DYALOG

Belfast 2018

D03: Technical Road Map: Under The Covers

Jay Foad

Let’s come back down to Earth

- Version 17.0
- Upcoming releases
- What else are we doing?

Version 17.0

Released 23 July 2018

Version 17.0

Released 23 July 2018

- Improved error diagnostics
- New and improved portable file functions
- Locals lines
-]LINK
-]HELP

Version 17.0

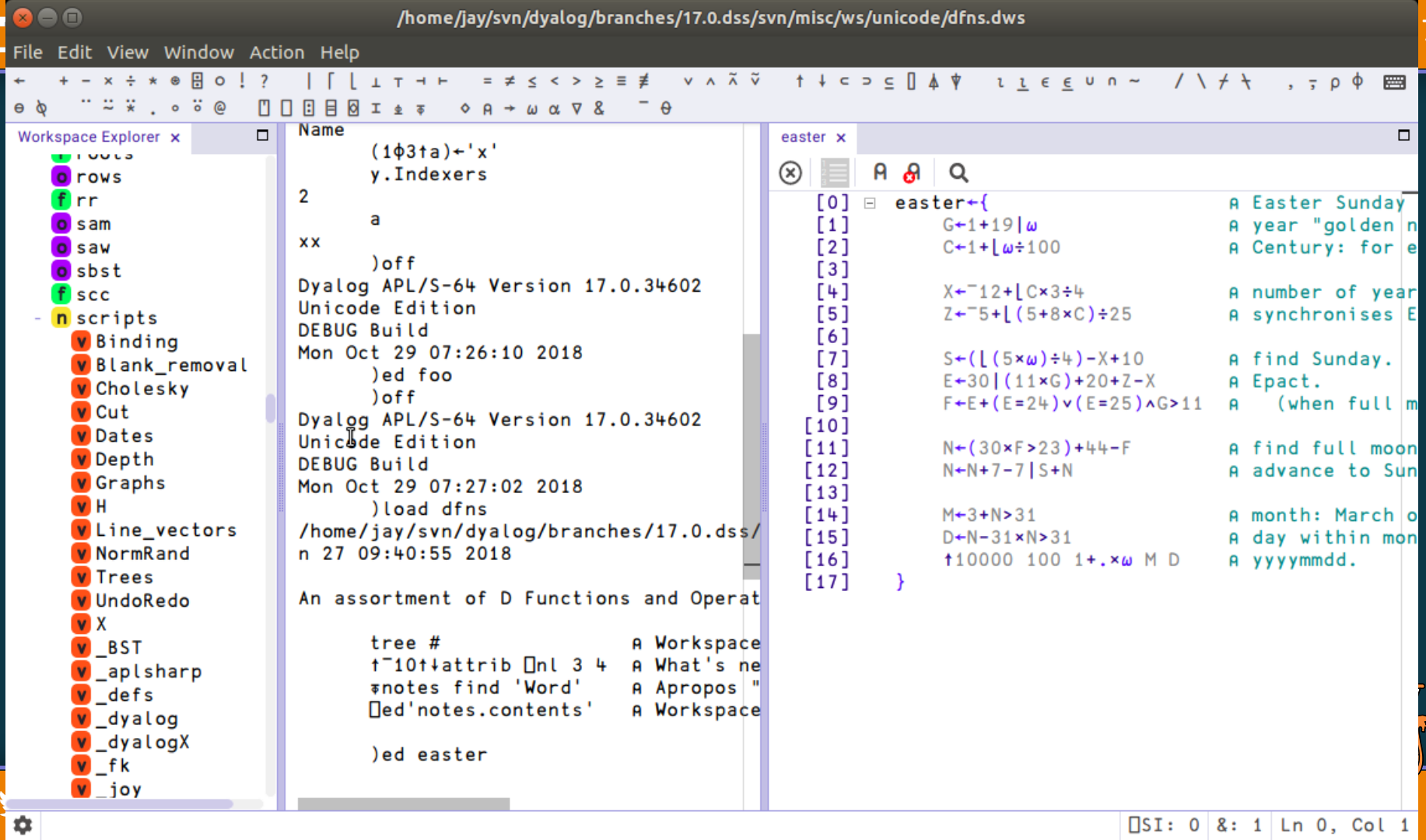
Released 23 July 2018

- HTML Renderer and data binding improvements
- APL as a shared library

Version 17.0

Released 23 July 2018

- RIDE 4.1
  - Floating windows
  - Snappier performance
  - Snappier look and feel
  - Default interface on Mac/Linux/Pi

