DYALOG

Mainz, April 30th 2024

Array Notation

Morten Kromberg, CTO

Dyalog – The Next Generation

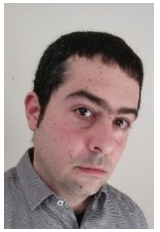
2010-2021





2022



2023





2024

2023 Summer Interns


Dyalog – The Next Generation

2010-2021

2022

2023

2024

2023 Summer Interns

A Way to Write Most Arrays ***Literally***

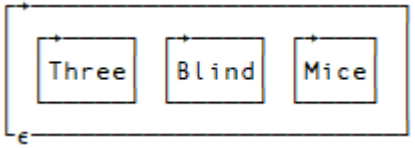
- ***Literally*** means: Statically parseable, without executing APL primitives

- Nested vectors using (...)
- High rank arrays using [...]
- Namespaces using (**name:value** pairs)

Hopefully "Intuitive" ... let's test it!

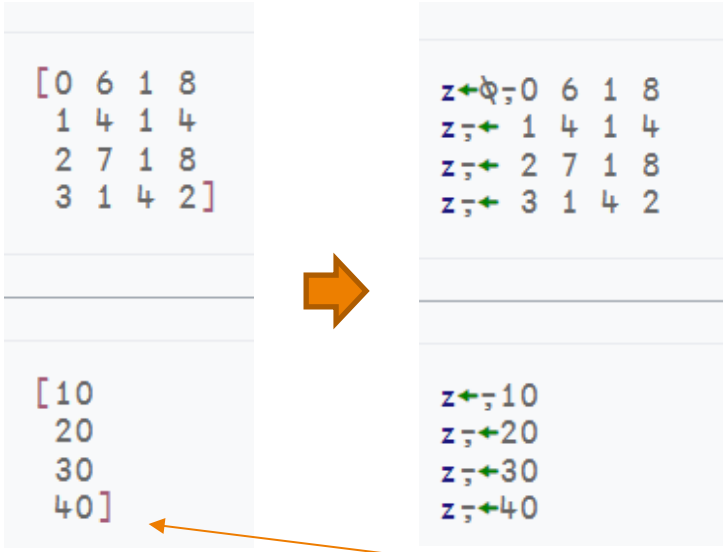
```
('Three'
 'Blind'
 'Mice')
```

```
z←,⊂'Three'
z,←⊂'Blind'
z,←⊂'Mice'
```



- Multiple statements within parentheses define a nested vector

Matrices



- Statements within brackets define a high rank array
- The results of the statements are "mixed" to produce the final array
- NB: The minimum rank of each individual result is 1 (vector)
  - The result will have 2 or more dimensions.

- The last example gives us a one-column matrix, not a vector

"Multiple Statements"

```
[1  0  1
 0  1  0
 1  0  1]
```

... is equivalent to ...

```
[1  0  1 ⋄ 0  1  0 ⋄ 1  0  1]
```

Array Notation uses brackets and parentheses...

... that span Multiple Statements.

Statements can be separated by new lines OR by diamonds

Vector of Matrices

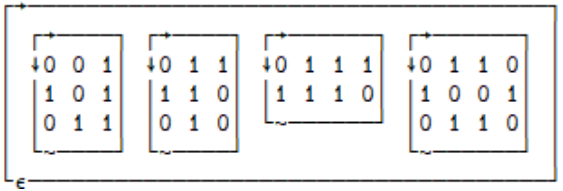
```
([0 0 1
 1 0 1
 0 1 1]

 [0 1 1
  1 1 0
  0 1 0]

 [0 1 1 1
  1 1 1 0]

 [0 1 1 0
  1 0 0 1
  0 1 1 0])
```

```
a←⍪0 0 1
a⍪← 1 0 1
a⍪← 0 1 1
z←,⊂a
a←⍪0 1 1
a⍪← 1 1 0
a⍪← 0 1 0
z,←⊂a
a←⍪0 1 1 1
a⍪← 1 1 1 0
z,←⊂a
a←⍪0 1 1 0
a⍪← 1 0 0 1
a⍪← 0 1 1 0
z,←⊂a
```



- A vector (…) of four matrices […]

