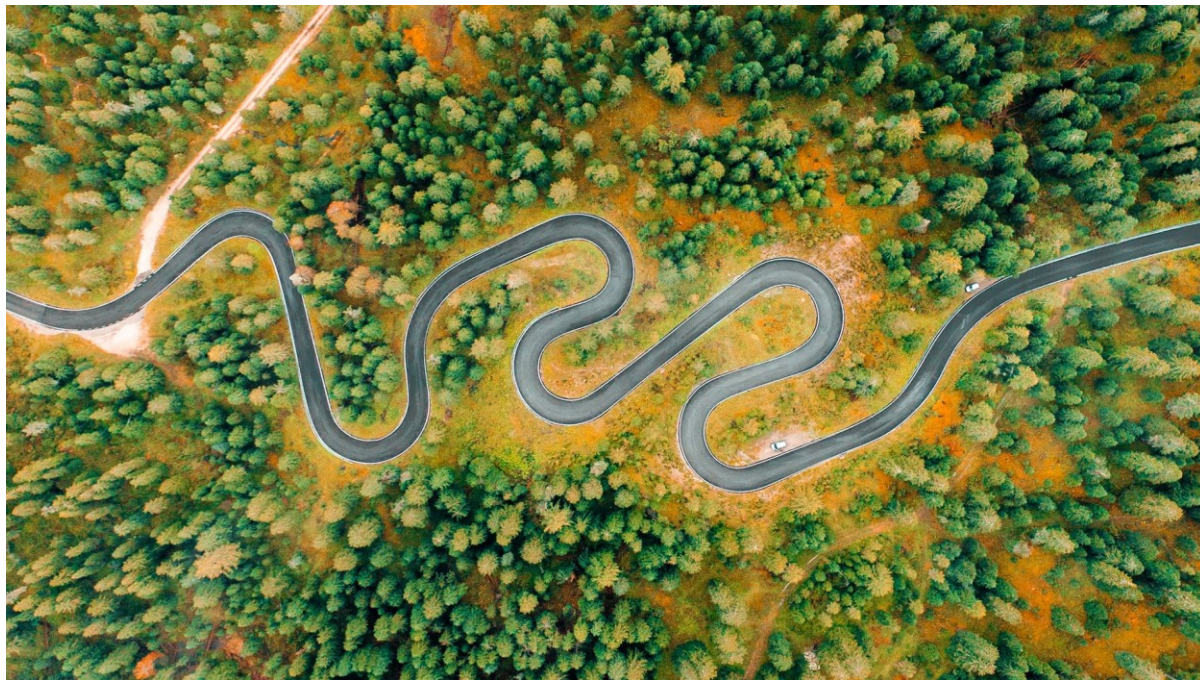


DYNA Fall 2025
A **DYALOG** USER MEETING

29-30 September

Dyalog Road Map

Morten Kromberg
CTO, Dyalog
(20 years 5 months)

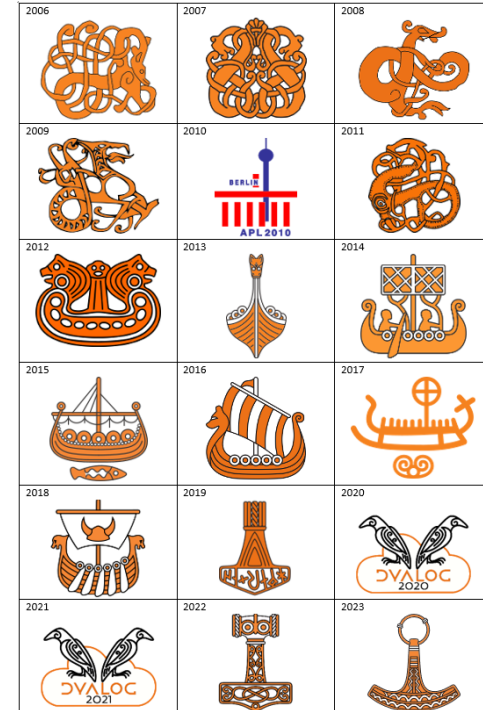


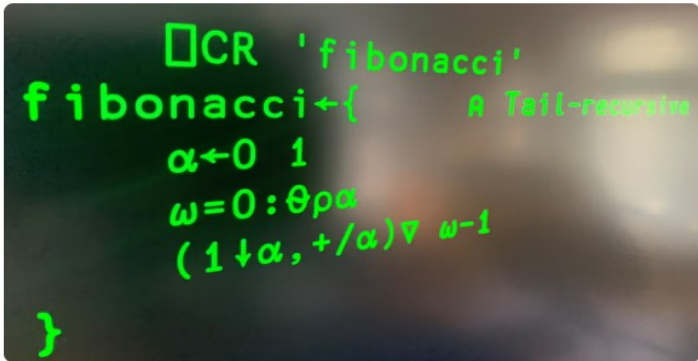
Welcome to DYNA Fall 2025

A DIALOG USER MEETING

Current Event Schedule

- Global Dyalog User Meeting in even years
 - 2026: Eastbourne, UK (return to Dyalog'14 venue)
- DYNA Fall in odd years (~September, 2 days)
- DYNA Spring every year (April, 1 day)
 - Looking at Monday 4/13/26
- What about the "Dyalog NYC Meetup"?