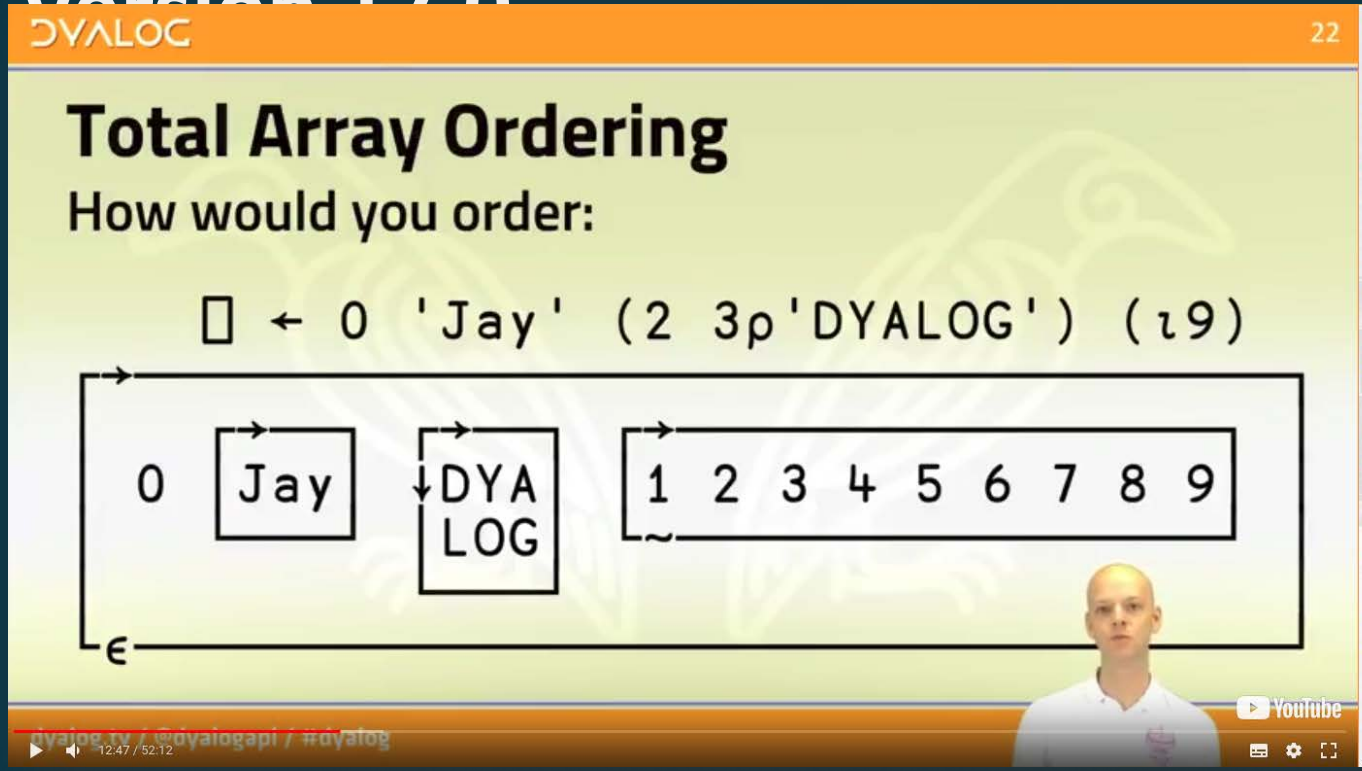

/home/jay/svn/dyalog/branches/17.0.dss/svn/misc/ws/unicode/dfns.dws
File Edit View Window Action Help
Workspace Explorer
rows
rr
sam
saw
sbst
scc
scripts
Binding
Blank_removal
Cholesky
Cut
Dates
Depth
Graphs
H
Line_vectors
NormRand
Trees
UndoRedo
X
_BST
_aplsharp
_defs
_dyalog
_dyalogX
_fk
_joy
Name
(1⌽3↑a)←'x'
y.Indexers
2
a
xx
)off
Dyalog APL/S-64 Version 17.0.34602
Unicode Edition
DEBUG Build
Mon Oct 29 07:26:10 2018
)ed foo
)off
Dyalog APL/S-64 Version 17.0.34602
Unicode Edition
DEBUG Build
Mon Oct 29 07:27:02 2018
)load dfns
/home/jay/svn/dyalog/branches/17.0.dss/
n 27 09:40:55 2018
An assortment of D Functions and Operat
tree #
⍎¯10↑⍕attrib ⎕nl 3 4
⍎notes find 'Word'
⎕ed'notes.contents'
)ed easter
easter
[0] easter←{
[1] G←1+19|⍵
[2] C←1+⌊⍵÷100
[3]
[4] X←¯12+⌊C×3÷4
[5] Z←¯5+⌊(5+8×C)÷25
[6]
[7] S←(⌊(5×⍵)÷4)-X+10
[8] E←30|(11×G)+20+Z-X
[9] F←E+(E=24)∨(E=25)∧G>11
[10]
[11] N←(30×F>23)+44-F
[12] N←N+7-7|S+N
[13]
[14] M←3+N>31
[15] D←N-31×N>31
[16] ↑10000 100 1+.×⍵ M D
[17] }
⎕SI: 0 &: 1 Ln 0, Col 1