Error Message Table

```
[0 'OK'
 1 'WS FULL'
 2 'SYNTAX ERROR'
 3 'INDEX ERROR'
 4 'RANK ERROR']
```

```
z←⍳⍪0 'OK'
z⍪← 1 'WS FULL'
z⍪← 2 'SYNTAX ERROR'
z⍪← 3 'INDEX ERROR'
z⍪← 4 'RANK ERROR'
```

- A two-column nested matrix

Literal Namespace Notation

```
      person←(first: 'Max' ⋄ last:'Mustermann')

      person.last,', ',person.first
Mustermann, Max
```

```
      person←(
          first: 'Max'
          last:  'Mustermann'
      )
```

- "JSON for APL"

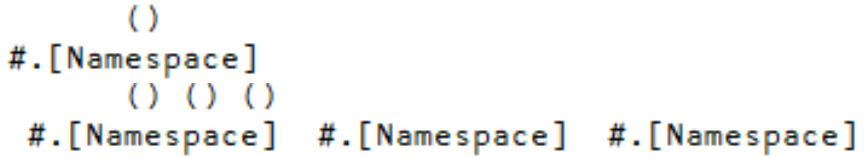
Namespace With a Matrix

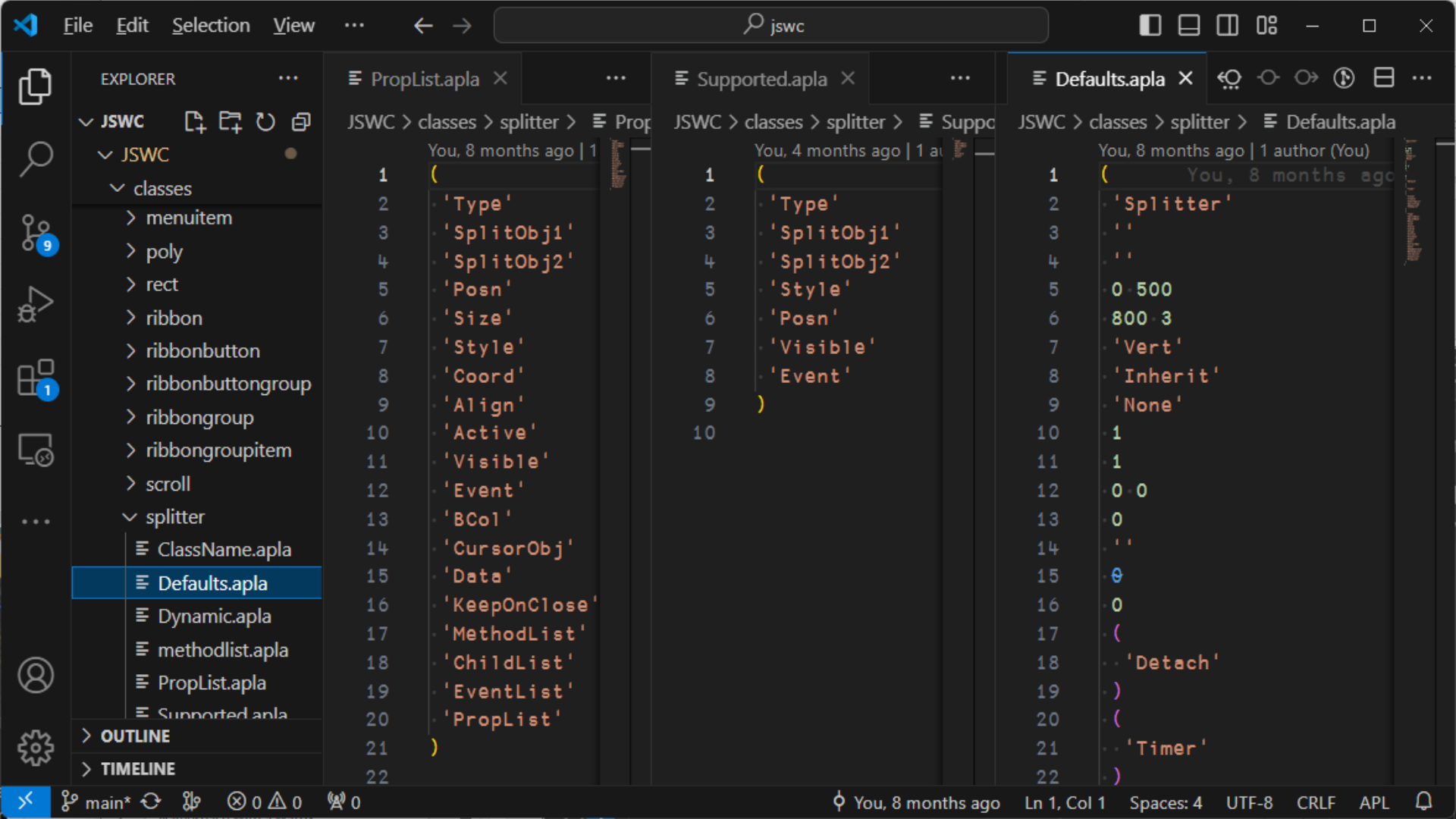
```
(y:(x:['hello'
       'world']))
```

