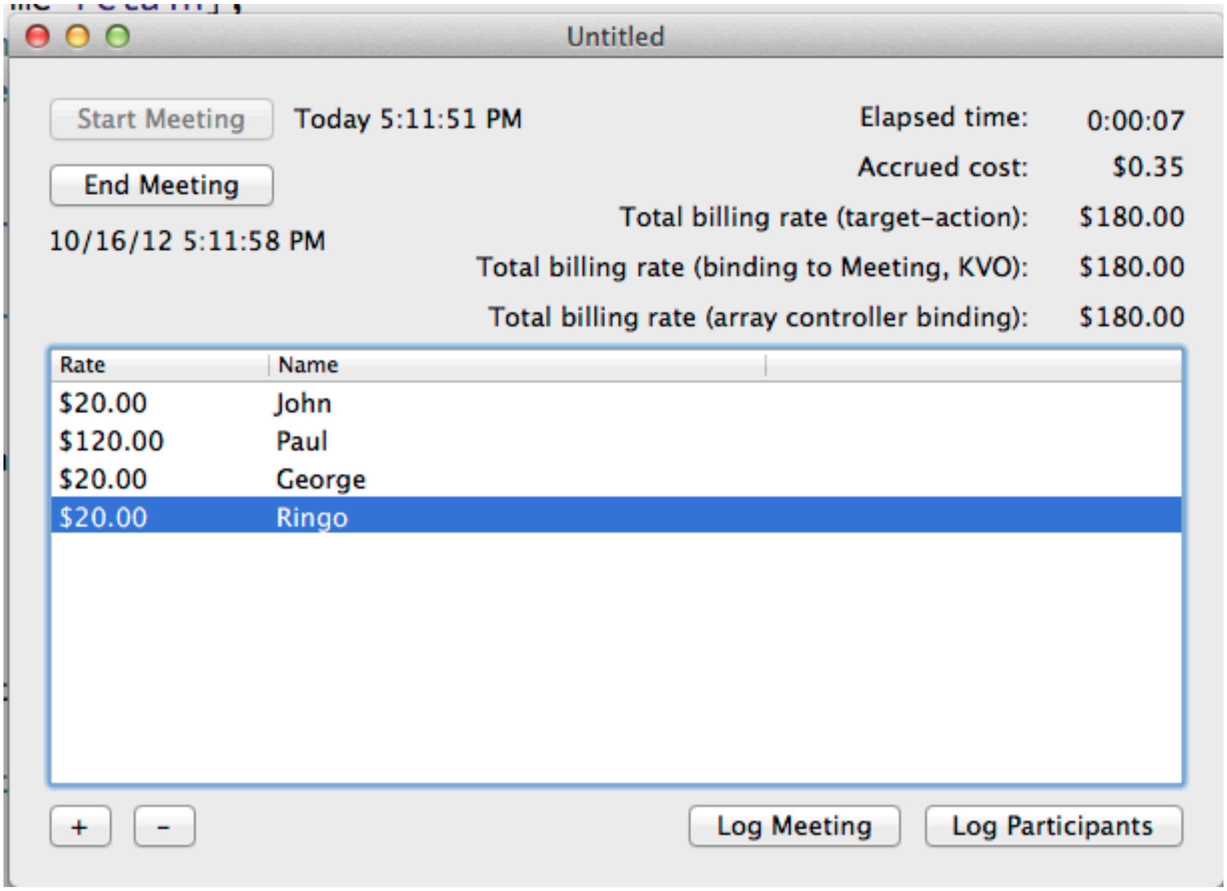
Meeting Cost Tracker II

This exercise adds UI to your Model code from last week, using NSArrayController, NSTableView, and bindings. It also adds archiving.



Today 5:11:51 PM

10/16/12 5:11:58 PM

Elapsed time: 0:00:07

Accrued cost: $0.35

Total billing rate (target-action): $180.00

Total billing rate (binding to Meeting, KVO): $180.00

Total billing rate (array controller binding): $180.00

Rate	Name
$20.00	John
$120.00	Paul
$20.00	George
$20.00	Ringo

Create a GUI for your document-based application with these features:

- a table of participants
- insert and delete buttons
- archiving
- an elapsed time clock
- display for hourly cost of meeting

- display for running dollar cost
- "start meeting" and "stop meeting" buttons
- start and stop time display
- the same Log buttons and live time display from last week

A newly created Meeting document should have no participants.

Do not implement Undo. We'll tackle that next week.

Use an NSTableView to display the participants. The NSTableView should list _only_ the names and rates. Assume that someone attended the ENTIRE meeting if they are in the array, and don't sweat the case of preventing person or rate changes while the meeting is running.

There are several techniques you can use to compute the combined hourly rate for the meeting. I've shown multiple options. You only need to do it once (although you're welcome to do it multiple ways for practice). Some ideas:

- walk the array with fast enumeration, and accumulate the total.
- use `-enumerateObjectsUsingBlock:` to walk the array.
- apply `@sum` to the participants array keypath using bindings on the window or array controller, per Hillegass and Preble (p. 126).

- apply `@sum` to the participants array within the model (`[personsArray valueForKeyPath:"@sum.hourlyRate"]`).
- use key-value observing to start/stop observation of each participant's `rate` as participants enter/leave the meeting. You'll also need to implement start/stop when the entire NSMutableArray is replaced. You may also want to implement – `insertObjectInParticipantsAtIndex:.`

The NSTableView needs an NSArrayController, which will require the use of bindings. Use bindings in other places too, if you like. Using target-action, as we did in IB Calisthenics, is also fine.

Grading Focus: multiple document windows open and close. Table view, array controller, and array work together correctly, and insert and remove participants. Table columns can be resized and rearranged. Archiving and unarchiving work. Meeting rate and accrued cost computations are done correctly using at least one technique (no penalty for additional attempts that don't work, as long as at least one does).