New York Dyalog APL Meetup

5.0 10 ratings

New York, NY, USA

39 members · Public group

Organized by Morten Kromberg and 2 others

Share: n f t in e

Request to join

About Events Members Photos Discussions

What we're about

We are a group for anyone interested in Dyalog APL.

Upcoming events (1)

See all

MON, SEP 29, 2025, 9:30 AM EDT

DYNA Fall 2025

Needs location



Join us in New York City for DYNA Fall 2025, a two-day in-person event aimed at bringing the North American APL community together through presentations, workshops, and hands-on learning....

3 attendees

Attend

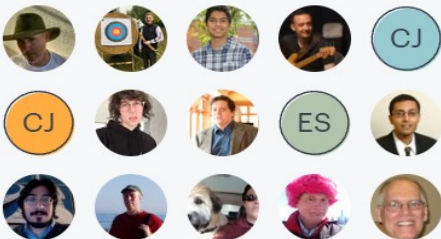
Organizers

Morten Kromberg and 2 others

Message

Members (39)

See all



Start	Mins	Title	Presenter	
9:30 AM	45	The Dyalog Road Map	Morten Kromberg	
10:15 AM	45	Dyalog and AI	Stefan Kruger	
11:00 AM	15	Break		
11:15 AM	30	JAWS – Jarvis And Web Sockets	Brian Becker	
11:45 AM	30	An Interface to Kafka	Martina Crippa	
12:15 PM	30	Static Analysis of APL	Aaron Hsu and Brandon Wilson	
12:45 PM	45	Lunch		
1:30 PM	30	Converting from APL+Win to Dyalog APL	Alex Holtzapple	
2:00 PM	60	Dyalog APL: Our (Not So) Secret Ingredient	Mark Wolfson	
3:00 PM	30	Break		
3:30 PM	30	The Data Science Journey	Josh David	
4:00 PM	30	What Can Vectorised Trees Do For You?	Asher Harvey-Smith	
4:30 PM	30	ArrayLab: Building a 3D APL Game	Holden Hoover	
5:00 PM	30	The APL Trust US	Diane Hymas	





The Plan is Unchanged



Continue to grow
the revenue base

Customers use of APL is growing
Migrants keep arriving



Grow and replenish
the team

Hire new people in good time
before old-timers retire



Hunt the new
generation of users

Get back into data science &
education
Continue to position APL in the
functional community



Business as Usual



Thanks, Mike!

Migration Tools

- **deltaWI:** Migrate **❏WI** (APL+Win / MicroAPL) GUI to **❏WC** (Dyalog APL)



HEAP LEACH EXAMPLE COPPER HEAP LEACH

Stream Number

ORE TYPES
PIX

Recycle Streams

Stream	Iter	Flow	Comp	Change
12	0	1	0	0

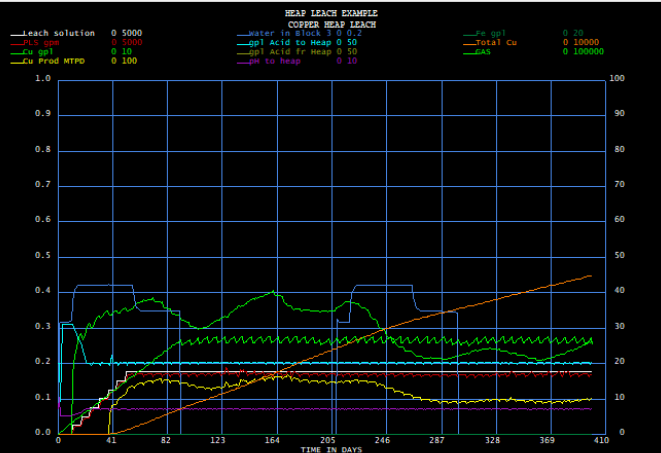
Run Parameter

Start	STOP	Low	TOL	High	TI	CTI	Time
1	9	9	-3	-3	-3	400	400 13.985

Process Values

INSTRU	LABEL	VALUE
1010	Leach solu	887.61892
1011	PLS gsm	860.80315...
1012	Cu gpl	2.6392104...
1013	Cu Prod M...	10.362450...
1015	Water in BL	0
1016	gpl Acid to...	10.080576...
1017	gpl Acid fr...	0
1018	pH to heap	0.7099708...
1019	Fe gpl	0
1020	Total Cu	0
1021	GAS	25440

RUNNING



Alarms

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56
57	58	59	60	61	62	63
64	65	66	67	68	69	70
71	72	73	74	75	76	77
78	79	80	81	82	83	84
85	86	87	88	89	90	91
92	93	94	95	96	97	98
99	100	101	102	103	104	105
106	107	108	109	110	111	112
113	114	115	116	117	118	119
120	121	122	123	124	125	126
127	128	129	130	131	132	133
134	135	136	137	138	139	140

Copper Heap Leach

Migration Tools

- **deltaWI:** Migrate $\square$ WI (APL+Win / MicroAPL) GUI to $\square$ WC (Dyalog APL)
- **EWC:** Migrate $\square$ WC (Windows) to Linux, macOS, and the Web



FILE EDIT HOME DATA DEVELOPER HELP

Access Locks Macros Dashboard Utilities Internet Security Corporate Action Black and Scholes VAR analysis New Securities Check RII Prime Segment Overall Data Quality Local Data Quality Data Quality

