Native Encoder

# Overview:

This is the main component where the video is encoded. We use our own xml parser to read the story configuration xml file, use WIC manipulate the photos, use Media Foundation to encode the video, and use the native transition library to handle transitions.

# Goals:

* Create a video encoder component.

# None Goals:

* Handle transitions. Transitions should be handled by the transition library.

# Design:

Except for property Get/Set methods, the VideoEncoder class provides a single public method: Encode. This method performs all the tasks. In summary, it performs the following steps:

1. Initialize all components we rely on, such as the xml parser, WIC factory, and so on.
2. Use the xml parser to parse the story configure file, and read it into an in-memory data structure. We use a list<Photo\*>, as a simple list should be enough to store the data.
3. Create and configure the Media Foundation SinkWriter.
4. Iterate through the photos. For each photo:
   1. Decode the photo using WIC.
   2. Convert the source jpg RGB format to RGBA. Some transitions may rely on the alpha channel.
   3. Insert frames to the sink writer according to the photo’s display time.
   4. Invoke the transition library to insert transition frames.
5. Finalize the sink writer to commit the video.

By using sink writer, we have a dependency on Windows Server 2008 R2 or later. But this is not a problem in Windows Azure, as we can use OS 2.x.