

Homework 5

The Model-View-Controller Pattern, and Background Jobs

due before class on **November 14th**

Outline

In this example, the student will implement a simple file-download utility: the user will be able to specify a URL, and that URL will be saved to a specified local file.

The student should use the Model-View-Controller pattern to implement the design, separating the task of downloading the file (the Model) and of controlling the behavior of the application (the Controller) from the display logic (the View). The exact design is not specified in detail; this is left to the student.

As in the last homework, your classes must be fully encapsulated; they must communicate only with events (which you will add), and through the use of utility methods (to get the URL or save-file path from the View, for example).

Students will turn in their submissions using iCollege. Students should submit a zip-file containing their project directory. Students should ensure that their code is submitted in a buildable form.

The View

The View should have two text fields – one for the URL to download, and one for the file to write to. Additionally, there should be a “select file” button, which produces a save-file dialog; if a file is selected, the path to that file should be filled in in the corresponding text field. Finally, there should be a “Start Download” button; when that button is pressed, a `download_requested` event should be emitted.

The View should have several additional states; it should be able to display an error (using a dialog), and it should also have a state for when a download is running (in which case the “Start Download” button should be disabled).

The Model and Controller

The design of the Model and Controller is left up to the student; the MVC pattern is very flexible, and the student is even free to merge the Model and Controller, for example, if they feel that is the best way to adapt the pattern to their solution.

Downloading in Another Thread

Finally, students should ensure that the download is performed in a separate thread. Students are reminded to take care to ensure that graphical components are manipulated only from the main thread.

Graduate Students

Graduate Students should additionally add a *progress indicator* to the view, and arrange for progress to be reported from the download thread.

Downloading Files

For reference, consider this [simple technique for downloading files in Java](#), and this method for [downloading files in C#](#).