MOS Protocol Proposal – storyStatus

Proposal from Kurt Simons, Hearst

PROBLEM AT HAND:

It is common practice for a producer during a show to float a story if the video, graphics, or live shot is not yet ready. It is also common practice to have a story floated in advance of a show if a producer does not feel the content will be ready in time but wanted to have the story written and coded just in case that it does become available in time for their show.

Unfortunately, the MOS Protocol treats a floated story the same as a deleted story, and thus are not even sent by the NRCS to the MOS Device.

Once an item is dropped by the NRCS the MOS device cannot provide status updates to the now unknown MOS item, an editor looking to link the video to the ENPS rundown can no longer do so, and/or if the asset is on a remote device the local MOS device cannot initiate MOS redirection (transfer) on the item.

The producer is thus forced to move the story down in the rundown rather than float it. A better workflow would be if they could float the story, still have all the other mechanisms work, and when the MOS status says “READY” the story can be unfloated and aired.

PROPOSAL:

storyStatus

A new element to any MOS story message which would indicate if the story is floated or z'd out (below the bottom line) "INACTIVE"

Floated stories would not be treated differently than any other story. Prompters would receive an roStorySend anytime the story body was modified.

An roCreate would be followed by an roStorySend for each story associated with the rundown including floated stories.

The concept of "floating" becomes a flag in the xml, and it becomes up to the MOS device to handle or ignore the floated story as it best sees fit.

roID
storyID
story+
storyID
storySlug?
storyNum?
storyStatus?

Blank is the default, as it does not need to be sent if it is blank

```
<storyStatus></storyStatus>
<storyStatus>FLOATED</storyStatus>
<storyStatus>INACTIVE</storyStatus>
```

In an roList:

mosID
ncsID
messageID
roList
roID
roSlug
roChannel?
roEdStart?
roEdDur?
roTrigger?
macroIn?
macroOut?
mosExternalMetadata*
story*
storyID
storySlug?
storyNum?
storyStatus?
mosExternalMetadata*