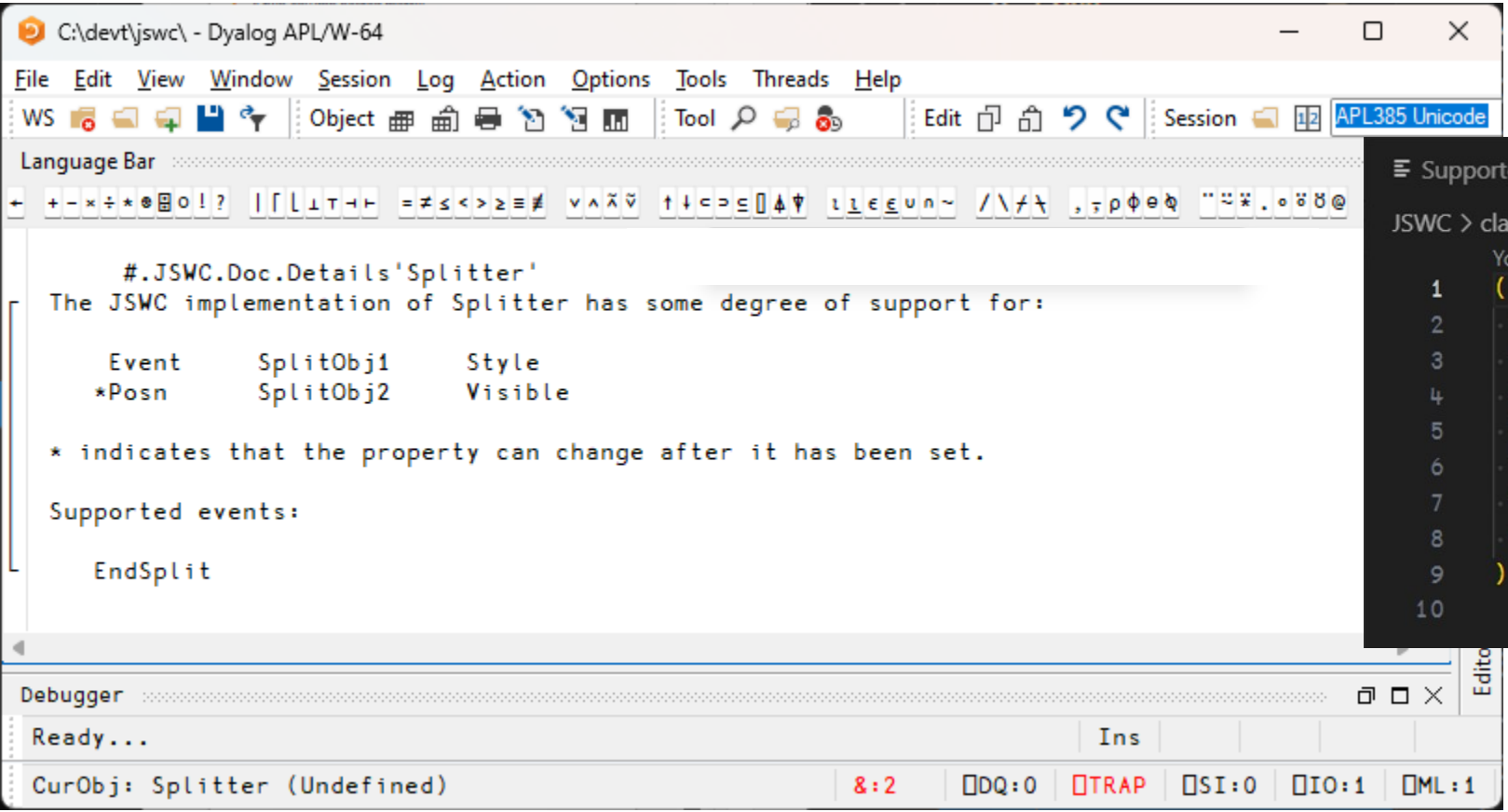
```
z←⎕NS⍬
z.y←⎕NS⍬
z.y.x←⍪'hello'
z.y.x,←'world'
```

