Namespaces

If you have already had a glance at the online API documentation then you know that there really is a huge number of functions and properties - currently there are almost 3000 (three-thousand) entries in the API. Not even the most professional programmer could know all these by heart.

Luckily there is a simple but useful system you stay on the helm of this maze: a hierachical system of names. This is used for functions, for properties, for macro IDs - and possibly for even more purposes. Simply put, you prepend each function/property/macro name with the space in that it lives. One abstract but simple example is in The Syntax of Functions:

> in our situation with hundreds of functions it's very helpful that functions are classified in namespaces. Thus you immediately see the difference between `playbacks.delete()` - which probably deletes a playback - and `fixtures.delete()` - which probably deletes a fixture

And if it gets more complicated, simply more namespaces are nested. This immediately explains something like

```
Playbacks.Editor.CopyCues.SelectCue
```

A rather normal human being would say, “select a cue to copy, when editing playbacks”

or, in dry software documentation language:

```
Namespace: Playbacks.Editor.CopyCues
```

There is a little caveat which isn't that grave then again:

Sometimes, instead of nesting more namespaces, one would find it more logical to regard some functions as sub functions. But then again, if instead of separating namespace and name you'd see the whole thing as 'full name' then you'll find your way anyway: the namespace gives you a rough indication in which area of the software this function/property might reside.

An example for this dilemma: the family of functions `ActionScript.SetProperty` :

- there is the main function `ActionScript.SetProperty()` – namespace ActionScript, function name SetPropery
- there is another function `ActionScript.SetProperty.Integer()` – namespace ActionScript.SetProperty, function name Integer - I'd rather tell this a 'sibling' or the main function, and let it live in the same namespace
- the same is true for `ActionScript.SetProperty.Boolean` , `ActionScript.SetProperty.String` and a few more (and some in some other namespaces...)

But again, don't bother too much. The main thing you need to keep in mind is that namespaces should help you find what you're looking for - as like as a book's index.

Namespaces in this Wiki

Here is a list of the namespaces covered in this wiki:

this namespace doesn't exist: namespace