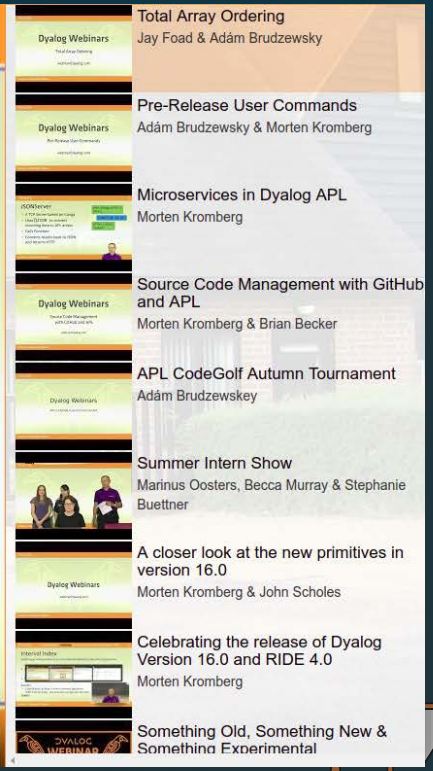
Version 17.0

Released 23 July 2018

- TAO: Total Array Ordering extends ⍋ and ⍒ to arbitrary nested arrays
- Extended Unique (∪) now works on matrices and higher ranked arrays
- Performance, performance, performance…

