Array Notation RC1

Adám Brudzewsky

♬ The saga continues…

Array Notation RC1

Adám Brudzewsky

Why now?

- Avoiding complex expressions when constructing arrays
  Might not fit comfortably on a single line
- Using array definitions with source code management
  These tend to handle changes on a line-by-line basis
- Arrays in text form
  Edit with any editor, email, transfer, create with 3rd party tools...

What?

- Medium sized arrays

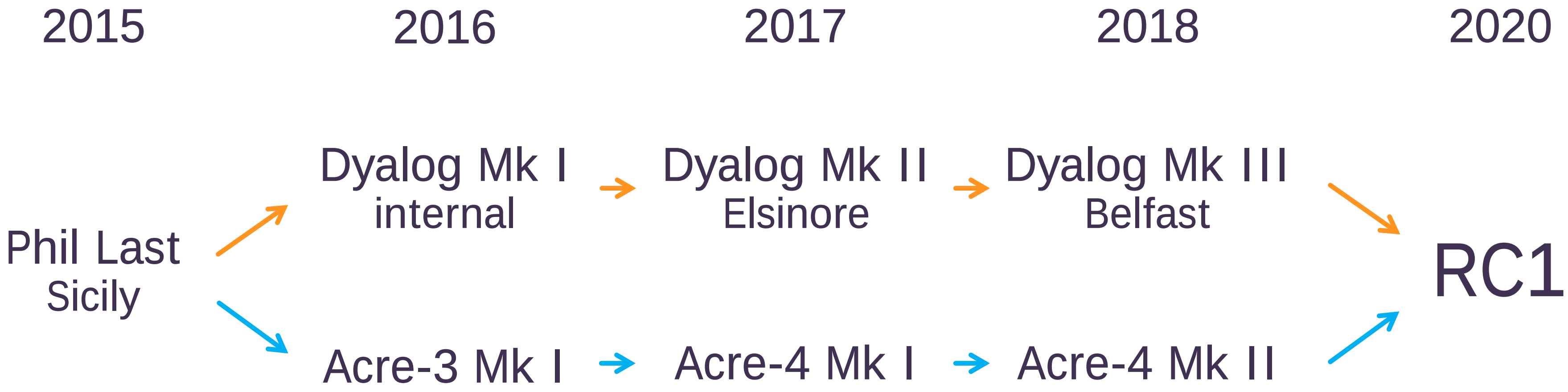
  Empty and trivial arrays are better done as expressions

- Higher rank arrays

  We have good notations for vectors and small vectors of vectors

- Depth deeper than 2

RC1‽



2015 2016 2017 2018 2020

Phil Last
Sicily

Dyalog Mk I
internal

Dyalog Mk II
Elsinore

Dyalog Mk III
Belfast

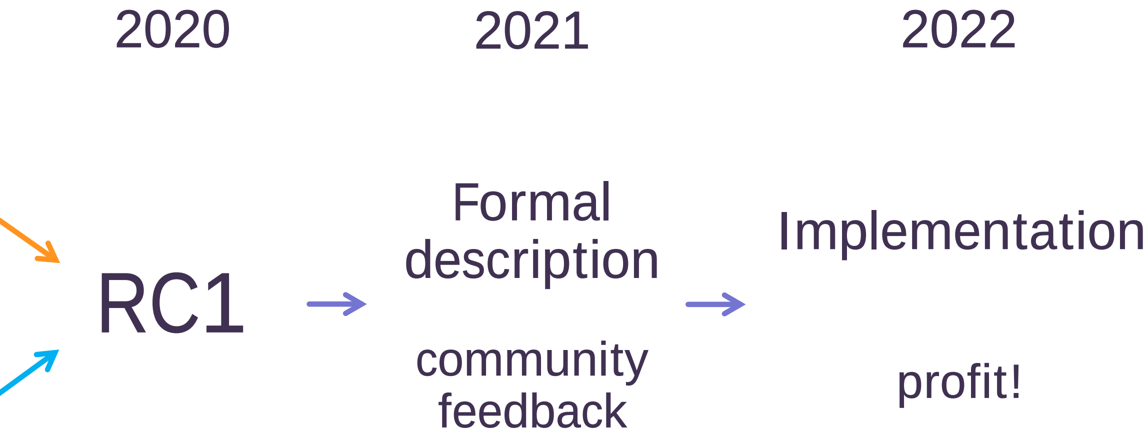
Acre-3 Mk I

Acre-4 Mk I

Acre-4 Mk II

RC1

RC1‽



2020

2021

2022

RC1

Formal description

community feedback

Implementation

profit!

