

APL Germany / GSE Autumn Conference 2019

# News from Dyalog

*Richard Smith*

# Where are Morten and Gitte?

- Morten and Gitte had to cancel at the last minute for family reasons
- RichardP and I managed to get one of the last available seats from London to Frankfurt
- ... but are somewhat unprepared, so please bear with us!

# Staff news

# Arrivals

- ◆ Richard Park joined late last year
- ◆ Works on documentation, media, training, webinars
- ◆ Based in Bramley, UK



# Arrivals

- ◆ Nathan Rogers is an APL programmer and part of the new United States Consulting Team
- ◆ Got to know him via the APL Orchard\*
- ◆ Joined Dyalog in April
- ◆ Based in Denver, Colorado

\* More about that later



# Arrivals

- ◆ Josh David is also an APL programmer and part of the United States Consulting Team
- ◆ Was winner of the APL Problem Solving Competition in 2016
- ◆ Joined Dyalog in June
- ◆ Based in Clinton, New Jersey



# Arrivals

- ◆ Matt Weise is expected to be joining the Bramley development team in January
- ◆ C developer working on the interpreter
- ◆ Originally from Canada; relocating from Cambridge, Massachusetts
- ◆ He can speak German!



# Arrivals

🟡 TBA





# APL evangelism

# Annual user meeting

- ◆ Dyalog '19 was held in Denmark in September
- ◆ 79 participants from 13 countries
- ◆ Videos coming online: <http://dyalog.tv>
- ◆ Planning for Dyalog '20 is under way!

## Talks outside the APL Community

- ◆ Marshall presented at LambdaConf (functional programming) in Boulder, Colorado in June
- ◆ Michael ran an APL workshop in a school here in Germany in October
- ◆ Aaron and Morten will present at FunctionalConf in Bangalore, India next week

# Webinars

- Live presentations third Thursday of the month
- Videos online: <http://dyalog.tv>



Next Dyalog Webinar

# Train Spotting in Dyalog APL

Presented by Richard Park  
Thursday 23 January at 16:00 UTC

<https://dyalog.tv>

# APL Problem Solving Competition

- ◆ Programming contest aimed at students
- ◆ Now in its 11<sup>th</sup> year
- ◆ 2019:
  - ◆ 858 registrations
  - ◆ 68 submissions with at least one correct answer in Phase I
  - ◆ 25 Phase II submissions

“ The problems were nice, splitting the challenges in sets & difficulties allowed me to skip stuff I don't like.

I am 17 years old and have been interested in programming for a long time. A couple years ago I found out about the Code Golf community in StackExchange, where, after a while, I was introduced to APL, and was immediately interested, because `1 2 3+10` worked.. Little by little, I learned the language, and now it's often my everyday calculator.

”

# Online forums and tutorials

- ◆ APL Forum
- ◆ The APL Orchard
- ◆ APLCart
  - ◆ Video online!
- ◆ APL Wiki
- ◆ For *all* APL dialects



# Blog

- In 2019:
  - Sorting
  - Community sponsorship
  - Speed vs Accuracy
  - News from Dyalog
  - Tolerated comparisons
  - RichardP, Josh and Nathan introductions
  - User meeting, and user meeting videos

# Removing barriers

# Online interpreters

- [TryAPL](#)
- [Try It Online](#)

# Open source content

- <https://github.com/Dyalog>

# Docker

- 🟡 Demo
- 🟡 No barriers for non-commercial use

<https://hub.docker.com/r/dyalog/dyalog>

# Licensing

- ◆ As of now, Dyalog is available free of charge, without registration, for non-commercial use
- ◆ Dyalog is free for non-commercial use but is not free software. A non-commercial licence can be used for experiments and proof of concept until the point in time that it is of value.

# Download Dyalog for free

- ◆ <https://www.dyalog.com/download-zone.htm>
- ◆ The interpreter is not open source

**Want to know more?**



## More info at [dyalog.com](https://dyalog.com) and [dyalog.tv](https://dyalog.tv)

- ◆ Licensing: D01 and D02 at Dyalog 2019
- ◆ Evangelism, new staff: ditto
- ◆ New staff: Dyalog blog
- ◆ APL Cart: U15 presentation at Dyalog '19
- ◆ Docker: D11 at Dyalog '18

# Coming up

## 18.0 (2020 release) preview

- ◆ New system functions for case folding and date/time handling
- ◆ New primitives: atop, over, constant, unique mask

## □C – case conversion (18.0 feature in development)

Case mapping – for display purposes

```
      1 □C 'Hello'
HELLO  -1 □C 'Hello'
hello  -1 □C 'Hello' # 123 'There'
hello # 123 there
      1 □C 'Seit dem Mittelalter gehört Frankfurt am Main...'
SEIT DEM MITTELALTER GEHÖRT FRANKFURT AM MAIN...
      A
```

