Loading and Saving Spreadsheet Data
Cpt S 322 Homework Assignment #9
by Evan Olds

Submission Instructions:

Submit source code (zipped) to Angel BEFORE the due date/time. If the Angel submission is not working, then submit to TA via email BEFORE the due date/time. "Angel wasn't working" is never an excuse.

Optional: Include a readme.txt file in the zip with any relevant information that you want the grader to be aware of.

Assignment Instructions:

Read each step's instructions *carefully* before you write any code.

In this assignment you will add loading and saving capabilities to your spreadsheet application. If you created a Workbook class for the previous assignment then add Load and Save methods to it. Otherwise you can add these methods to your Spreadsheet class.

Design an XML format for your spreadsheet data. At a high level it will probably have a structure somewhat like:

```
<spreadsheet>
  <cell>
    <bg>FF8000</bg>
    <text>=A1+6</text>
  </cell>
</spreadsheet>
```

You'll obviously have more than one cell in most cases. Make sure you do the following:

- Save to and load from streams
- Add save and load menu options in the interface
- Make sure the saving and loading code is in the logic engine
- Use existing XML classes from the .NET framework (see notes on Angel)
- When saving, only write data from cells that have one or more non-default properties. This means that if a cell hasn't been changed in any way then you don't need to write data for it to the file.

- Clear the undo/redo stacks after loading a file
- Clear all spreadsheet data before loading file data. The load-from-file action is NOT a merge with existing content.
- Make sure formulas are properly evaluated after loading.
- You may assume only valid XML files will be loaded, but make sure loading is resilient to XML that has different ordering from what your saving code produces as well as extra tags. As a simple example, if you're always writing the <bg> tag first for each cell followed by the <text> tag, then your loader must still support files that have these two in the opposite order. Also if you didn't write more than these two tags within the <cell> content, your loader should just ignore extra tags when loading. See the example below.

If you saved:

```
<spreadsheet>
 <cell>
  <bg>FF8000</bg>
  <text>=A1+6</text>
 </cell>
</spreadsheet>
```

Then you should be able to load:

```
<spreadsheet>
 <cell> <text>=A1+6</text> <bg>FF8000</bg>
  <some_tag_you_didnt_write>blah</some_tag_you_didnt_write>
 </cell>
</spreadsheet>
```