Where?

- Link

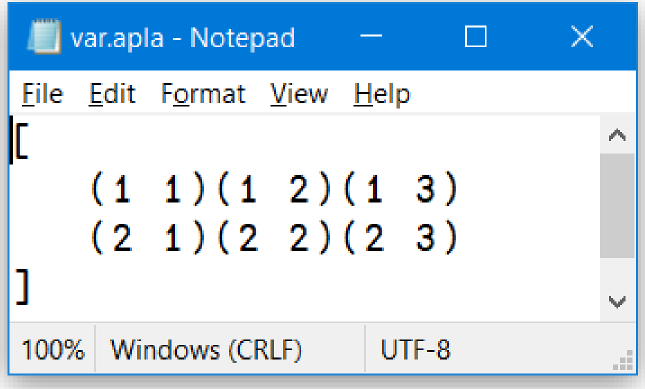
  part of your Dyalog installation

- Acre

  github.com/the-carlisle-group/Acre-Desktop

Acre

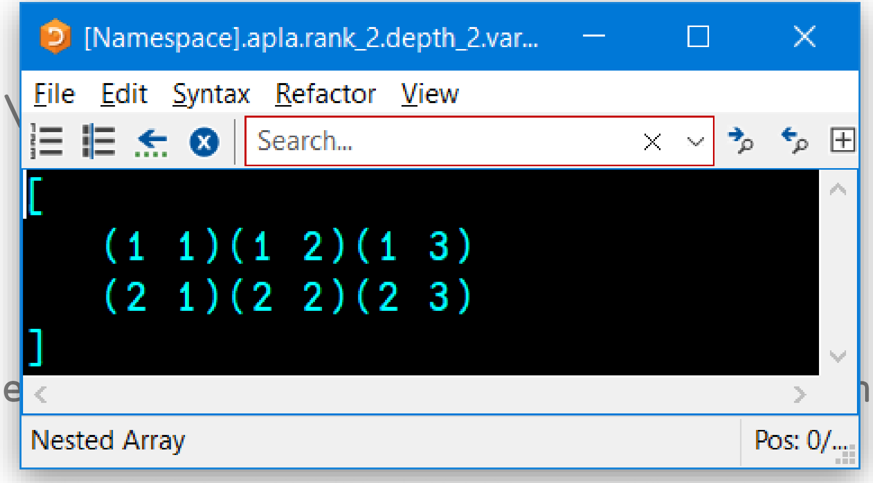
```
      ⎕←var←⍳2 3
┌───┬───┬───┐
│1 1│1 2│1 3│
├───┼───┼───┤
│2 1│2 2│2 3│
└───┴───┴───┘
      ]CreateProject C:\tmp\acretest #
#
      ]SetChanged var
┌─────┐
│#.var│
└─────┘
      ]Open "C:\tmp\acretest\APLSource\var.apla" -using=notepad
```



```
[
    (1 1)(1 2)(1 3)
    (2 1)(2 2)(2 3)
]
```

Acre

```
      ⎕←var←⍳2 3
┌───┬───┬───┐
│1 1│1 2│1 3│
├───┼───┼───┤
│2 1│2 2│2 3│
└───┴───┴───┘
      ]CreateProject C:\
#
      ]SetChanged var
┌─────┐
│#.var│
└─────┘
      ]Open "C:\tmp\acre                              notepad
      ]EditArray var
```


[Namespace].apla.rank_2.depth_2.var...
File Edit Syntax Refactor View
Search...

```
[
    (1 1)(1 2)(1 3)
    (2 1)(2 2)(2 3)
]
```

Link

```
      ]Link.Create # C:\tmp\acretest\APLSource
Linked: # ←→ C:\tmp\acretest\APLSource
      var
┌───┬───┬───┐
│1 1│1 2│1 3│
├───┼───┼───┤
│2 1│2 2│2 3│
└───┴───┴───┘
      ⎕SE.Link.Serialise var
[(1 1)  (1 2)  (1 3)
 (2 1)  (2 2)  (2 3)]
```

