

# Element Status Code

*A proposal for enhancing RO element statuses.*

By: Andre Edmonson

## Contents

Motivation.....	3
Proposal .....	4
Example.....	5
Video of kitten up a tree - Success.....	5
Video of kitten up a tree - Failure .....	8

## Motivation

Element statuses have been a critical part of newsroom user feedback. The intent of standardizing a set of statuses was to allow an NCS to provide enhanced feedback, such as color coordination, to aid users in identifying playable, and non-playable content. The current list defined list of statuses are as follows:

"NEW" "UPDATED" "MOVED" "BUSY " "DELETED", "NCS CTRL", "MANUAL CTRL",  
"READY", "NOT READY", "PLAY," "STOP"

In recent years, new workflows have emerged which have prompted the need to add to this list. However, it is difficult and impractical for vendors and the MOS committee to support a list that is so volatile. This has led to devices sending proprietary statuses which provide more detailed messages, at the expense of losing the semantic meaning to the NCS. Today, most NCS's will treat "READY" and "NOT READY" as an indication of playable vs non-playable, while other statuses are handled arbitrarily and often without the enhanced feedback.

## Proposal

The problem with the current statuses is they are trying to convey both a message and meaning in a single field. Rather than standardize on the *message*, it is better to agree on a set of *meanings* and allow the messages to be proprietary. This proposal will add two new optional tags to the <roElementStat> message, that will provide a *meaning to the message* that is <status>.

The first field is called <statusCode>. This is a numeric range between 100-599, classifying the status from playable to non-playable. Values 100-299 are considered the playable range, 400-599 are non-playable and 300-399 are transitioning states. The ranges are intentionally similar to HTTP status codes as the semantics are comparable.

The second field is called <statusLiteral>. This is a human-readable text field that provides context to the <statusCode>. In contrast to <status>, this field is **not** meant to be seen by an end-user, but can be useful when scanning through logs of MOS communication.

There are 7 reserved codes:

Status Code	Status Literal	Meaning
100	COMPLETED	The item has been played on air
110	AIRING	The item is currently on air
120	PREPARED	The device has prepared an item to be aired
200	READY	The item has content that is ready to be aired
300	TRANSFERRING	The item is being transferred to the device
400	NOT READY	No content is available to be aired
410	PLACEHOLDER	Object hasn't yet been fulfilled
500	UNSUPPORTED	The content in this item cannot be aired

Additional codes within the ranges can be standardized and reserved in the future. All non-reserved codes may be used by vendors for proprietary statuses.

Devices are encouraged to send, at a minimum, 200 (READY) for playable content and 400 (NOT READY) for non-playable content. However, it is the device's discretion as to what status codes, if any, will be reported. It is also the devices responsibility to adhere to the semantic meaning of each code and ensure the accompanying status message is cohesive.

NCS's are encouraged to treat the playable range as positive, non-playable as negative, and transitioning as neutral, when presenting statuses to users. However, it is the discretion of the NCS as to how a status code will affect user feedback.

With the addition of these two fields, the <status> field is now free to provide more detail for end-users. For instance, device specific or translated messages can now be displayed in the NRCS.

## Example

### Video of kitten up a tree - Success

#### Placeholder created

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>1</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
<itemID>1</itemID>
<objID>kitten1</objID>
<status>PLACEHOLDER</status>
<statusCode>410</statusCode>
<statusLiteral>PLACEHOLDER</statusLiteral>
</roElementStat>
</mos>
```

#### Placeholder fulfilled and sending to the playout device

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>2</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
<itemID>1</itemID>
<objID>kitten1</objID>
<status>TRANSFERRING (25%)</status>
<statusCode>300</statusCode>
<statusLiteral>TRANSFERRING</statusLiteral>
</roElementStat>
</mos>
```

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>3</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
<itemID>1</itemID>
<objID>kitten1</objID>
<status>TRANSFERRING (75%)</status>
<statusCode>300</statusCode>
<statusLiteral>TRANSFERRING</statusLiteral>
</roElementStat>
</mos>
```

#### **Video on playout device**

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>4</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
<itemID>1</itemID>
<objID>kitten1</objID>
<status>READY</status>
<statusCode>200</statusCode>
<statusLiteral>READY</statusLiteral>
</roElementStat>
</mos>
```

#### **Video server has cued clip**

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>5</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
```

```
<itemID>1</itemID>  
<objID>kitten1</objID>  
<status>CUED</status>  
<statusCode>120</statusCode>  
<statusLiteral>PREPARED</statusLiteral>  
</roElementStat>  
</mos>
```

### **Video is airing**

```
<mos>  
<mosID>mam.mos</mosID>  
<ncsID>ncs.mos</ncsID>  
<messageID>6</messageID>  
<roElementStat element="ITEM">  
<roID>5PM</roID>  
<storyID>KITTEN UP A TREE</storyID>  
<itemID>1</itemID>  
<objID>kitten1</objID>  
<status>PLAY</status>  
<statusCode>110</statusCode>  
<statusLiteral>AIRING</statusLiteral>  
</roElementStat>  
</mos>
```

### **Video completed**

```
<mos>  
<mosID>mam.mos</mosID>  
<ncsID>ncs.mos</ncsID>  
<messageID>6</messageID>  
<roElementStat element="ITEM">  
<roID>5PM</roID>  
<storyID>KITTEN UP A TREE</storyID>  
<itemID>1</itemID>  
<objID>kitten1</objID>  
<status>DONE</status>  
<statusCode>100</statusCode>  
<statusLiteral>COMPLETED</statusLiteral>  
</roElementStat>  
</mos>
```

## Video of kitten up a tree - Failure

### Placeholder created

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>1</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
<itemID>1</itemID>
<objID>kitten1</objID>
<status>PLACEHOLDER</status>
<statusCode>410</statusCode>
<statusLiteral>PLACEHOLDER</statusLiteral>
</roElementStat>
</mos>
```

### Placeholder fulfilled but failed QC

```
<mos>
<mosID>mam.mos</mosID>
<ncsID>ncs.mos</ncsID>
<messageID>2</messageID>
<roElementStat element="ITEM">
<roID>5PM</roID>
<storyID>KITTEN UP A TREE</storyID>
<itemID>1</itemID>
<objID>kitten1</objID>
<status>QUALITY CHECK FAILED</status>
<statusCode>500</statusCode>
<statusLiteral>UNSUPPORTED</statusLiteral>
</roElementStat>
</mos>
```