**High Priority (i.e., before we release)**

1. The new multiple folders scheme allows a user to select a template in a folder without any images in it. This introduces all sorts of problems. For example, if the folder name ‘Site’ contains the template, but stores images in subfolders Set1 and Set2, the user can just load the template with no images.

**Problems.**

* 1. The menu item name ‘Load Images from Folder’ doesn’t reflect the semantics of what now happens (i.e., its Load Template (and optionally images))
  2. The dialog box ‘No Images Found in the Image Set’ is displayed, which needs to be altered to tell the user to use the add images using the Add Images to Data File (I don’t like the name of that… to be consistent, we should name it something like ‘Add Images to Image Set’.
  3. The screen still shows the main help text after this operation, which looks like nothing has happened (although the template appears to be actually loaded). This is, of course, confusing.

**Solution? I**f a person loads a template from a folder with no images in it, it should immediately raise a dialog asking the user to add images. If the user doesn’t, then the whole operation is cancelled (i.e., no template is loaded). Alternately, the help text should be replaced by something that tells the user to add images using the Add Images to Data File menu item.

1. The Folder field needs to contain the name of the folder containing the template, rather than just ‘.’ Or empty.

**Problems.**

* 1. This is a backwards compatibility issue, as my user base needs the source folder name to identify where the images (and template) came from.
  2. It shows up in the base vanilla case, i.e., loading a folder that contains the template and the images.

**Solution:** Always include the root folder in the path. For example, if the folder name ‘Site’ contains the template, but stores images in subfolders Set1 and Set2, the Folder Path for the images would be Site/Set1 and Site/Set2 instead of ./Set1 ./Set2 etc. If the folder path was somewhere else, it would be stored relative to Site e.g., Site/../AnotherFolder. If the folder Site contained images, the folder path would just be Site.

While we could have alternate solutions by recording both separately (e.g., a field called Template Folder and a field called ImageFolder, I think this adds un-needed complexity and has backward compatibility issues.

1. CSV Import and Export

**Problems.**

* 1. **FIXED by SAUL** The CSV Import doesn’t seem to be importing flag values
  2. **FIXED by SAUL** Check if CSV Export exports flag values (they are set in the database)
  3. **Tested by SAUL – LIST OF REMAINING PROBLEMS THAT CAN BE FIXED IN THE FUTURE, but OK for now as the user is provided with rules on what not to change and formats in a dialog box.** The following problems can be solved by doing some error checking and by restricting what can be imported, as listed below.
     1. If a user tries to change the CSV column order (or add/delete columns), data will be imported into the wrong column.   
        ***SUGGESTED FIX:*** Check the column names against the datalabel. If they differ, either abort, or make sure updates are done to the datalabel as identified by the column name,
     2. If a user changes a value to a non-legitmate one (e.g., Foo instead of False for flags, non-numbers for counters, etc., strings that don’t match fixed choices) the data should be considered corrupted. ***SUGGESTED FIX:*** Check the data type against the format that is required (which you can find by examining the data label.) If it doesn’t conform, then indicate the problem and abort.
     3. Changing File / Folder names will likely screw things up seriously.  
        ***SUGGESTED FIX:*** Ignore any changes to File/Folder names
     4. Importing often changes the format of the dates and time, which then affects the export format. This is because some spreadsheet packages – like excel – read in dates and time and changes it to a particular format. So this can lead to an undesired change.   
        ***SUGGESTED FIX:*** Ignore any changes to File/Folder names, or parse the dates to change them into the standard Timelapse date representation.
     5. ***Not a problem:*** If a subset of image data is exported to the CSV file, the import seems to work as it only updates only the data associated with the image named in those rows.
  4. **Tested by SAUL** - WORKS FINE. CSV export will currently export only those images shown in the current filter list. Not sure what happens if we import an abridged set of images.
  5. **FIXED BY SAUL – WARNING DIALOG BOX RAISED FOR IMPORT.** When a user tries to import a CVS, a warning dialog box is raised that lists the rules and dangers of an import.
  6. **FIXED BY SAUL – WARNING DIALOG BOX RAISED FOR EXPORT.** For the point above, we should raise a dialog (or perhaps just make it two separate submenu options) that has the user either export all images, or only the filtered images. Otherwise its easy for the user to accidentally export only a subset without realizing it.
  7. **FIXED by SAUL** The last field in each CSV file row isn’t imported.

1. **FIXED by SAUL** Template Editor – adjusting output columns can’t be done if there are many options.

**Problems and solution**

* 1. The horizontal scrollbar doesn’t appear, which would fix this problem. This is an easy fix (I’ve already done this on a test copy).

1. Check the use of the images (in ImageSource) in the Timelapse XAML against Todd’s change that swaps WriteableBitmaps for the various bitmap types I used before.

**Problem:**

* 1. Todd wrote:What about the other 11 instances of <Image> in Timelapse XAML?  It’s concerning FeedbackCtl exposes a contract where the caller can pass any ImageSource but only behaves properly with certain derived types thereof.  The implication’s supplying WriteableBitmaps might break the other uses of Image controls.  A naïve fix would be to type FeedbackCtl’s API to BitmapFrame rather than ImageSource but the concern seems broad enough a deeper look might be in order.
  2. Note that things do seem to be working correctly, but check the various dialogs that show images (e.g., delete, swap day and month, etc.

1. Test TImelapse hard.
   1. Do all the routing things with all data types. Also try a complex Template that swaps the various row order, as well as the output data for the spreadsheet.

**Medium Priority**

1. Add the ability to code .avi files as well as jpgs. (Todd/Saul talk about different options). I would prefer to keep the markers in there, as it simplifies coding greatly to avoid special cases. My idea is to grab the first frame and create a jpg out of it and use that as a static image that can be marked, but to have a ‘play’ button that allows the video to be played (which can easily be done by overlaying or popping up a window atop the image). I also think the user should have the ability to swap the static image for any frame in the video. This is easy to do in a robust way.
2. Update the user manual
3. TODD SAYS THIS IS FIXED. NoteCtl\_TextChanged bug. This may already be fixed.
   1. Todd wrote:

Hi, had a little time to poke around in getting closer to more normal keyboard handling for tab order among controls and stuff.  I’m calling it a night with some hacky prototype fixes not worth sharing.

I do have a fix for text entry mangling which seems solid, though.  Doesn’t make sense to pump the full git process for just this bit:

            // Don't allow leading whitespace in the note

            // Updating the text box moves the caret to the start position, which results in poor user experience when the text box initially contains only

            // whitespace and the user happens to move focus to the control in such a way that the first non-whitespace character entered follows some of the

            // whitespace---the result's the first character of the word ends up at the end rather than at the beginning.  Whitespace only fields are common

            // as the Template Editor defaults note fields to a single space.

            int cursorPosition = textBox.CaretIndex;

            string trimmedNote = textBox.Text.TrimStart();

            if (trimmedNote != textBox.Text)

            {

                cursorPosition -= textBox.Text.Length - trimmedNote.Length;

                if (cursorPosition < 0)

                {

                    cursorPosition = 0;

                }

                textBox.Text = trimmedNote;

                textBox.CaretIndex = cursorPosition;

            }