Link

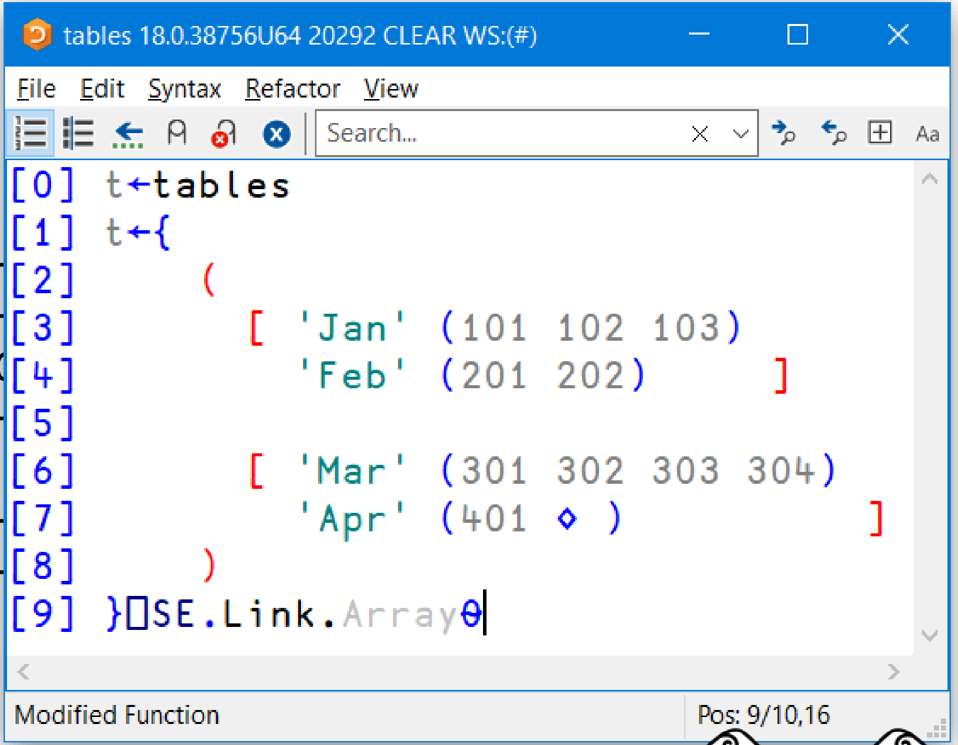
```
      ↑ s←'([1 2 ⋄ 3 4]' ' [5 6 ⋄ 7 8])'
([1 2 ⋄ 3 4]
 [5 6 ⋄ 7 8])

      ⎕SE.Link.Deserialise s
┌───┬───┐
│1 2│5 6│
│3 4│7 8│
└───┴───┘

      {['ABC' ⋄ 'DEF']}⎕SE.Link.Array⍬
ABC
DEF
```

Link

```
      )ed tables
      tables
┌──────────────────┬────────────────
│┌───┬───────────┐ │┌───┬───────────
││Jan│101 102 103│ ││Mar│301 302 30
│├───┼───────────┤ │├───┼───────────
││Feb│201 202    │ ││Apr│401
│└───┴───────────┘ │└───┴───────────
└──────────────────┴────────────────
```


tables 18.0.38756U64 20292 CLEAR WS:(#)
File Edit Syntax Refactor View
Search...
Aa
Modified Function
Pos: 9/10,16

```
 t←tables
 t←{
      (
         [ 'Jan' (101 102 103)
           'Feb' (201 202)       ]

         [ 'Mar' (301 302 303 304)
           'Apr' (401 ⋄ )           ]
      )
 }⎕SE.Link.Array⍬
```

Tutorial

- New: parentheses and brackets containing more than one expression
- New: parentheses containing zero expressions
- Expressions can be separated by a line break or a ⋄
- Round parentheses: `( a ⋄ b ⋄ c )`

  Each non-blank expression becomes an element in a new vector
- Square brackets: `[ a ⋄ b ⋄ c ]`

  Each non-blank expression becomes a major cell (of rank≥1) in a new array
- Round parentheses that are empty or contain at least one `:`

  Each expression is a name-value pair separated by a `:`

dfns.dws: cal

```
Q1←'January' 'February' 'March    '~¨' '     ⍝ 1st quarter month names.
Q2←'April  ' 'May     ' 'June     '~¨' '     ⍝ 2nd   ..     ..   ..
Q3←'July   ' 'August  ' 'September'~¨' '     ⍝ 3rd   ..     ..   ..
Q4←'October' 'November' 'December '~¨' '     ⍝ 4th   ..     ..   ..
months←Q1,Q2,Q3,Q4                          ⍝ month names for year.
```

