

Von: Gregory Haynes

Gesendet: Mittwoch, 22. Dezember 2021 17:30

An: Sebastian Beutel

Betreff: RE: macro question: adding playback to playback group

Hello,

Sorry, I came across this old email and thought I would give a belated response.

There is a final implicit conversion step that occurs within the menu system when it is about to call the specified function. It looks at what the arguments have evaluated to be and if they don't match the types required by the function it tries to find a matching converter. If it is unable to find a converter to the destination type or the converter is unable to handle the inputted value it will fail and an error will occur.

Here are some examples where there is an implicit conversion happening:

```
PlaybackGroups.SetCurrentPlaybackGroupFromUserNumber(2)
```

In this case the function is expecting an AcwUserNumber so the integer is being converted to that.

```
Playbacks.FilterByPlaybackHandle(userNumber:2)
```

Here there is an explicit cast conversion from integer to AcwUserNumber but then an implicit conversion from that user number to IEnumerable<Handle>. If you didn't have the first cast it would assume that the number specified represented a Titan ID.

```
Handles.SetSourceHandleFromHandle("groupHandleUN=1")
```

Here there is an implicit conversion from a string to a handle, specifically a group handle however as this is a general method giving it a user number would not be enough for it to determine the correct type of handle.

You could add explicit casts to the above and even chain them for example:

```
Playbacks.FilterByPlaybackHandle(handle:userNumber:2)
```

Although this would still be implicitly converting from a single Handle to IEnumerable<Handle> although you could try adding handleList: to the beginning to do all three casts (they work from right to left when evaluating).

For reference global converters which are ones that can be used implicitly are currently as follows:

Destination Type	Cast prefix	Accepted Values
Handle	handle	integer (Titan ID), AcwUserNumber, string, HandleReference
IEnumerable<Handle>	handleList	IEnumerable<AcwUserNumber>, IEnumerable<int>, AcwUserNumber, integer, string (comma or semicolon separated)
AcwTimeSpan	time	integer/float (treated as seconds), string ("00:00", "00:00:00", "As In", "Global" etc.)
LevelAdjust	level	float (value between 0 and 1), integers (0 and 1) or string (that can be converted to float)
AcwUserNumber	userNumber	integer, float, string (ideally with full stop as decimal separator however comma may work in some cases)
IEnumerable<AcwUserNumber>	userNumberList	string (with comma separated list of user numbers with or without braces e.g. "{ 5, 6, 10, 11 }")
DmxAssignment	address	string ("1.2" for Universe.Address or "32" for absolute address), anything else will be converted to a string then parsed as that.
AcwColour	colour	string ("#000000", "000000"), integer (numeric value of hex string)

TimecodeTime		string ("00:00:00.00/25", all four parts of the time are required, frame rate can be omitted and defaults to 100fps)
--------------	--	--

I have realised while compiling the table above that you could get user numbers to a list of handles by doing:
`PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup, userNumberList:"{ 1, 2 }")`

Hope you have a great Christmas and a happy new year.

Gregory

From: Sebastian Beutel
Sent: 13 June 2021 16:09
To: Gregory Haynes
Subject: Re: macro question: adding playback to playback group

Hi Gregory,

thanks a lot for this really amazing and extensive answer, this explains really a lot!

I took some days to digest it and put this into the Wiki in various places. If you want to have a look you may start at <https://www.avolites.de/wiki/macros:example:playbackgroupcreateadd>, I also wrote something in <https://www.avolites.de/wiki/macros:identifier:usernumber>.

One minor question though: the API at <https://api.avolites.com/15.0/api/PlaybackGroups.SetCurrentPlaybackGroupFromUserNumber.html> states that the userNumber needs to be passed as AcwUserNumber. To my mind it is passed as Int32 like in <https://api.avolites.com/15.0/api/UserMacros.SetCurrentMacroFromUserNumber.html> (and possibly the other '...FromUserNumber()' functions as well). Or do you count

```
SetCurrentPlaybackGroupFromUserNumber(2)
```

as passing the AcwUserNumber (object) as well? I understand this is really a tiny detail for you but it might help other users to be clear with the formatting...

