

Von: Gregory Haynes

Gesendet: Mittwoch, 21. April 2021 15:08

An: Sebastian Beutel

Betreff: RE: Macro question: condition

Hello,

Unfortunately there is an issue with the cast expression (in this case "time:2") which means that they won't work inside a block. Some of the cast converters require to know the function that they are being passed in to so that they can convert to the correct type, this is done generally for all casts whether or not they actually end up needing it. When it tries to figure out what the function is all it has to go on is the macro step which since it contains a block does not just reference a single function and it ends up failing because of it.

I have come up with a alternative version that you could try which is based on declaring a variable inside the macro (see below). As it happens it is possible to set the value directly when the variable is declared for AcwTimeSpan objects but in other cases you may need to call SetProperty in an additional step before the other code. As all variables are global and macro variables are initialised on start up it is possible to change the value from another macro and then re-run the original macro and it will use the updated value (the second and third macros below).

When I was testing this I swapped out EnsurePlaybackCueSelected for SelectLiveCue, EnsurePlaybackCueSelected calls SelectLiveCue if there is no current cue selection but does nothing if there is. Possibly as a consequence of me having the Playback View window or how I recorded the playback, meant I had the dummy cue with ID zero selected which represents a cue to be appended (e.g. by tapping the bottom of the window when recording). This shouldn't be confused with the memory cue which is shown to have cue number zero but does have a valid, non-zero, cue ID. That being said there are some special cases in the software where it just handles this for memories as it knows there is only ever one cue however the CueFadeInTime works on any type of cue without really knowing what kind of playback it is from.

I can't think of any specific macro or WebAPI changes in 15.0 apart from some new functions exposed for Timeline (which mostly happens automatically as we write features). There is a small allocation of work in the schedule for some minor updates to macros however unfortunately it is the sort of thing which tends to get pushed later when other features or issues come up. I believe that part of this was to add menus and a mechanism so that a macro could request input from the user e.g. number, text or maybe even a handle selection.

Perhaps the best option is to send your ideas to JB or Paul, it isn't guaranteed but they have more control on what gets worked on than I do.

Hope you are well.

Gregory

```
<?xml version="1.0" encoding="utf-8"?>
<avolites.macros xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Avolites.Menus.xsd">
  <macro id="Avolites.Macros.SetFadeIn" name="Set Fade In">
    <variables>
      <object id="Time" type="Avolites.Acw.Titan.AcwTimeSpan"
assembly="Avolites.Acw.Titan" value="2" />
    </variables>
    <sequence>
      <step>
        {
          Handles.SetSourceHandle("PlaybackWindow", 0);
          if (Playbacks.IsCueHandle(Handles.SourceHandle) == true)
            {
              ActionScript.SetProperty("Playbacks.Editor.SelectedPlayback",
Handles.SourceHandle);
              Playbacks.Editor.SelectLiveCue();
              ActionScript.SetProperty("Playbacks.Editor.Times.CueFadeInTime",
Avolites.Macros.SetFadeIn.Time);
            }
          Handles.ClearSelection();
        }
      </step>
    </sequence>
  </macro>
  <macro id="Avolites.Macros.SetFadeIn2" name="Set Fade In 2">
    <sequence>
      <step>ActionScript.SetProperty("Avolites.Macros.SetFadeIn.Time", time:2)</step>
    </sequence>
  </macro>
  <macro id="Avolites.Macros.SetFadeIn3" name="Set Fade In 3">
    <sequence>
      <step>ActionScript.SetProperty("Avolites.Macros.SetFadeIn.Time", time:3)</step>
    </sequence>
  </macro>
</avolites.macros>
```

--
Gregory Haynes
Senior Developer
Avolites Limited

www.avolites.com



Please consider the environment before printing this e-mail

This email and any attachments to it may be confidential and are intended solely for the use of the individual to whom it is addressed. If you are not the intended recipient of this email, you must neither take any action based upon its contents, nor copy or show it to anyone. Please contact the sender if you believe you have received this email in error.

From: Sebastian Beutel
Sent: 19 April 2021 16:24
To: Gregory Haynes
Subject: Macro question: condition

Hi Gregory,

we haven't talked in ages – I hope you are doing fine!

I have a little macro question: I am trying to combine some steps which all are to be executed only if a condition is met, like this:

```
<step>Handles.SetSourceHandle("PlaybackWindow", 0)</step>
  <step>{
    if(Playbacks.IsCueHandle(Handles.SourceHandle) == true){
      ActionScript.SetProperty("Playbacks.Editor.SelectedPlayback",
Handles.SourceHandle);
      Playbacks.Editor.EnsurePlaybackCueSelected();
      ActionScript.SetProperty("Playbacks.Editor.Times.CueFadeInTime",
time:2);
    }
  }</step>
```

However this throws a null-reference exception. As I know that the individual commands are correct (they are executed if I omit the condition) there must be a syntax issue. Or is it impossible to have multiple commands in one condition like this altogether?

And another more general question: are there any changes regarding macros in v15, and is there something more planned with macros in the near future? I'd be happy to submit some ideas of mine 😊

Thanks in advance, best regards - Sebastian