dfns.dws: cal

```
months←(                                    ⍝ month names for year.
    'January'◇'February'◇'March'            ⍝ 1st quarter month names.
    'April'  ◇'May'     ◇'June'             ⍝ 2nd   ..      ..    ..
    'July'   ◇'August'  ◇'September'        ⍝ 3rd   ..      ..    ..
    'October'◇'November'◇'December'         ⍝ 4th   ..      ..    ..
)
```

dfns.dws: morse

```
{⍵~¨' '}\¨{          ⍝ plain-text and Morse codes.
 ('A' ' .-   ')('B' ' -... ')('C' ' -.-. ')('D' ' -..  '),⍵}{
 ('E' ' .    ')('F' ' ..-. ')('G' ' --.  ')('H' ' .... '),⍵}{
 ('I' ' ..   ')('J' ' .--- ')('K' ' -.-  ')('L' ' .-.. '),⍵}{
 ('M' ' --   ')('N' ' -.   ')('O' ' ---  ')('P' ' .--. '),⍵}{
 ('Q' ' --.- ')('R' ' .-.  ')('S' ' ...  ')('T' ' -    '),⍵}{
 ('U' ' ..-  ')('V' ' ...- ')('W' ' .--  ')('X' ' -..- '),⍵}{
 ('Y' ' -.-- ')('Z' ' --.. '),⍵}{

 ('0' ' -----')('1' ' .---- ')('2' ' ..--- ')('3' ' ...-- '),⍵}{
 ('4' ' ....-')('5' ' ..... ')('6' ' -.... ')('7' ' --... '),⍵}{
 ('8' ' ---..')('9' ' ----. '),⍵}{

 ('.' ' .-.-.- ')(',' ' --..-- ')(':' '---... '),⍵}{
 ('?' ' ..--.. ')('''' ' .----. ')('-' '-....- '),⍵}{
 ('/' ' -..-.  ')('(' ' -.--.  ')(')' '-.--.- '),⍵}{
 ('"' ' .-..-. ')('@' ' .--.-. ')('=' '-...- '),⍵}{

 ⍵}⊂' ' ' / '        ⍝ blank / inter-word separator.
```

dfns.dws: morse

```
(                    ⍝ plain-text and Morse codes.
  'A'  '.-'    ⋄ 'B'  '-...'  ⋄ 'C'  '-.-.'  ⋄ 'D'  '-..'
  'E'  '.'     ⋄ 'F'  '..-.'  ⋄ 'G'  '--.'   ⋄ 'H'  '....'
  'I'  '..'    ⋄ 'J'  '.---'  ⋄ 'K'  '-.-'   ⋄ 'L'  '.-..'
  'M'  '--'    ⋄ 'N'  '-.'    ⋄ 'O'  '---'   ⋄ 'P'  '.--.'
  'Q'  '--.-'  ⋄ 'R'  '.-.'   ⋄ 'S'  '...'   ⋄ 'T'  '-'
  'U'  '..-'   ⋄ 'V'  '...-'  ⋄ 'W'  '.--'   ⋄ 'X'  '-..-'
  'Y'  '-.--'  ⋄ 'Z'  '--..'

  '0'  '-----' ⋄ '1'  '.----' ⋄ '2'  '..---' ⋄ '3'  '...--'
  '4'  '....-' ⋄ '5'  '.....' ⋄ '6'  '-....' ⋄ '7'  '--...'
  '8'  '---..' ⋄ '9'  '----.'

  '.'  '.-.-.-' ⋄ ','  '--..--' ⋄ ':'  '---...'
  '?'  '..--..' ⋄ ''''  '.----.' ⋄ '-'  '-....-'
  '/'  '-..-.'  ⋄ '('  '-.--.'  ⋄ ')'  '-.--.-'
  '"'  '.-..-.' ⋄ '@'  '.--.-.' ⋄ '='  '-...-'

  ' '  ' / ' )       ⍝ blank / inter-word separator.
```