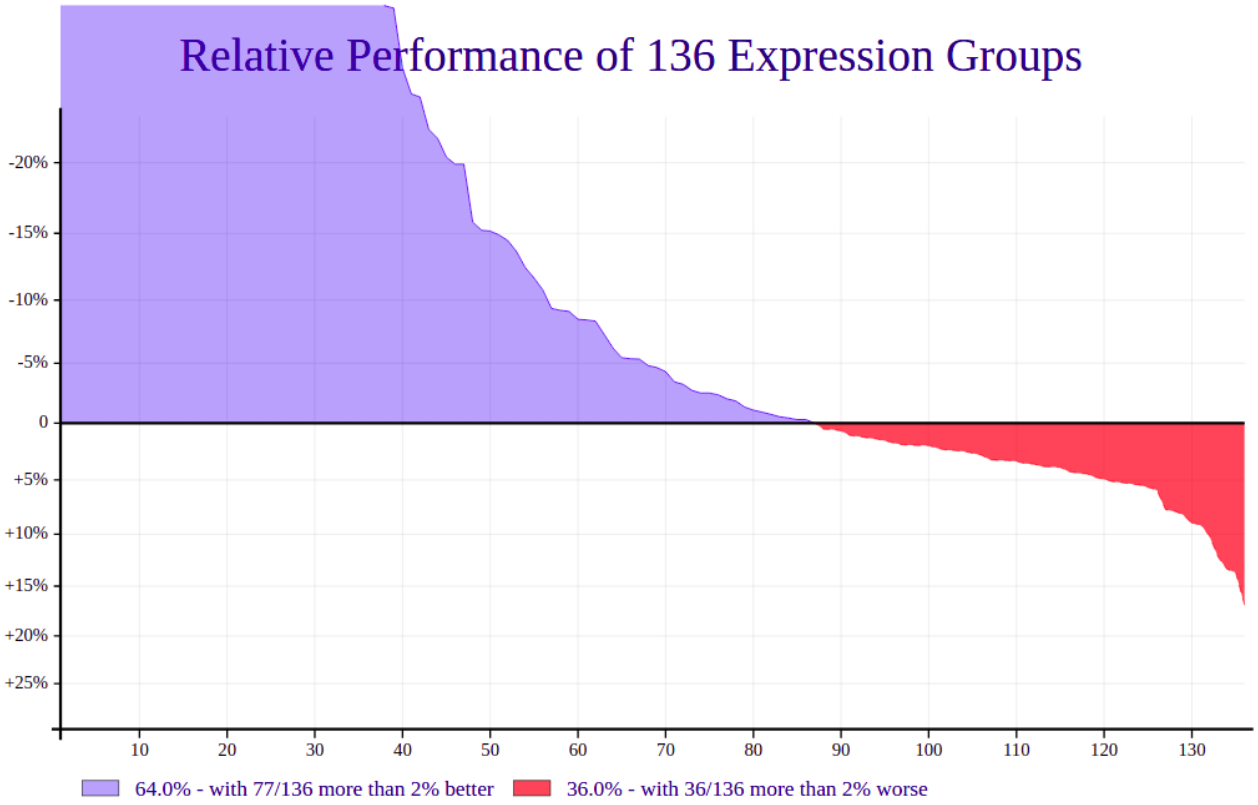
Version 17.0


DYALOG
22
Total Array Ordering
How would you order:
⍞ ← 0 'Jay' (2 3⍴'DYALOG') (⍳9)
0
Jay
DYA
LOG
1 2 3 4 5 6 7 8 9
YouTube
12:47 / 52:12


Total Array Ordering
Jay Foad & Adám Brudzewsky
Pre-Release User Commands
Adám Brudzewsky & Morten Kromberg
Microservices in Dyalog APL
Morten Kromberg
Source Code Management with GitHub and APL
Morten Kromberg & Brian Becker
APL CodeGolf Autumn Tournament
Adám Brudzewskey
Summer Intern Show
Marinus Oosters, Becca Murray & Stephanie Buettner
A closer look at the new primitives in version 16.0
Morten Kromberg & John Scholes
Celebrating the release of Dyalog Version 16.0 and RIDE 4.0
Morten Kromberg
Something Old, Something New & Something Experimental

Performance Comparison
Between Windows-64 17.0.33755.0 W Development and Windows-64 16.0.30270.0 W Development

Geometric mean of 136 expression groups: -26.0%



DYALOG The tool of thought for software solutions

Contact Us

Home | Business | Learning | Community | Resources | News | About Us

- Products
- Services
- Prices and Licences
- Support (DSS)
- Download Dyalog – Free
- Dyalog '18

Home >> Business >> Products >> Dyalog >> Dyalog Versions >> Version 17.0 >> Performance

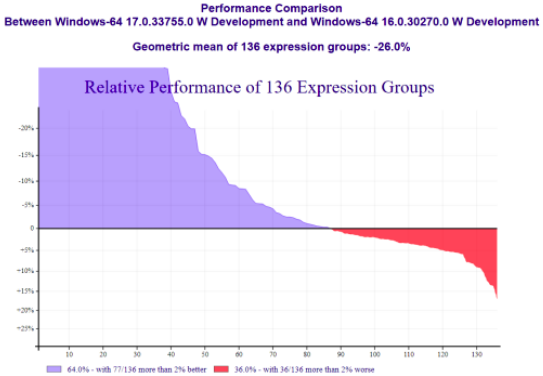
Performance

Caveat: Factors specified on this page are obtained from micro-benchmarks performed on specific primitive functions; in real applications factors will depend on a mix of primitives.

All benchmark tests were performed on 64-bit interpreters on Linux/Microsoft Windows operating systems.

Internal Benchmarks

Internal benchmarking was performed on the initial release of Dyalog version 17.0 and the results compared with the initial release of Dyalog version 16.0.



The benchmarking process comprises over 13,000 benchmarks in more than 130 groups; the group geometric mean timing ratios are measured and plotted against the groups sorted by their means. The vertical axis of the graph shows the ratios as a percentage change; negative values are shown in blue and indicate a performance enhancement, and positive values are shown in red and indicate a deterioration in performance.

Results showed that core interpreter performance in Dyalog version 17.0 has an average improvement of 26% over Dyalog version 16.0.

