IEnumerable

This is a kind of a collection of objects where the system can walk through. E.g. when multiple handles are selected and you want to apply a certain function to each of them, IEnumerable is probably the type this functions needs to cope with.

Status 2018

There is no way to define the contents of an IEnumerable from a macro - you can only copy an existing IEnumerable. See http://forum.avolites.com/viewtopic.php?f=20&t=5787:

> There is no syntax to define an IEnumerable in a macro so you can only use these functions if you have property that contains or method that returns that type. For example the following would be valid but it would only allow you to select fixtures which conflict with a previous exchange operation:
>
> ```
> Handles.SetSourceHandleRangeFromHandles(Handles.GetHandles(Fixtures.ConflictingHandles), false)
> ```

Example Patch - Freeze selected fixtures:

```
<step>Programmer.Editor.Selection.GetSelectedHandles("Windows.PatchView.Handles")</step>
<step>Programmer.Editor.Fixtures.Patch.FreezeFixtures(Windows.PatchView.Handles, True)</step>
```

The first step puts the currently selected fixtures into a specific property (in that case, “Windows.PatchView.Handles”).
The second step applies a function - which is capable of handling an IEnumerable - to this property. In this case, it sets all fixtures which are in this property to 'frozen'.

Status 2021 (v15)

> Gregory Haynes:
> 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 (this functions requires the second parameter as IEnumerable):
>
> `PlaybackGroups.AddPlaybacksToGroup(handle:“playbackGroupHandleUN=1”, “cueHandleUN=3”)` ⇒ one element as user number
>
> `PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup,`

> `“{cueHandleUN=3,cueListHandleUN=1}”)` ⇒ two elements by user number. No space allowed!
>
> `PlaybackGroups.AddPlaybacksToGroup(PlaybackGroups.CurrentPlaybackGroup, “{ 1812; 1999 }”)` ⇒ two elements by TitanID
>
> `Handles.SetSourceHandleRange(“Playbacks”, {0, 1})` ⇒ two elements by location, current page only

Also used in

- Patch - Freeze selected fixtures
- Handles.SetSourceHandleRangeFromHandles
- PlaybackGroups.AddPlaybacksToGroup
- PlaybackGroups.CreatePlaybackGroupWithPlaybacks
- Playbacks.FilterByPlaybackHandle
- Programmer.Editor.Fixtures.Patch.FreezeFixtures
- Programmer.Editor.Fixtures.Patch.InvertFixtureControls
- Programmer.Editor.Selection.GetSelectedHandles
- Handles.ContextHandles
- Windows.PatchView.Handles
- Types