Shares

- Portfolio
- Market
 - Specifications
 - Prices
 - Fair prices
 - Exchange Rates
 - Indices
 - Unified Rating
 - Federal Funds Rate
 - Synthetic Indices
 - Echelons
 - VAR (deltaN+EWMA)
 - Services
- Bonds
- Futures
- Foreign Exchange
- Derivatives
- Totals
- Risk
- Special reports
- Accounting
- System

List of Shares Share Options

	Security Name (long)	Security Code (ISIN)	Issue Date	F L G	Security Code (Reuters)	Curr.	Asset Type (CB RF)
39	FinEx RusRTSEquityUSD	IE00BQ1Y6480	24/02/16			USD	SHS7
40	FinEx US REIT UCT CI USD	IE0000CHPRB9	16/11/21			USD	SHS7
41	FinEx US TIPS UCT CI RUB	IE00096YBNR3	09/06/21			RUR	SHS7

Dividend Plan

Description/1	Description/2	Description/3	Classifications	Dividend Plan	Extra Parameters	Historical Information
---------------	---------------	---------------	-----------------	---------------	------------------	------------------------

FILE EDIT HOME DATA DEVELOPER HELP

Access Locks Macros Dashboard Utilities Internet Security Corporate Action Black and Scholes VAR analysis Overall Data Quality Local Data Quality Data Quality

	Date of Dividend Payment	Date of Calc.	C N D	Gross Dividend per Share (DC)	Exchange Rate (FC/LC)	Gross Dividend per Share (LC)	W.Tax on Single Div. (%)	Net Dividend per Share (LC)	Gross Dividend per Share USD	Exchange Rate (DC/FC)
1	00/00/00	00/00/00			na					

Black & Scholes - Share Option

Value Date: 00/00/00 Exercise Date: 00/00/00

Spot Price: Strike Price:

Volatility: Risk Free Interest Rate:

Call: Put:

Delta Call: Delta Put:

Theta Call: Theta Put:

Vega: Gama:

Duration: 0

Calculation Cancel Tools... Help

FILE EDIT HOME DATA DEVELOPER HELP

Access Macros Dashboard Internet Security Corporate Action Black and Scholes

Locks Macros Dashboard Utilities Application

List of Shares Share Options

	Security Name (long)	Security Code (ISIN)
39	FinEx RusRTSEquityUSD	IE00BQ1Y6480
40	FinEx US REIT UCT CI USD	IE0000CHPRB9
41	FinEx US TIPS UCT CI RUB	IE00096YBNR3
42	FinEx US TIP	
43	FinEx USA IT	
44	FinEx USA IT	
45	FinEx USA U	
46	FinEx USD C	
47	Fix Price Gro	
48	GC Bonds K	
49	GC Expanded	
50	GC Shares K	
51	Globaltrans	
52	HeadHunter	
53	HMS Group	
54	ITI Fds Rus-F	
55	ITI Funds Ru	
56	McDonalds-	
57	MD Medical	
58	Meta Platfor	
59	Microsoft-ac	
60	Nike-ao	
61	NVIDIA-ao	
62	Okey Group	
63	Polymetal-ac	
64	QIWI, depoz	
65	Ros Agro GD	
66	Softline Hol	
67	TCS GroupH	
68	Tesla-ao	
69	United Medi	
70	VEON-ao	
71	VK Company	
72	X5 Retail Gro	

Dividend Plan

Description/1	Description/2	Description/3	Classifications	Dividend Plan	Extra Parameters	Historical Information
1	00/00/00	00/00/00				

Select field

FILE EDIT HOME DATA DEVELOPER HELP

Access Macros Dashboard Internet Security Corporate Action Black and Scholes VAR analysis

Locks Macros Dashboard Utilities Application Data Quality

List of Shares & Share Options

Security Code (Num.)	Security Name (long)	Security Code (ISIN)	Issue Date	F L G	Security Code (Reuters)	Curr.	Asset Type (CB RF)	Asset Type (O)
21	BOCH-ao	IE00BD5B1Y92	00/00/00			EUR	SHS6	ESV
22	Cian ADR Lev3	US83418T1088	00/00/00			USD	DR2	ED
23	Etalon Group GDR RegS	US29760G1031	00/00/00			USD	DR2	ED
24	Exxon Mobil Corp-ao	US30231G1022	00/00/00			USD	SHS6	ESV
25	FinEx Blockchain ETF USD	IE000GBF7ZU3	09/02/22			USD	SHS7	CE

Dividend Plan

Description/1	Description/2	Description/3	Classifications	Dividend Plan	Extra Parameters	Historical Information
1	00/00/00	00/00/00				

Black & Scholes - Share Option

Value Date: 00/00/00 Exercise Date: 00/00/00

Spot Price: Strike Price:

Volatility: Risk Free Interest Rate:

Call: Put:

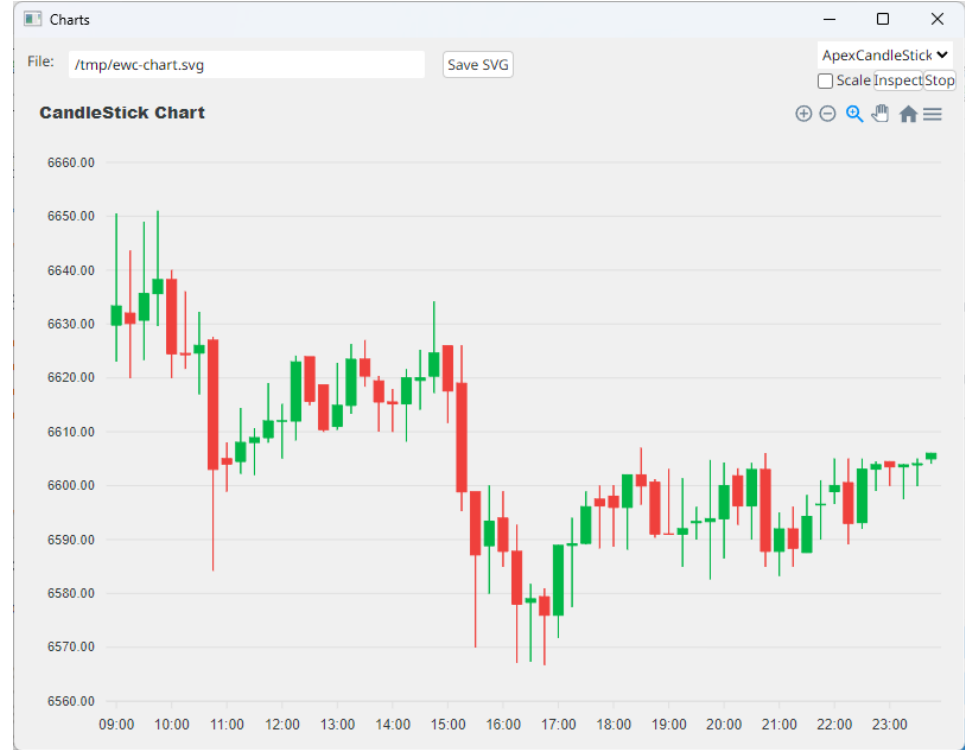
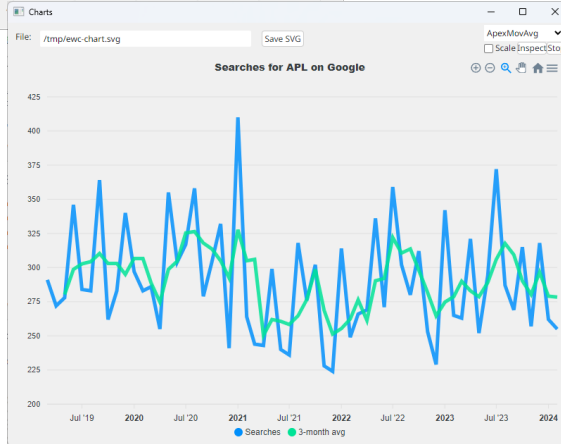
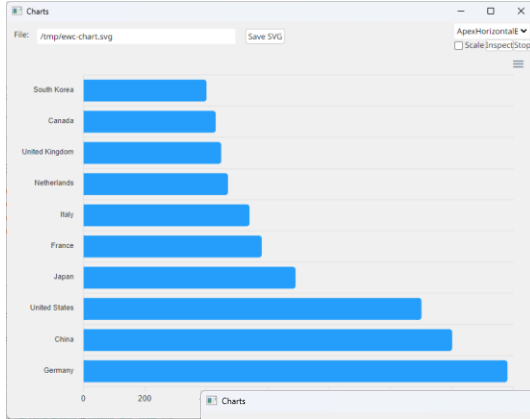
Delta Call: Delta Put:

Theta Call: Theta Put:

Vega: Gama: Duration: 0

Calculation Cancel Tools... Help

Additional Classes in EWC (ApexCharts)



Migration Tools

- **deltaWI:** Migrate □WI (APL+Win / MicroAPL) GUI to □WC (Dyalog APL)
- **EWC:** Migrate □WC (Windows) to Linux, macOS, and the Web
- **ATFIN:** Read APL Transfer Format files and perform translation to DyalogAPL
 - Includes emulation of many APL+Win system functions

Focus has been on working with launch customers, rather than publishing the tools – we will do that now.

- Davin Church will rewrite the □WI emulator this Winter



Rise of the ARM64

- ❖ v19.0 Raspberry Pi & "New" Apple Macs
- ❖ v20.0 ARM64 Linux w/ Docker containers
- ❖ v21.0 Possible Experiments
 - ❖ ARM64 Windows
 - ❖ Maybe even Android
 - ❖ Not "supported", but available for experimentation



.NET Generics

The use of generic classes and methods in .NET libraries is spreading like wildfire.

v20.0

```
□NEW GenericClass (43I632) T1 T2  
  (GenericClass (43I632) T1 T2).StaticMember args
```

v21.0

```
□NEW GenericClass[T1 T2]  
  GenericClass[T1 T2].StaticMember args
```