Again, thank you so much!

Best, Sebastian

Am 07.06.2021 um 16:15 schrieb Gregory Haynes:
Hello,

There was an attempt to make it possible to pass in lists (IEnumerable) of handles, this was at least partially successful. If you pass in just a single user number, integer (Titan ID) or a string e.g. "cueHandleUN=3" it will treat that with a list with one item in it.

Examples:

```
PlaybackGroups.AddPlaybacksToGroup(handle:"playbackGroupHandleUN=1", "cueHandleUN=3")
```

```
PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup,  
Playbacks.FilterByPlaybackHandle(userNumber:2))
```

Note that in the second example I didn't just put `userNumber:2` as the parameter to `AddPlaybacksToGroup` as it would assume the user number referred to a playback group (based on the namespace/class the function is in). The `FilterByPlaybackHandle` function will take a list of any kind of handle and only return the handles which are playback handles and as it is part of `Playbacks` when automatically converting the user number it assumes it belongs to a playback.

When passing in multiple values they must all be in one string which is then split up, if the item is an integer number these will be treated as Titan IDs. You can also, at least in macros, pass in the string values as well. Over WebAPI some of the separator characters get converted so it becomes impossible to tell the difference between the equals sign and the comma (or semi-colon). It is possible that to resolve the WebAPI issue we might become stricter about which characters can be used where and perhaps do less conversions internally.

Examples:

```
PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup,  
"{cueHandleUN=3,cueListHandleUN=1}")
```

```
PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup, "{ 1812; 1999  
}")
```

At least currently the braces are optional although they probably help to show intent. The second example is referencing Titan IDs and can have whitespace however in the first example any whitespace before the handle type will cause the lookup to fail (because it doesn't trim it and attempts to find a matching string including any spaces). I haven't found any way of specifying user numbers directly (without the string notation so things such as `"{ userNumber:3 }"` do not work.

Specifying the playback group is slightly tricky at the moment: as it stands it will not attempt to automatically convert strings or user numbers in that parameter. You can either use the handle cast with a string as in the example at the very top or use the `PlaybackGroups.CurrentPlaybackGroup` property like this:

```
PlaybackGroups.SetCurrentPlaybackGroupFromUserNumber(2)  
PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup,  
"cueHandleUN=3")
```

It is not currently possible to lookup a playback group from just a user number or Titan ID. Even though in principle it should be able to determine the handle type as this is specifically defined in the function this is not currently used by the parser.

Another example, this time creating a new playback group:

```
Handles.SetSourceHandleRange("Playbacks", {0, 1})  
PlaybackGroups.CreatePlaybackGroupWithPlaybacks("Test",  
Playbacks.FilterByPlaybackHandle(Handles.ContextHandles))
```

One reason for including this was the syntax of the handle indexes which you might notice is not a string in quotes. This has actually existed for a long time within the menu system itself as opposed to the parsers and converters that are used by macros and WebAPI. This syntax is specific to a list of literal integers such as handle indexes so you couldn't nest anything else inside the braces. This is another example where the `FilterByPlaybackHandle` is required because even if the source handles (`ContextHandles`) are all playback handles it is necessary to filter them otherwise the `CreatePlaybackGroupWithPlaybacks` thinks that they might be wrong and fails immediately.

Gregory

--

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: 04 June 2021 11:09
To: Gregory Haynes
Subject: macro question: adding playback to playback group

Hi Gregory,

another question: is it possible to add playbacks to playback groups via macro?

There is the function `PlaybackGroups.AddPlaybacksToGroup` but this requires the playbacks to be added as `IEnumerable` which afaik I cannot set within a macro. Hence, is there a way to achieve this at all, maybe with another function? Or, can I now set an `IEnumerable` in a macro?

Thanks, Sebastian