorite Things". This being a computer science class, we will not expect perfect grammar. "Paul Marcos's Favorite Things" is perfectly acceptable.
3. When finished editing any UI element, the window's change indicator should show the window has an unsaved change.
4. Clicking the Save button should save changes and clear the window's change indicator.
5. Trying to close the window with unsaved changes should not be allowed - clicking the close button on the window should do nothing.
6. When launched the user's saved favorite things should be read in and displayed. The window should appear with the saved favorite information already filled in (as opposed to showing the window first with empty fields and then the saved values appearing).
7. When the "Quit when window closed" checkbox is checked, closing the window should also quit the application. When it is unchecked, closing the window should not quit the application.

Hints

The NSDatePicker is a handy control for displaying and editing dates. Note that it can be configured in Interface Builder to display both a date and a time, or only one or the other.

The favorite number does not need to be an NSNumber, an int is fine. In Cocoa we tend to only use NSNumber when a C numeric type needs to be wrapped as an object. The date should be an NSDate. The word should be an NSString.

File paths: You can use the NSString category you created in exercise 1B to retrieve the path to the user's Application Support folder. You may want to create two more category methods, one that returns the path to your app's folder within the Application Support folder, and one that returns the path to the archive file itself.

Remember that the NSFileManager class is handy for checking if a file exists (regardless of whether it is a directory or data file) and is also handy for creating new folders - which you will need to do.

Note that when you archive a root object to a file, the previously existing file, if present is automatically replaced.

Troubleshooting

Some common problems when using a delegate are forgetting to hook up the delegate (either in Interface Builder, or programmatically) and misspelling the name of a delegate method. If your delegate methods don't seem to be working correctly, this is a great opportunity to try out the debugger in Xcode. Set a breakpoint in your delegate methods and see if they get hit when you would expect them to be. You can also get some practice with NSLog statements to verify that your methods are being called correctly.

The troubleshooting hints from assignment 1A are still valid - especially regarding checking connections in Interface Builder for outlets and actions.

Extra Credit

Keep in mind that extra credit only applies if the required behavior outlined above is working. Don't spend time on extra credit until you've got the basics down.

- Aesthetics matter: a particularly attractive, innovative, or clever design of the interface is always appreciated. (If you want consideration for extra credit in this case, you must let us know when you submit your project.)
- Add a "Favorite URL." Add a text field where the user can enter their favorite website/URL. Make sure you have a way to store the URL in the model, and that the URL persists between launches of the application. Note that the text field will display a string value, but you'd like to store an NSURL in the model.

Also add a button next to the URL text field that launches the user's default browser and opens that URL. Take a look at the shared workspace of NSWorkspace for a way of opening the URL.

- Currently when the "Quit when window closed" checkbox is unchecked, and you close the Favorite Things window, there is no way to get the window to reappear. Add a menu item to the Window menu that targets the window and tells it to make itself the key window and order itself to the front. Make sure the title of this menu item matches the title of the window.

Also, if you add the menu item to the Window menu, you'll notice when the window is open, it appears in the Window menu twice—once because of the menu item you added, and a second time because windows are automatically added to the Window menu. Take a look in NSWindow to find a way to tweak this behavior so the second listing doesn't appear.