Areas of Focus

For previous releases we have presented the ways in which Dyalog has been sped up in tabular form. We tried to make such

Version 17.0

Released 23 July 2018

- TP1: Dyalog Version 17.0 In Depth
  Jay Foad, Richard Smith and Adám Brudzewsky
  Thursday 13:45

Upcoming releases

Upcoming releases

17.1	18.0
Short cycle	Long cycle
Early 2019 release	Mid 2020 release
Tying up loose ends	Major new projects
Developed concurrently	

Version 17.1 (2019 release)

- HTML renderer on all desktop platforms
- Better support for headless (Linux) images: run under Docker, debug with RIDE

Version 17.1 (2019 release)

- HTML renderer on all desktop platforms
- Better support for headless (Linux) images: run under Docker, debug with RIDE
- Packaging and signing
- Ongoing performance work
- (And routine maintenance and bug fixes as usual)

17.1: Docker

- Windows Server Core: 6 GB
- Nano Server: 435 MB

- Dyalog install with CEF: 200 MB
- Dyalog install without CEF: 48 MB

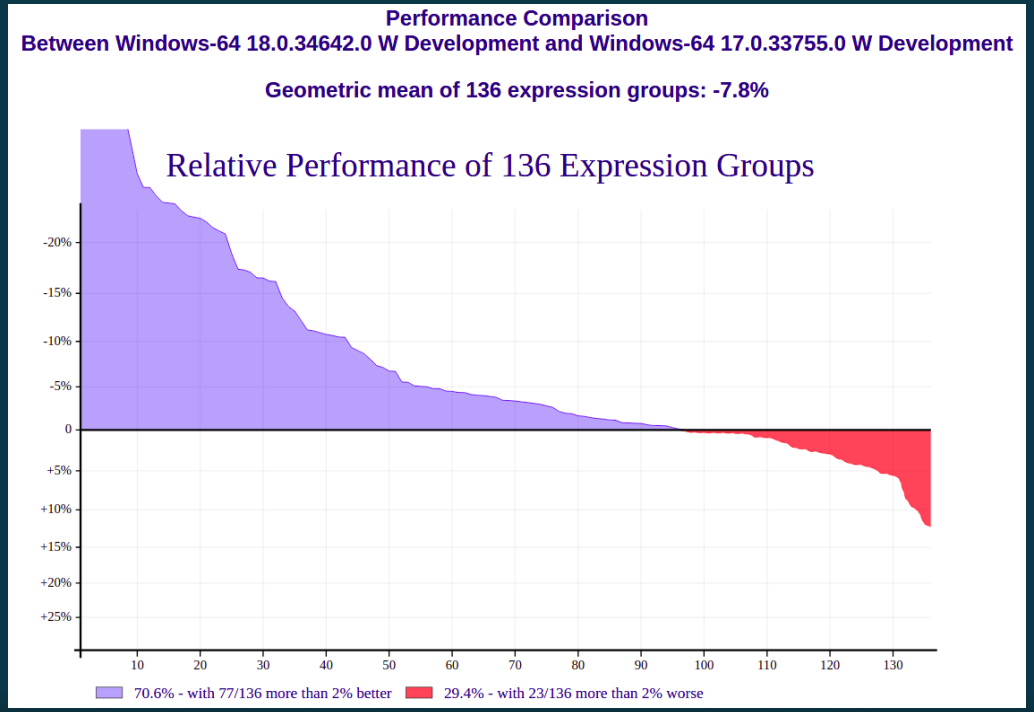
- Ubuntu Linux: 29 MB
- Alpine Linux: 2 MB

17.1: HTML renderer

Announcement!

17.0 runs on Debian 7 or Centos 6
CEF requires Debian 8 or Centos 7

Performance (17.1 and 18.0)



Version 18.0 (2020 release)

Version 18.0 (2020 release)

These are plans, not promises!

18.0: Theory of Everything

- An internal object model for Dyalog APL

➢ D09: JD's Big Toe
John Daintree
Tuesday 11:20

18.0: Increased maximum array rank

… and lift a few other limitations:

- 4k lines in a functions
- 4k names/constants in a function
- etc.

18.0: Increased maximum array rank

Announcement!

We want to stop supporting ⎕LOAD of a workspace
that was ⎕SAVEd with an SI stack
by an older version of the interpreter.