dfns.dws: morse

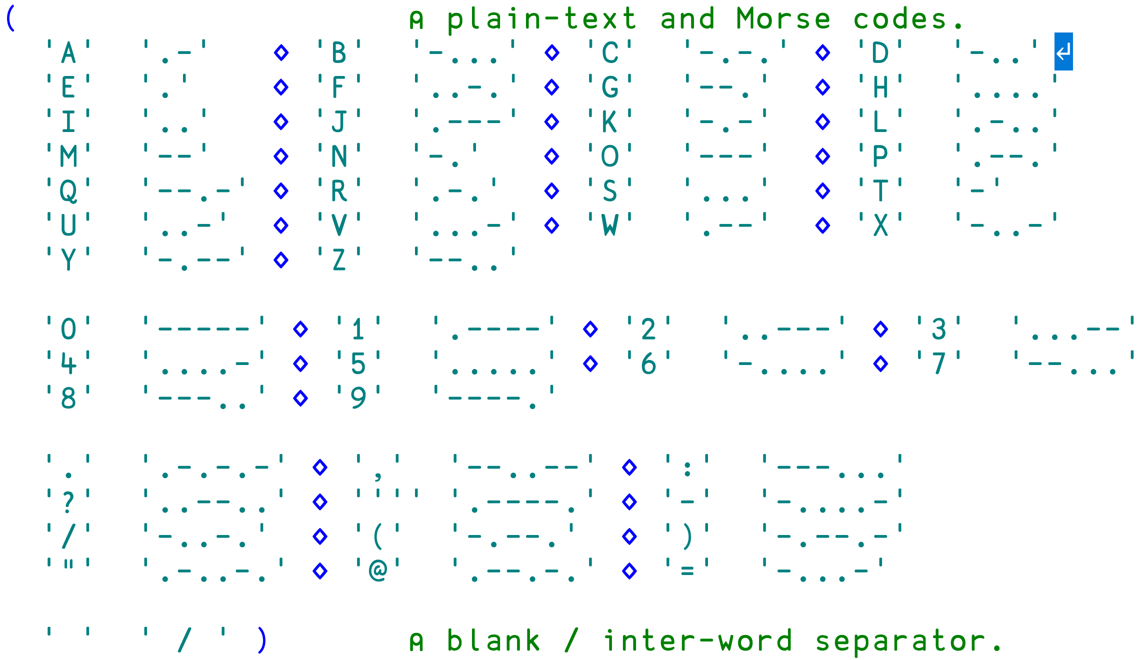
```
(                    ⍝ plain-text and Morse codes.
  'A'  '.-'   ⋄ 'B'  '-...' ⋄ 'C'  '-.-.' ⋄ 'D'  '-..'
  'E'  '.'    ⋄ 'F'  '..-.' ⋄ 'G'  '--.'  ⋄ 'H'  '....'
  'I'  '..'   ⋄ 'J'  '.---' ⋄ 'K'  '-.-'  ⋄ 'L'  '.-..'
  'M'  '--'   ⋄ 'N'  '-.'   ⋄ 'O'  '---'  ⋄ 'P'  '.--.'
  'Q'  '--.-' ⋄ 'R'  '.-.'  ⋄ 'S'  '...'  ⋄ 'T'  '-'
  'U'  '..-'  ⋄ 'V'  '...-' ⋄ 'W'  '.--'  ⋄ 'X'  '-..-'
  'Y'  '-.--' ⋄ 'Z'  '--..'

  '0'  '-----' ⋄ '1'  '.----' ⋄ '2'  '..---' ⋄ '3'  '...--'
  '4'  '....-' ⋄ '5'  '.....' ⋄ '6'  '-....' ⋄ '7'  '--...'
  '8'  '---..' ⋄ '9'  '----.'

  '.'  '.-.-.-' ⋄ ','  '--..--' ⋄ ':'  '---...'
  '?'  '..--..' ⋄ ''''  '.----.' ⋄ '-'  '-....-'
  '/'  '-..-.'  ⋄ '('  '-.--.'  ⋄ ')'  '-.--.-'
  '"'  '.-..-.' ⋄ '@'  '.--.-.' ⋄ '='  '-...-'

  ' '  ' / ' )       ⍝ blank / inter-word separator.
```

dfns.dws: morse



```
(                    ⍝ plain-text and Morse codes.
  'A'  '.-'   ⋄ 'B'  '-...' ⋄ 'C'  '-.-.' ⋄ 'D'  '-..' ↵
  'E'  '.'    ⋄ 'F'  '..-.' ⋄ 'G'  '--.'  ⋄ 'H'  '....'
  'I'  '..'   ⋄ 'J'  '.---' ⋄ 'K'  '-.-'  ⋄ 'L'  '.-..'
  'M'  '--'   ⋄ 'N'  '-.'   ⋄ 'O'  '---'  ⋄ 'P'  '.--.'
  'Q'  '--.-' ⋄ 'R'  '.-.'  ⋄ 'S'  '...'  ⋄ 'T'  '-'
  'U'  '..-'  ⋄ 'V'  '...-' ⋄ 'W'  '.--'  ⋄ 'X'  '-..-'
  'Y'  '-.--' ⋄ 'Z'  '--..'

  '0'  '-----' ⋄ '1'  '.----' ⋄ '2'  '..---' ⋄ '3'  '...--'
  '4'  '....-' ⋄ '5'  '.....' ⋄ '6'  '-....' ⋄ '7'  '--...'
  '8'  '---..' ⋄ '9'  '----.'

  '.'  '.-.-.-' ⋄ ','  '--..--' ⋄ ':'  '---...'
  '?'  '..--..' ⋄ '''' '.----.' ⋄ '-'  '-....-'
  '/'  '-..-.'  ⋄ '('  '-.--.'  ⋄ ')'  '-.--.-'
  '"'  '.-..-.' ⋄ '@'  '.--.-.' ⋄ '='  '-...-'

  ' '  ' / ' )       ⍝ blank / inter-word separator.
```

