Worker Role

# Overview:

We use a Worker Role to encode the video. The actual encoding is implemented in the native component. But the Worker Role still performs certain tasks, such as interacting with Azure storage.

# Goals:

* Prepare the environment for the native component to encode the video.
* Upload the encoded video to Azure storage

# None Goals:

* Encode the video.

# Design:

It is debatable on which tasks to be implemented in managed code and which to be implemented in native code. Our current design is:

* Write as much as managed code as possible, because it’s much easier.
* Write performance critical code in native code.
* Avoid too much P/Invoke.

So we choose to encode the video in native code (as this task is performance critical and use Media Foundation in managed code will result in too many P/Invokes). We also parse the xml file in native code, to avoid a separate P/Invoke for each photo.

But we can safely download the blobs in managed code, and upload the encoded video to blob in managed code. The managed code is also responsible for tracing and logging.

Currently we choose to use public blobs for the encoded videos, so everyone can view the videos. In the future, we may need to implement access control.

Finally, the Worker Role must provide environment for the native component. We use startup tasks to install VC++ 2010 runtime, and enable Desktop Experience.