I hereby present my latest project: **True Binary Array support for AHK.** Besides from what u'd expect of binary Arrays, the arrayobject will return the string ***Array()*** if being used form a standard AHK String processing command such as MsgBox.

Be carefull with that: commands such as VarSetCapacity might destroy the length due to binary zeros within and render the Object unusable. See Lexikos comment below for additional informations.

List of currently supported functions

A_Put((ByRef) *Array*, (ByRef) *Data* [, (int) Index, (int) dSize])
Stores given *data* into given *Array*. Creates the Arraystructure if neccessary. Returns elements index in *Array* on success

A_Get((ByRef) *Array*, (int) Index)
Returns Element and stores Size as ErrorLevel

A_Implode((ByRef) *Array*, (str) glue)
Returns a joined string of given *Array* and glue parameter. Additionally Length is returned as ErrorLevel

A_Explode((ByRef) *Array*, (str) delimiterString, (str) sourceString
[, (int) Limit, (str) trimChars, (bool) trimCharsIsRegEx, (bool) dStringIsRegEx])

Returns an *Array* of strings, each of which is a substring of sourceString formed by splitting it on boundaries formed by the delimiterDtring. Unlike Stringsplit, multiple chars are allowed in delimiterString to be used as a seperator.

!! ATENTION - Use trimCharsIsRegEx at own risk !!
By setting trimCharsIsRegEx to true, you might use directly more complex RegEx to trim Chars off the Element to insert

!! ATENTION - Use dStringIsRegEx at own risk !!
By setting dStringIsRegEx to true, you might use directly more complex RegEx to delimit a given String into subsets

A_Del((ByRef) *Array* , (int) Item)
Deletes ItemIndex of given *Array*

A_Pop((ByRef) *Array*)
Returns the element off the end of *Array* and removes it. Additionally length is returned as ErrorLevel

A_Shift((ByRef) *Array*)
Shift an element off the beginning of *Array* and returns it. Additionally length is returned as ErrorLevel

A_Swap((ByRef) *Array*, (int) Index_A, (int) Index_B)
Swaps Index_A's element with Index_B's element in given Array

A_Slice((ByRef) *Array*, (ByRef) *SourceArray*, (int) Start, (int) End)
SourceArray's given intersection Elements are appended to *Array*, which will be created at runtime if neccessary

A_Merge((ByRef) Array, (ByRef) *SourceArray*)
Appends entire *SourceArray* to *Array*. Returns A_Count of *Array*, -1 on Error with Details in ErrorLevel

A_Array((ByRef) *Array*)
Returns TRUE if *Array*, FALSE otherwise - Thx, Lexikos

A_Count((ByRef) *Array*)
Returns current element count of given *Array* or -1 if no valid binary array

A_Dump((ByRef) *Array*)
Dumps existing *Array* into Human readable List
DOES NOT SUPPORT NESTED ARRAYS FOR NOW

List of currently used internal functions

A_Init((ByRef) *Array*)
for internal use only - initialises bytestructure of *Array*

A_Size((ByRef) *Array*)
for internal use only - returns ArrayStructuresize

A_Length((ByRef) *Array*)
for internal use only - returns DataLength of *Array*

A_ArrayMM((ptr) Target, (ptr) Source, (uint) Length)
for internal use only - SyntaxSugar for RtlMoveMemory

List of currently used debugging functions for RC2

A___ArrayBin((ByRef) *Array*, (uint) Offset, (uint) Length)
For debugging purposes only - dumps Array content in hexView

A___ArrayInsideView(*Array*)
For debugging purposes only - to dump an *Array* use A_Dump(Array)
This function returns a view on table setup with relative offsets

below is a lil testsuite which also introduces the functions. ATM its not a release but a release candidate.

Code (Expand):

```
; tst-a_array.ahk

; TestSuite for AHK's Binary Array RC2
; (w) by derRaphael / released under the WTFPL 1.0
;       see http://sam.zoy.org/wtfpl/ for details.

Loop,5 ; Add 5 elements to Array
   VarSetCapacity(t,5,asc("0")+A_Index), A_Put(MyArray,t)

; Display AHK array Signature
MsgBox,0,DirectAccess Array,% MyArray

Data := "aaaa", A_Put(MyArray, Data, 1) ; Change 1st element w/ shorter size
Data := "bbbbbbbbb", A_Put(MyArray, Data, 3) ; ; Change 3rd element w/ longer size
```