dfns.dws: morse

```
(                    ⍝ plain-text and Morse codes.
  'A'  '.-'   ⋄ 'B'  '-...' ⋄ 'C'  '-.-.' ⋄ 'D'  '-..'
  'E'  '.'    ⋄ 'F'  '..-.' ⋄ 'G'  '--.'  ⋄ 'H'  '....'
  'I'  '..'   ⋄ 'J'  '.---' ⋄ 'K'  '-.-'  ⋄ 'L'  '.-..'
  'M'  '--'   ⋄ 'N'  '-.'   ⋄ 'O'  '---'  ⋄ 'P'  '.--.'
  'Q'  '--.-' ⋄ 'R'  '.-.'  ⋄ 'S'  '...'  ⋄ 'T'  '-'
  'U'  '..-'  ⋄ 'V'  '...-' ⋄ 'W'  '.--'  ⋄ 'X'  '-..-'
  'Y'  '-.--' ⋄ 'Z'  '--..' ↲
↲
  '0'  '-----' ⋄ '1'  '.----' ⋄ '2'  '..---' ⋄ '3'  '...--'
  '4'  '....-' ⋄ '5'  '.....' ⋄ '6'  '-....' ⋄ '7'  '--...'
  '8'  '---..' ⋄ '9'  '----.'

  '.'  '.-.-.-' ⋄ ','  '--..--' ⋄ ':'  '---...'
  '?'  '..--..' ⋄ '''' '.----.' ⋄ '-'  '-....-'
  '/'  '-..-.'  ⋄ '('  '-.--.'  ⋄ ')'  '-.--.-'
  '"'  '.-..-.' ⋄ '@'  '.--.-.' ⋄ '='  '-...-'

  ' '  ' / ' )       ⍝ blank / inter-word separator.
```

math.dws: Eigen

```
⌽{⍵,⊂' <C1   ' 'V'}{              ⍝ JOBZ
  ⍵,⊂' <C1   ' 'L'}{              ⍝ UPLO
  ⍵,⊂' <I4   'n}{                 ⍝ N
  ⍵,⊂' =F8[] '(∊⍉mat)}{           ⍝ A
  ⍵,⊂' <I4   'n}{                 ⍝ LDA
  ⍵,⊂' >F8[] 'n}{                 ⍝ W
  ⍵,⊂' >F8[] '(¯2+4×n)}{          ⍝ WORK
  ⍵,⊂' <I4   '(¯1+2×n)}{          ⍝ LWORK
  ⍵,⊂' >F8[] '(¯2+3×n)}{          ⍝ RWORK
  ⍵,⊂' >I4   '0}⍬                 ⍝ INFO
```

math.dws: Eigen

```
[ ' <C1 ' 'V'                  ⍝ JOBZ
  ' <C1 ' 'L'                  ⍝ UPLO
  ' <I4 '  n                   ⍝ N
  ' =F8[] '(∊⍉mat)             ⍝ A
  ' <I4 '  n                   ⍝ LDA
  ' >F8[] 'n                   ⍝ W
  ' >F8[] '(¯2+4×n)            ⍝ WORK
  ' <I4 '  (¯1+2×n)            ⍝ LWORK
  ' >F8[] '(¯2+3×n)            ⍝ RWORK
  ' >I4 '  0       ]           ⍝ INFO
```

Profile ucmd: DBMenuCB

```
poss←1 2⍴'fns'((0 1)(0.7 0)(0.7 0)×size)
poss⍪←'fnd'((0 1)(0 0)(0 0)×size)
poss⍪←'lines'((0 0)(0.7 0)(0.7 0)×size)
poss⍪←'lnd'((0 0)(0 0)(0 0)×size)
```