Case mapping – used (erroneously) for caseless comparison

```
      'hello' ≡ 'HELLO'
0
      -1 □C 'hello' 'HELLO'
hello hello
      CI←{αα/-1 □Cα ω}
      'hello' ≡CI 'HELLO'
1
      'goodbye' ≡CI 'HELLO'
0
      'hello' # 123 ≡CI 'HELLO' # 123
1
```

Case folding – for caseless comparison

```
      A
      A σςΣ
      A
      1 □C 'σς'
ΣΣ
      -1 □C 'ΣΣ'
σσ
      'σς' ≡CI 'ΣΣ'
0
      -1 □C 'σς' 'ΣΣ'
σς σσ
      □C 'ΣΣ'
σσ
      □C 'σς'
σσ
      CI←{αα/□Cα ω}
      'σς' ≡CI 'ΣΣ'
1
      A
```

Note: no folding where there would be changes in text length

```
'strasse' ≡CI 'STRABE'  
0  
  □C 'STRABE' 'strasse'  
straÙe strasse  
  ↑'STRABE' 'STREET'  
STRABE  
STREET  
  □C ↑'STRABE' 'STREET'  
straÙe  
street
```

## DT – datetime conversion and formatting (18.0 feature in development)

Conversion of 7-element to timestamps to scalar numerics (to allow computations and reduce space requirements)

```

      A Is it Christmas yet?
      A
      12 25 2 11 47 46 527
0
      Q←now←TS
2019 11 6 11 47 46 527
      xmas←2019 12 25 0 0 0 0
      xmas-now
0 1 19 -11 -47 -46 -527
      A
      +2 NQ #'DateToIDN' now
43774
      dn←{+2 NQ #'DateToIDN' ω}
      dn`xmas now
43823 43774
      -/dn`xmas now
49
      (24 60 60 1000 13 now)÷86400000
0.4915107292
      dn←{(2NQ#'DateToIDN' ω)+(24 60 60 1000 13 ω)÷86400000}
      dn`xmas now
43823 43774.49151
      -/dn`xmas now
48.50848927
      A
      2000 0 0 -1 -1 -1 -1 +? 100 12 28 24 60 60 1000
2002 12 10 8 47 37 780
      d1←{2000 0 0 -1 -1 -1 -1 +? 100 12 28 24 60 60 1000}`i1000
      ]disp 2↑d1
2068 4 5 11 20 40 724 | 2075 8 15 6 56 21 921
      QSIZE 'd1'
56032
      d2←dn`d1
      2↑d2
61457.47269 64145.28914
      QSIZE 'd2'
8032
      (QSIZE 'd2')÷QSIZE 'd1'
0.1433466591
```

## Proposed new system function to do the conversions

```
1 ^1[]DT xmas now
43823 43774.49151
1[]DT xmas now
43823 43774.49151
- /1[]DT xmas now
48.50848927
A
1[]DT<now
43774.49151
^1[]DT 1[]DT<now
2019 11 6 11 47 46 527
^1[]DT 250 + 1[]DT<now
2020 7 13 11 47 46 527
A
^1 []DT 0
1899 12 31 0 0 0 0
^1 []DT 1
1900 1 1 0 0 0 0
^1 []DT ^0.5
1899 12 30 12 0 0 0
^1 []DT ^100000
1626 3 17 0 0 0 0
```

## Multiple supported scalar representations

```
t^'cf.dcf' []FCREATE 0
'Hello' []FAPPEND t
[]FREAD t 1
Hello
[]FRDCI t 1
136 0 9.438245348E10
[]^cfts^3^[]FRDCI t 1
9.438245348E10
^1 2[]DT cfts
2019 11 6 11 48 11 316
1 2[]DT cfts
43774.4918
A
A 3: UNIX time
A 4-9: Julian time variants
4 []DT <now
2458793.992
A 10: Microsoft Win32 FILETIME
A 11: Microsoft CLR DateTime.ticks
10 []DT <now
1.321751447E17
A 12-13: Excel
12 []DT<1900 2 28
59
12 []DT<1900 3 1
61
```

```

12 [DT<1900 2 29
DOMAIN ERROR: Invalid timestamp
12 [DT<1900 2 29
^
a 14: Microsoft OLE Automation Date

```

### Current time in various timezones

```

-1 [DT 'J' a local time
2019 11 6 11 48 28 666
-1 [DT 'Z' a UTC time (the same in this instance)
2019 11 6 11 48 29 596
1 [DT 'JZ'
43774.49202 43774.49202
24x-/1 [DT 'JZ'
0
-1 [DT 'B' a UTC+2
2019 11 6 13 48 35 847
[←h←1 [DT 'B'
43774.57544