+ Export APL code with generic signatures



□SHELL to ~~replace~~ complement □CMD

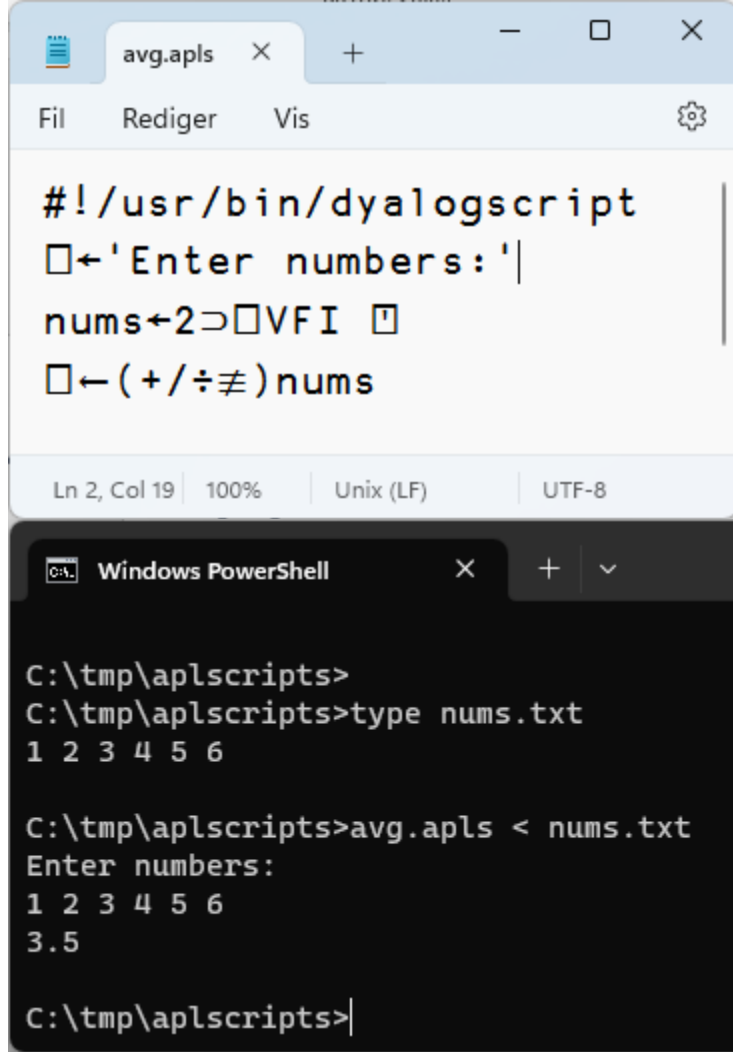
Execute External Programs from APL

- Interruptible
- Manage stdin, stdout & stderr (& other streams) independently
- Handle variety of data encodings
- Optionally return data as an asynchronous stream
- Select which shell to use
 - Defaults to PowerShell under MS Windows and /bin/sh elsewhere
 - Can also invoke programs directly, bypassing shell overhead



"Script Engine"

- ❖ #! (hash bang) scripting
- ❖ Script engine is popular with new users **AND** "Continuous Integration" engineers
- ❖ Has been gradually enhanced for several releases
- ❖ Will be further improved in v21.0
 - ❖ RIDE debugging of scripts



The image shows a code editor window titled 'avg.apls' with a menu bar (Fil, Rediger, Vis) and a settings icon. The code in the editor is:

```
#!/usr/bin/dyalogscript
☐←'Enter numbers: '|
nums←2☐☐VF I ☐
☐←(+ / ÷ ≠)nums
```

Below the editor is a Windows PowerShell terminal window. It shows the execution of the script 'avg.apls' with input from 'nums.txt'.

```
C:\tmp\aplscrip>
C:\tmp\aplscrip>type nums.txt
1 2 3 4 5 6

C:\tmp\aplscrip>avg.apls < nums.txt
Enter numbers:
1 2 3 4 5 6
3.5

C:\tmp\aplscrip>
```

Link Enhancements

Recent Highlights:

- -text=plain
 - Simple representations of character data
- -preloaded
 - Create links to pre-loaded workspace content
 - Useful when you have thousands of source files

To come:

- The "crawler"
 - To find things the File System Watcher will miss



Miscellaneous

- New Open-Source Documentation Format



Pete Donnelly
(finally retired)

Release Notes

v20.0 Release Notes

Introduction

System Requirements

Announcements

New and Enhanced Features

Minor Updates and Bug Fixes

Interoperability

Earlier Release Notes

Core Reference

Dyalog APL Language

Programming

Dyalog for Microsoft Windows

Installation/Configuration

User Interface

Object Reference

Interfaces

.NET Framework Interface

Dyalog for UNIX

Installation/Configuration

User Interface

About

Conventions

Third-party Licences

New Features, Changes, and Enhancements

This page describes the changes and new features in Dyalog v20.0 compared with Dyalog v19.0.

Syntax Changes

Array Notation

Array notation is a literal syntax for most arrays (including nested and high-rank arrays) and namespaces. Array notation is an extension of APL syntax, and, as such, can be used inside and around all other APL expressions, and wherever an APL expression can appear (for example in the Session, in functions, and in namespace scripts). It can also be used in the Editor to manipulate arrays directly.

You can edit variables using array notation in the following ways:

- Invoke the *Edit* command (**<ED>**) from within the Editor.
- Call the system command **)ED** and prefix the variable name with a diamond character, for example, **)ED ♦foo**
- Call the system function **⌈ED** with a left argument **'♦'**, for example, **'♦' ⌈ED 'foo'**.

