

Bug 23949 - Improved functionality and performance for array manipulation

Status: HIBERNATED

Reported: 2022-09-20 04:53 EDT by Anton H.

Version: .Upcoming

Modified: 2022-09-27 11:46 EDT

CC List: 1 user ([show](#))

See Also:

Depends on:

Engine: IDE

Blocks:

Desktop OS: Windows

[Show dependency tree / graph](#)

Windows Version: 7

Attachments

[Add an attachment](#) (proposed patch, testcase, etc.)

Anton H. 2022-09-20 04:53:11 EDT

[Description](#)

Arrays in LC are synonymous to sheer power. The more you use them the more you love them and the more you HATE REPEAT LOOPS.

It has always puzzled me why there haven't been some implementation to take very basic array manipulation functions to a lower code level so that slow repeat loops in scripting are avoided for such basic but crucial and frequent tasks.

Let's say you have an `myArray[1]...[1000]` and want to delete some keys inbetween (i.e. delete `myArray[500]`). To keep the array consistent you must renumber the keys and for this you need to pass it through a repeat loop. This is both slow and a PIA as hell every single time. The same goes for sorting arrays.

It would be such a life saver if a compiled low level function could do that (i.e. something like:

```
renumber myArray,startValue,numberFormat  
or  
renumber myArray["myApples"],startValue,numberFormat
```

where startValue would be the start number of the index, i.e. 0 or 1 or any other number.

where numberformat would be the number of preceding zeros i.e. "###".
)

The other think I miss terribly is how to get rid of some subkey in a multi dimensional array.

Let's say you have `MyArray[1...n]["MyApples"], MyArray[1...n]["MyBananas"], MyArray[1...n]["MyKiwis"]` and you want to get rid of all ["MyKiwis"] keys in all `MyArray[1...n]...` this is such a pain needing a slow repeat loop for that!

I've tried with the union, intersect, difference, symmetric difference commands but - unfortunately - in this use case these commands also need to be run in a repeat loop if the array is multi dimensional.

It would be so damn cool if a compiled low level function could do that in one go (i.e. something like:

```
delete myArray,tDimensionLevel,tKey
```

In the above use case the command would look like:

```
delete myArray,2,"MyKiwis"
```

which would delete all "MyKiwis" keys (and their subkeys) from EVERY preceding array key.

Best regards
Tony

Panos Merakos 2022-09-27 11:46:32 EDT

[Comment 1](#)

Hello Tony,

Thank you for the report.

Indeed, all these would be a useful addition to the LC engine syntax.

We will update you when the status of this report changes.

Kind regards,
Panos
--

Note

You need to [log in](#) before you can comment on or make changes to this bug.