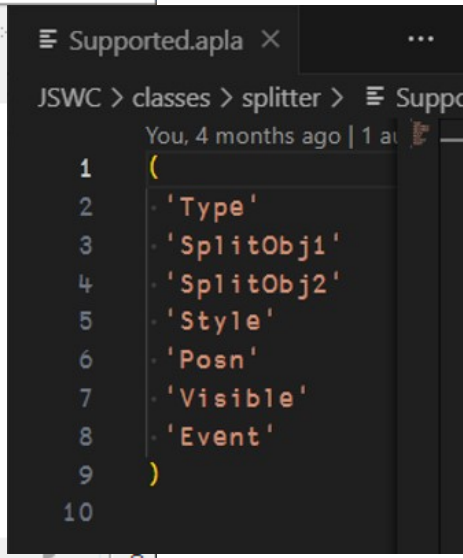
- Vector notation, Matrix notation, and Namespace notation can be contained within each other.

Empty Namespace(s)




File Edit Selection View
jswc
EXPLORER
JSWC
JSWC
classes
menuitem
poly
rect
ribbon
ribbonbutton
ribbonbuttongroup
ribbongroup
ribbongroupitem
scroll
splitter
ClassName.apla
Defaults.apla
Dynamic.apla
methodlist.apla
PropList.apla
Supported.apla
OUTLINE
TIMELINE
PropList.apla
JSWC > classes > splitter > Prop
You, 8 months ago | 1
('Type' 'SplitObj1' 'SplitObj2' 'Posn' 'Size' 'Style' 'Coord' 'Align' 'Active' 'Visible' 'Event' 'BCol' 'CursorObj' 'Data' 'KeepOnClose' 'MethodList' 'ChildList' 'EventList' 'PropList')
Supported.apla
JSWC > classes > splitter > Suppc
You, 4 months ago | 1 au
('Type' 'SplitObj1' 'SplitObj2' 'Style' 'Posn' 'Visible' 'Event')
Defaults.apla
JSWC > classes > splitter > Defaults.apla
You, 8 months ago | 1 author (You)
(You, 8 months ago
'Splitter'
''
''
0 500
800 3
'Vert'
'Inherit'
'None'
1
1
0 0
0
''
0
0
(
'Detach'
)
(
'Timer'
)
main*
0 0 0
You, 8 months ago
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
APL


C:\devt\jswc\ - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL385 Unicode
Language Bar
#.JSWC.Doc.Details'Splitter'
The JSWC implementation of Splitter has some degree of support for:
Event SplitObj1 Style
*Posn SplitObj2 Visible
* indicates that the property can change after it has been set.
Supported events:
EndSplit
Debugger
Ready...
Ins
CurObj: Splitter (Undefined)
&:2
DQ:0
TRAP
SI:0
IO:1
ML:1


Supported.apla
JSWC > classes > splitter > Suppo
You, 4 months ago | 1 au
1 (
2 'Type'
3 'SplitObj1'
4 'SplitObj2'
5 'Style'
6 'Posn'
7 'Visible'
8 'Event'
9)
10

Array Notation - Status

- APL model used by Link to store variables for a couple of years
- `⎕SE.Dyalog.Array.Serialise`
  `|Deserialise`

- Public proposal published
  - https://aplwiki.com/wiki/Array_notation
- Very little feedback from the community

- Will be integrated with interpreter in v20.0

Prototypes

V20.0 Production

Live Demos..?