In addition, in the Microsoft Windows IDE, array notation can be accessed in the following ways:

- Click the  icon in the **Session** toolbar – this button toggles on/off the use of array notation for output (when possible).
- Click the  icon in the **Object** toolbar when the cursor is over the name of an array – this opens the array in the Editor in the same way as **)ED ♦foo**.
- Click the  icon in the Editor's toolbar – this displays the contents of the Editor using array notation.
- Select **Show as Array Notation** from the Editor's **Syntax** menu – this displays the

On this page

Syntax Changes

Array Notation

Language Changes

Primitive Functions/Operators

System Functions

I-beams

Development Environment Changes

Inline Tracing

Configuration Parameters

Command Shortcuts

Home and End Keys

Microsoft Windows IDE

TTY Interface

Interfaces and Libraries

PCRE Library

.NET Interface

Open-Source!

Documentation

Release Notes

v20.0 Release Notes ✓

Introduction

System Requirements

Announcements >

New and Enhanced
FeaturesMinor Updates and Bug
Fixes

Interoperability

Earlier Release Notes >

Core Reference

Dyalog APL Language >

Programming >

Dyalog for Microsoft
Windows

Installation/Configuration >

User Interface >

Object Reference >

Interfaces >

.NET Framework Interface >

Dyalog for UNIX

Installation/Configuration >

User Interface >

About

Conventions

Third-party Licences

Welcome to the documentation for Dyalog v20.0!

This documentation site is a new project, and we are continuing to add documents from the full documentation set (which is available at the [Dyalog v20.0 Documentation Centre](#)).

 Hints and RecommendationsNew to APL? See [Getting started](#).

On this page

Help us improve this
documentation

Help us improve this documentation

If you discover an error on this site or would like to request an enhancement, [email us](#) your error report or enhancement request.

Alternatively, if you have a GitHub account, you can click the appropriate button below. We also welcome [direct content contributions](#).

**Content error**

For example: typos, broken links, inaccurate information, example doesn't work, images are too small

**Technical error**

For example, content not wrapping correctly, APL font not displaying, images failing with screen reader

**Content enhancement**

For example: additional examples for a primitive function, more screenshots of a process

**Technical enhancement**

For example: add a "dark mode" option, enhance search features, add ability to "bookmark" favorites

Miscellaneous

- New Documentation Format
- Kafka Interface
- □NINFO
 - Set File Attributes
 - Progress Callbacks for long-running moves & copies
- Platform Features



Pete Donnelly
(finally retired)

Platform Features: 407510

(undocumented – future PLATFORM)

```
(JSON Compact 0) 407510
{
  "CommandLine": {
    "Args": ["C:\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe",
             "CONFIGFILE=C:\\tmp\\dcfg\\200U64.dcfg"],
    "CodeArgs": ["C:\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe",
                  "CONFIGFILE=C:\\tmp\\dcfg\\200U64.dcfg"],
    "Full": ["C:\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe\\",
              "CONFIGFILE=C:\\tmp\\dcfg\\200U64.dcfg"],
  }
  "CurrentDirectory": "C:\\Users\\mkrom\\Desktop",
  "DirectorySeparator": "\\",
  "Executable": {
    "Bits": 64,
    "BuildTarget": "win",
    "BuildType": "opt",
    "Path": "C:\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe",
    "RideConnected": 0,
    "Runtime": 0,
    "Unicode": 1,
    "Version": [20, 0, 51721],
    "VersionMoniker": "200U64",
    "VersionNumber": 20,
  }
  "Features": {
    "DDE": 1,
    "DotNet": [4, 0, 30319],
    "Interactive": 1,
    "OLE": 1,
    "PCRE": [10, 44]},
}
```

```
"Host": {
  "ComputerName": "MKROM-LENOVOE16",
  "DNSDomainName": "dyalog.bramley",
  "EffectiveUserId": -1,
  "EffectiveUserName": "",
  "FQDN": "MKrom-LenovoE16.dyalog.bramley",
  "GroupId": "S-1-5-21-3838592348-3279234526-3710572232",
  "GroupName": "",
  "NetBIOSDomainName": "MKROM-LENOVOE16",
  "UserId": "1110",
  "UserName": "mkrom"},

"OS": {
  "Bits": 64,
  "Description": "Windows",
  "DirectorySeparator": "\\",
  "Family": "Windows",
  "LibcPath": "",
  "Newline": [13, 10],
  "NullDevice": "nul",
  "PathSeparator": ";",
  "SharedLibraryExtension": ".dll",
  "TempDirectory": "C:/Users/mkrom/AppData/Local/Temp",
  "UTCOffset": 2,
  "Version": [10, 0, 26100],
  "VolumeSeparator": ":"},

"Process": {
  "CurrentDirectory": "C:\\Users\\mkrom\\Desktop",
  "Id": 10024,
  "InitialDirectory": "C:\\Users\\mkrom\\Desktop",
  "LaunchTarget": "",
  "ParentId": -1},

"Runtime": 0,
"Unicode": 1,
"VersionNumber": 20}
```

```

    (["JSON" 'Compact' 0) 407510
{
  "CommandLine": {
    "Args":      ["C:\\\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe",
                  "CONFIGFILE=C:\\\\tmp\\dcfg\\200U64.dcfg"],
    "CodeArgs": ["C:\\\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe",
                  "CONFIGFILE=C:\\\\tmp\\dcfg\\200U64.dcfg"],
    "Full":      "\"C:\\\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe\"
                  CONFIGFILE=\"C:\\\\tmp\\dcfg\\200U64.dcfg\",

  "CurrentDirectory":  "C:\\\\Users\\mkrom\\Desktop",
  "DirectorySeparator": "\\ ",

  "Executable": {
    "Bits":             64,
    "BuildTarget":      "win",
    "BuildType":        "opt",
    "Path":             "C:\\\\...\\Dyalog APL-64 20.0 Unicode\\dyalog.exe",
    "RideConnected":    0,
    "Runtime":          0,
    "Unicode":          1,
    "Version":          [20, 0, 51721],
    "VersionMoniker":   "200U64",
    "VersionNumber":    20},