```

### Formatting

```

dtf←1200I
'DD/MM/YYYY hh:mm:ss' dtf h
06/11/2019 13:48:38
'D/M/YY hh:mm:ss' dtf h
6/11/19 13:48:38
'_D/_M/YYYY hh:mm:ss' dtf h
6/11/2019 13:48:38
'D MMM YYYY hh:mm:ss' dtf h
6 NOV 2019 13:48:38
'D MMMM YYYY hh:mm:ss' dtf h
6 NOVEMBER 2019 13:48:38
'D Mmmm YYYY hh:mm:ss' dtf h
6 November 2019 13:48:38
'M _M MM Mmm Mmmm' dtf h
11 11 11 Nov November
'WWW-"W"ww-d' dtf h
2019-W45-3
'WWW-"W"ww-d' dtf 1 [DT <2019 12 31
2020-W01-2
'd Ddd' dtf h
3 Wed
'Dddd D Mmmm YYYY hh:mm:ss' dtf h
Wednesday 6 November 2019 13:48:38
'Dddd Doo Mmmm YYYY hh:mm:ss' dtf h
Wednesday 6th November 2019 13:48:38
'Dddd Mmmm D YYYY t:mm:sspp' dtf h
Wednesday November 6 2019 1:48:38pm
'__fi__Dddd Doo mmmm YYYY hh:mm:ss' dtf h
Keskiviikko 6. marraskuu 2019 13:48:38
dtf←1200I[Language' 'fi'

```



```

'Dddd Doo mmmm YYYY hh:mm:ss' dtf h
Keskiiviikko 6. marraskuu 2019 13:48:38
'__en__Dddd Doo mmmm YYYY hh:mm:ss' dtf h
Wednesday 6th november 2019 13:48:38
'__de__Dddd Doo Mmmm YYYY hh:mm:ss' dtf h
Mittwoch 6. November 2019 13:48:38
'__ru__Dddd Doo mmmm YYYY hh:mm:ss' dtf h
Среда 6 ноябрь 2019 13:48:38
'__cy__Dddd Doo mmmm YYYY hh:mm:ss' dtf h
DOMAIN ERROR: Unknown language 'cy'
'__cy__Dddd Doo mmmm YYYY hh:mm:ss' dtf h
^
↑=NGET 'Dict.json' 1
{
  "cy": {
    "MonthNames": [
      "Ionawr", "Chwefror", "Mawrth", "Ebrill", "Mai", "Mehefin",
      "Gorffennaf", "Awst", "Medi", "Hydref", "Tachwedd", "Rhagfyr"
    ],
    "ShortMonthNames": [
      "Ion", "Chw", "Maw", "Ebr", "Mai", "Meh",
      "Gor", "Awst", "Medi", "Hyd", "Tach", "Rhag"
    ],
    "WeekdayNames": [
      "Dydd Sul", "Dydd Llun", "Dydd Mawrth", "Dydd Mercher",
      "Dydd Iau", "Dydd Gwener", "Dydd Sadwrn"
    ],
    "ShortWeekdayNames": [
      "Sul", "Llun", "Maw", "Mer", "Iau", "Gwen", "Sad"
    ],
    "MorningAfternoon": [
      "yb", "yh"
    ],
    "Ordinals": [
      "af", "il", "ydd", "ydd", "ed", "ed", "fed", "fed", "fed",
      "fed", "eg",
      "fed", "eg", "eg", "fed", "eg", "eg", "fed", "eg", "fed",
      "ain", "ain",
      "ain", "ain", "ain", "ain", "ain", "ain", "ain", "ain", "ain"
    ]
  }
}

dict←JSON=NGET'Dict.json'
dict.cy.ShortWeekdayNames
Sul Llun Maw Mer Iau Gwen Sad
dtf←1200I('Dictionary'dict)('Language' 'cy')
'Dddd Doo mmmm YYYY hh:mm:ss' dtf h
Dydd Mercher 6ed tachwedd 2019 13:48:38

```

Using .NET to parse a date and bring it into the workspace as a timestamp

```

    presents<'mittwoch 25 dezember 2019 07:00:00'
    USING<'System'
'System.Globalizati0n,System.Globalizati0n.dll' 'Dyalog'
    lang<CultureInfo.CreateSpecificCulture<'de-DE'
    format<'dddd dd MMMM yyyy HH:mm:ss'
    ret<NEW ByRef
    DateTime.TryParseExact presents format lang
DateTimeStyles.None ret
1
    ret.Value.ToOADate
43824.29167
    -1 14 DT ret.Value.ToOADate
2019 12 25 7 0 0 0
R
R Happy Christmas!
```