18.0 .NET Core bridge

- Open source and cross-platform
- Required for headless Windows (Nano Server)
- Provides new functionality on Linux and Mac

18.0: Array notation

- Generalises strand notation to matrices and higher rank arrays

➢ D04: Array Notation Mk III
Adám Brudzewsky
Monday 11:00

18.0: Magic arrays

- You (the wizard) invent a new representation:
  - Sparse
  - Inverted
  - Etc
- Wave your wand…
- … and it appears as a normal APL array
- (No change to the rest of your application code)

18.0: Magic arrays

```
:Magic Inverted
    :Field private vec ⍝ vector of column vectors
    ∇ r←ShapeOf y
      :Implements ⍴⍵
      r←(⍴⊃vec),(⍴vec)
    ∇
    ∇ r←x Take y
      :Implements ⍺↑⍵
      ...
    ∇
    ...
:EndMagic
```

18.0: New operators

Over	`α f⍥g ω`	`↔`	`(g α) f (g ω)`
Under	`α f⍢g ω`	`↔`	`(g⍣¯1) (g α) f (g ω)`
Obverse	`α f⍨g ω`	`↔`	`α f ω`
	`f⍨g⍣¯1`	`↔`	`g⍨f`

18.0: Cross platform config

Replaces registry, command line, environment in a way that is

- cross platform
- easy to change per-application
- easy to share between interpreter versions

18.0: Executable scripts

Storing source code in text files is good
Running code from text files is better

We need better support for:

- Running APL batch jobs
- Loading code under program control
- Managing dependencies between scripts

Version 18.0 (2020 release)

- Theory of Everything
- Increased maximum array rank
- .NET Core bridge
- Array notation
- Magic arrays
- New operators
- Cross platform config
- Executable scripts

What else are we doing?

What else are we doing?

- RIDE 4.2…

What else are we doing?

- RIDE 4.2…
- … and VS Code integration

What else are we doing?

- RIDE 4.2…
- … and VS Code integration

➢ D05: RIDE 4.1 and Next Generation Integrations
Gilgamesh Athoraya
Monday 13:30

What else are we doing?

- Package management

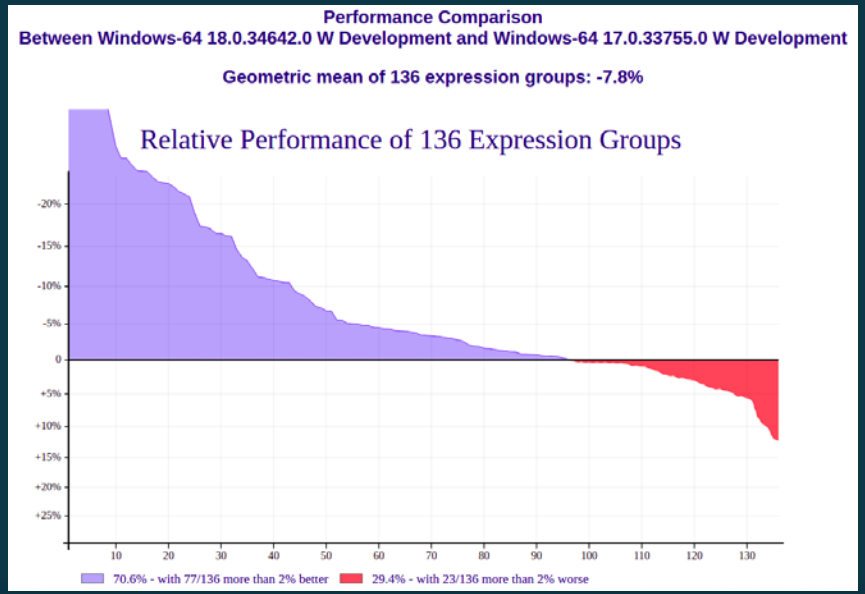
➢ U05: The APL Package Manager
Gilgamesh Athoraya
Tuesday 09:00

What else are we doing?

- Performance, performance, performance

What else are we doing?

- Performance, performance, performance



What else are we doing?

- Performance, performance, performance

- D08: Sub-nanosecond Searches Using Vector Instructions
  Marshall Lochbaum
  Monday 16:45

What else are we doing?

- Performance, performance, performance

- D08: Sub-nanosecond Searches Using Vector Instructions
- D14: Inverted Tables
  Roger Hui
  Thursday 11:00

What else are we doing?

- Performance, performance, performance

- D08: Sub-nanosecond Searches Using Vector Instructions
- D14: Inverted Tables
- D15: The Interpretive Advantage
  Marshall Lochbaum
  Thursday 11:30

What else are we doing?

- Co-dfns

- U04: Co-dfns 2018 – What's New?
  Aaron Hsu
  Monday 17:30

What else are we doing?

- Co-dfns

- U04: Co-dfns 2018 – What's New?
- U18: Introducing the Mystika Project
  Erik Wallace
  Thursday 10:00

What else are we doing?

- D10: Dfns – Past, Present and Future
  John Scholes
  Wednesday 11:00

In summary…

- 17.1 due out early(ish) in 2019
- 18.0 due out mid 2020