```

Less Usual Business

- Language enhancements
 - Array Notation
 - VGET & VPUT
 - Reverse Compose
- Inline Tracing



Less Usual Business

- Language enhancements
 - Array Notation
 - VGET & VPUT
 - Reverse Compose
- Inline Tracing



Array Notation

```
z ← c 'Three'  
z ← c 'Blind'  
z ← c 'Mice'
```

=

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

```
z ← 0 6 1 8  
z ← 1 4 1 4  
z ← 2 7 1 8  
z ← 3 1 4 2
```

=

```
[0 6 1 8  
1 4 1 4  
2 7 1 8  
3 1 4 2]
```

```
z ← 10  
z ← 20  
z ← 30  
z ← 40
```

=

```
[10  
20  
30  
40]
```

Namespace Notation

□VGET and □VSET

```
people← (name: 'Jack' ♦ weight:75) (name: 'Jill')
```

```
people.name
```

```
Jack  Jill
```

```
people.weight
```

```
VALUE ERROR: Undefined name: weight
```

```
people.weight
```

```
^
```

```
people □VGET 'name' ('weight' 50) # Default weight to 50
```

```
Jack  75  Jill  50
```

```
people[1].□VGET -2 # Get all class 2 names
```

```
name  Jack      weight  75
```

```
]repr (people/~40<people.age)[]VGET'name'  
(0p<'')  
]repr (people/~40<people.age)[]VGET'age'  
(0p<'')  
]repr (people/~40<people.age)[]VGET<'age' 0  
⍉  
people[1].children
```

Reese	Quinn	Jamie
-------	-------	-------

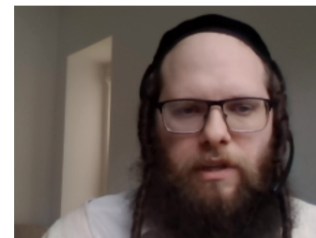
```
people.children  
VALUE ERROR: Undefined name: children  
people.children  
  ^
```

```
people.(40[]ATX'children')  
2 2 2 0 2 0  
people[]VGET <'children' (0p<'')
```

Reese	Quinn	Jamie	Taylor	Emerson	Sage	Rowan	Avery	Finley
-------	-------	-------	--------	---------	------	-------	-------	--------

```
people.children←people[]VGET <'children' (0p<'')
```

Adam Brudzewsky
v20.0 Videos



Ready...

00:03:25

CurObj: peoplepeople (Undefined)

nss

Ins NUM

00:09:28

8:1

DQ:0

TRAP

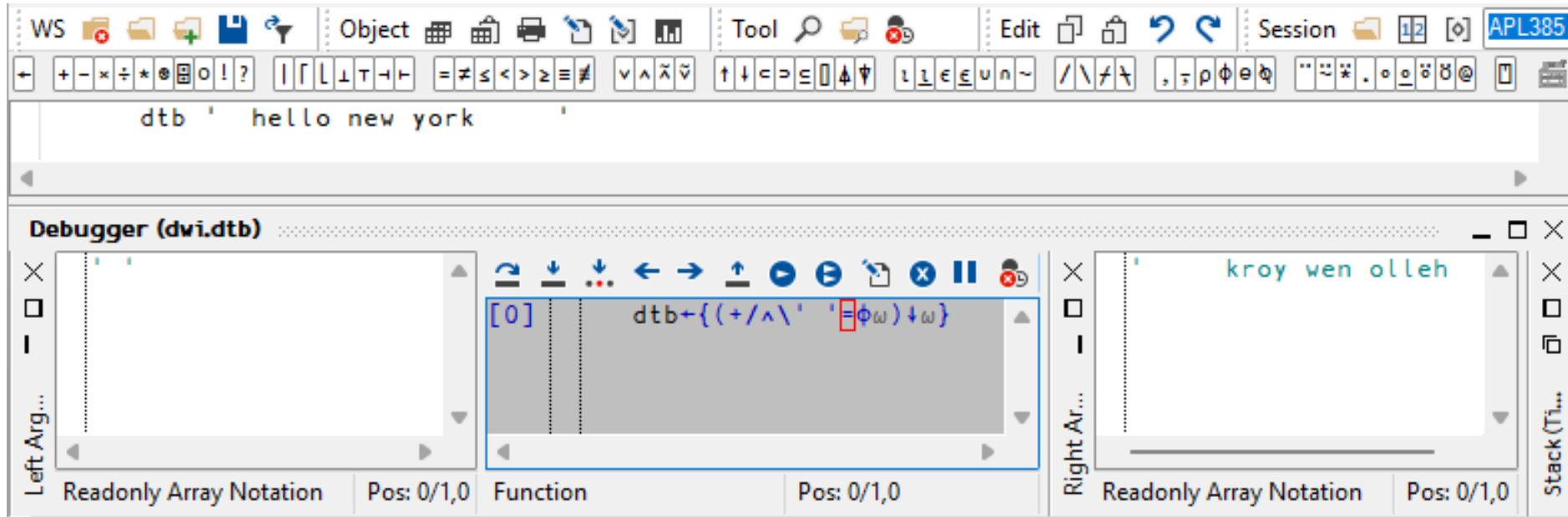
SI:0

IO:1

ML:1

...