Profile ucmd: DBMenuCB

```
poss←['fns'  ((0.0 1 ⋄ 0.7 0 ⋄ 0.7 0)×size)
      'fnd'  ((0.0 1 ⋄ 0.0 0 ⋄ 0.0 0)×size)
      'lines'((0.0 0 ⋄ 0.7 0 ⋄ 0.7 0)×size)
      'lnd'  ((0.0 0 ⋄ 0.0 0 ⋄ 0.0 0)×size)]
```

```
[3 2⋄4 1][[2 1⋄2 2];[1 2⋄1 1]]
```

Profile ucmd: DBMenuCB

```
poss←['fns'  ((0.0 1 ⋄ 0.7 0 ⋄ 0.7 0)×size)
      'fnd'  ((0.0 1 ⋄ 0.0 0 ⋄ 0.0 0)×size)
      'lines'((0.0 0 ⋄ 0.7 0 ⋄ 0.7 0)×size)
      'lnd'  ((0.0 0 ⋄ 0.0 0 ⋄ 0.0 0)×size)]
```

```
[3 2⋄4 1][[2 1⋄2 2];[1 2⋄1 1]]
```

SALT: SettingsTable

```
UserFolder←'[HOME]',(WIN↓'/'),'MyUCMDs'
CmdDir←UserFolder,PATHDEL[1],SALTFOLDER,FS,'spice'
split←{1↓¨(s□□)□s←(□□1↑□)↓□,□}
:field shared SettingsTable←0 5□''
□ name; description; registry name; default; value
□ e.g. [ProgramFiles]BeyondCompare
SettingsTable□←'compare;the comparison program to use;CompareCMD;APL;'split';'
□ Cmd Folders are locations where Spice commands are stored - their existence is not challenged
□ 2nd folder is something like this: C:\Users\DanB2\Documents\Dyalog APL 14.0 Unicode Files
SettingsTable□←('cmddir□list of Spice folders (commands) to use separated by ',PATHDEL[1],'□CommandFolder□',CmdDir,'□')split'□'
SettingsTable□←'debug;debug level;DebugLevel;0;0' split';'
SettingsTable□←'editor;the editor program to use;EditorCMD;notepad;'split';'
SettingsTable□←'edprompt;whether the editor prompts for confirmation;EdPrompt;1;' split';'
SettingsTable□←'fndels;whether tradfns are saved enclosed in □s;FnDels;0;'split';'
SettingsTable□←'mapprimitives;whether to map some primitives to □Uxxxx on Classic;MapPrim;1;' split';'
□ automatic for all but PI
SettingsTable□←('newcmd;detection of new user commands;CmdDetect;',(□Pi□'auto' 'manual'),';') split';'
SettingsTable□←'track;saving of new items and which info to stored in SALT tags;Track;;' split';'
□ only APL and XML so far
SettingsTable□←'varfmt;whether variables are saved as XML docs or APL expressions;VarFmt;xml;' split';'
□ WorkFolders are locations where files are searched - their existence is not challenged
SettingsTable□←('workdir□list of storage directories to use (separated by ',PATHDEL[1],')□SourceFolder□',SALTFOLDER,'□')split'□'
```

SALT: SettingsTable

```
UserFolder←'[HOME]',(WIN↓'/'),'MyUCMDs'
CmdDir←UserFolder,PATHDEL[1],SALTFOLDER,FS,'spice'

:field shared SettingsTable←[
□ name          □ description                                                     □ registry name □ default         □ value
□ e.g. [ProgramFiles]BeyondCompare
'compare'       'the comparison program to use'                                   'CompareCMD'    'APL'             ''
□ Cmd Folders are locations where Spice commands are stored - their existence is not challenged
□ 2nd folder is something like this: C:\Users\DanB2\Documents\Dyalog APL 14.0 Unicode Files
'cmddir'        (list of Spice folders (commands) to use separated by ',PATHDEL[1])  'CommandFolder' CmdDir          ''
'debug'         'debug level'                                                     'DebugLevel'    '0'               '0'
'editor'        'the editor program to use'                                       'EditorCMD'     'notepad'         ''
'edprompt'      'whether the editor prompts for confirmation'                     'EdPrompt'      '1'               ''
'fndels'        'whether tradfns are saved enclosed in □s'                        'FnDels'        '0'               ''
'mapprimitives' 'whether to map some primitives to □Uxxxx on Classic'             'MapPrim'       '1'               ''
□ automatic for all but PI
'newcmd'        'detection of new user commands'                                  'CmdDetect'     (□Pi □'auto' 'manual') ''
'track'         'saving of new items and which info to store in SALT tags'        'Track'         ''                ''
□ only APL and XML so far
'varfmt'        'whether variables are saved as XML docs or APL expressions'      'VarFmt'        'xml'             ''
□ WorkFolders are locations where files are searched - their existence is not challenged
'workdir'       ('list of storage directories to use (separated by ',PATHDEL[1],')') 'SourceFolder' SALTFOLDER      ''
]
```

Link: DefaultOpts

```
(
  beforeRead:       ''
  beforeWrite:      ''
  caseCode:         0
  codeExtensions:   ( 'aplf'
                      'aplo'
                      'apln'
                      'aplc'
                      'apli'
                      'dyalog'
                      'apl'
                      'mipage'  )
  customExtensions: ''
  flatten:          0
  forceExtensions:  0
  forceFilenames:   0
  source:           'dir'
  typeExtensions:   [ 2   'apla'
                      3   'aplf'
                      4   'aplo'
                      9.1 'apln'
                      9.4 'aplc'
                      9.5 'apli'  ]
  watch:            'ns'
)
```

Link: Export

```
(defopts←⎕NS ⍬).(overwrite)←0
```

```
defopts←(overwrite:0)
```

Link: Import

```
opts←⎕NS ⍬
```

```
opts←()
```

Compete for prizes totaling €500

```
   ⎕SE.Link.Serialise ↓3 4⍴⎕A
( 'ABCD'
  'EFGH'
  'IJKL' )
```

```
   ⎕SE.Link.Serialise ↓2 3 2⍴⎕A
['AB'  'CD'  'EF'
 'GH'  'IJ'  'KL']
```

```
   ⎕SE.Link.Serialise ⍳¨⍳3
( ( 1 ⋄)
  1 2
  1 2 3  )
```

```
   ⎕SE.Link.Serialise ↓2 3 2⍴⍳10
[(1 2)  (3 4)   (5 6)
 (7 8)  (9 10)  (1 2)]
```

Compete for five prizes of €100 each

```
    ⎕SE.Link.Serialise  ⍪↓⍪⎕TC
[ (( (⎕UCS 8) ⋄ ) ⋄ )
  (( (⎕UCS 10) ⋄ ) ⋄ )
  (( (⎕UCS 13) ⋄ ) ⋄ ) ]
```

⇩

```
[((⎕UCS  8 ⋄ ) ⋄ )
 ((⎕UCS 10 ⋄ ) ⋄ )
 ((⎕UCS 13 ⋄ ) ⋄ )]
```

```
    ⎕SE.Link.Serialise  ↓↓2 3 2⍴⍳10
( ( 1 2 ⋄ 3 4 ⋄ 5 6 )
  ( 7 8 ⋄ 9 10 ⋄ 1 2  ) )
```

⇩

```
((1 2 ⋄ 3  4 ⋄ 5 6)
 (7 8 ⋄ 9 10 ⋄ 1 2))
```

Matrices & higher-rank arrays

Multi-line   Inline   Expression

```
[1 2
 3 4     ⇔     [1 2 ⋄ 3 4 ⋄ 5 6]     ⇔     3 2⍴1 2 3 4 5 6
 5 6]

[1
 2       ⇔     [1 ⋄ 2 ⋄ 3]           ⇔     3 1⍴1 2 3
 3]
```

Vectors & nested arrays

```
   Multi-line             Inline                     Expression

   (1 2
    3 4        ⇔     (1 2 ⋄ 3 4 ⋄ 5 6)     ⇔     (1 2)(3 4)(5 6)
    5 6)

    (1
     2         ⇔        (1 ⋄ 2 ⋄ 3)        ⇔          1 2 3
     3)
```

Namespaces

Multi-line

```
(
   a:'APL'
   b:c,c←1 2
)
```

⇔

Inline

```
(a:'APL' ⋄ b:c,c←1 2)
```

⇔

Expression

```
{
   ⍺←⎕NS⍬
   ⍺.a←'APL'
   ⍺.b←{
      c,c←1 2
   }⍬
   ⍺
}⍬
```

```
(
 )
```

⇔

```
()
```

⇔

```
⎕NS⍬
```

Questions?

DYALOG

2020

♫ To be continued…

```
[1 2 ⋄ 3 4 ⋄ 5 6]
(1 2 ⋄ 3 4 ⋄ 5 6)
(a:'APL' ⋄ b:c,c←1 2)
```

aplwiki.com/wiki/array notation
Adám Brudzewsky
adam@dyalog.com