Inline Tracing



Behind / Reverse Compose

(the "missing combinator")

Monadic: $f \circ g \ \omega \iff (f \ \omega) \ g \ \omega$

```

f ← 5 ◦ <
f ◦ / 2 7 1 8 2 8
7 8 8

```

A Predicate function
 A Filter by f

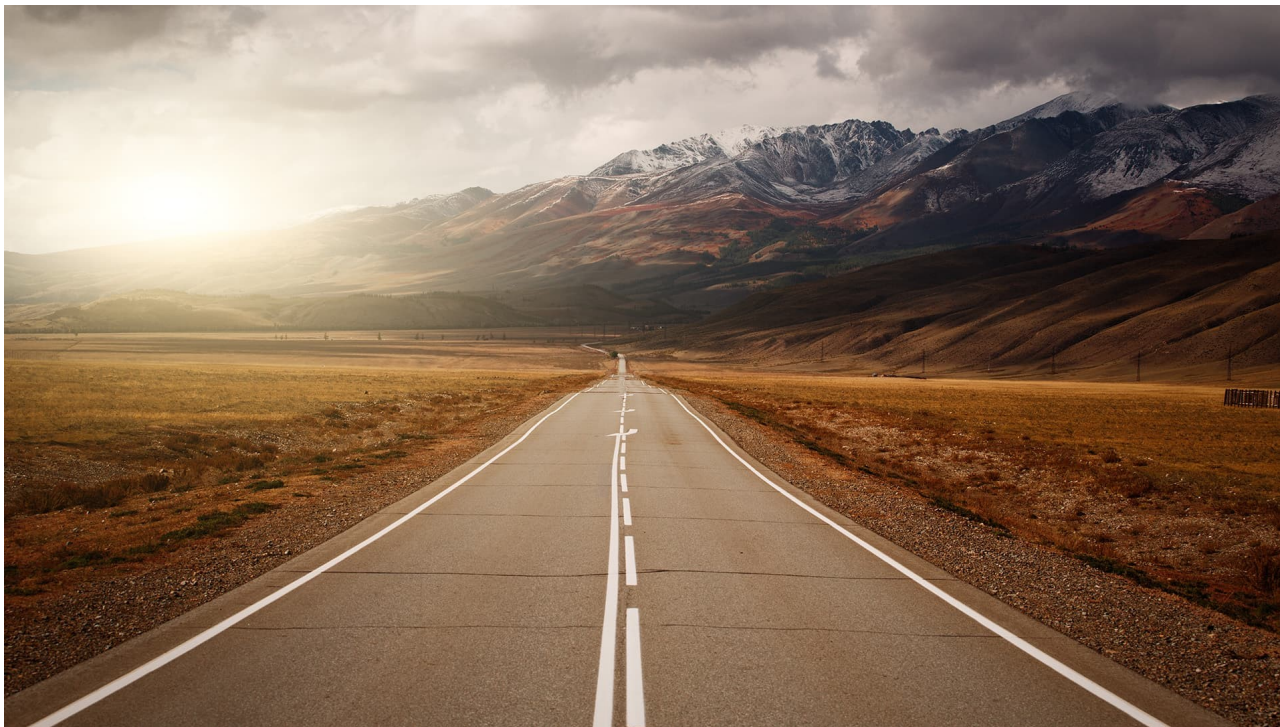
```

[ / ◦ = 2 7 1 8 2 8 3
0 0 0 1 0 1 0

```

A Max behind Equal

The Longer Perspective





Data Science

Async / Parallel

APL Libraries

AI / LLMs

Static Analysis

Compilation

APL Source Code as Text

- Foundation for much modern tooling
- Link 4.1 with v20.0 is "mature"
- v21.0 interpreter to load text source w/out Link
 - Create foundation for loading modules or libraries
- Link will be enhanced & maintained
 - To support features required by applications that started out with binary code structures



⊞IMPORT and :Insert

myns.apln

```
:Namespace myns  
  
  :Insert part1  
  :Insert part2  
  
  calc←{foo goo hoo ω}  
  
:EndNamespace
```

part1.apls

```
▽foo  
  ⊞←⊞NL -3  
  ▽  
  
▽goo  
  ...  
  ▽
```

part2.apls

```
▽hoo  
  ...  
  ▽
```

```
myns←⊞IMPORT 'myns.apln'  
myns.foo  
calc foo goo hoo
```

NB This syntax is
not confirmed!

APL Libraries

- Work on the package manager for APL (Tatin)
- Command-line interface to install packages
- Language support for loading modules will simplify implementation of Tatin
- Write more packages
 - Statistics libraries
 - General utilities
 - Other stuff



Open-Source Interfaces

We cannot open-source the language engine

- Most tools written **in APL** are free & open
 - And destined for the Package Manager
- We will now open-source some C code
- This will allow
 - Community Contribution
 - Inspection & Verification
 - Easier compliance with open-source licencing

Done

Kafka Interface

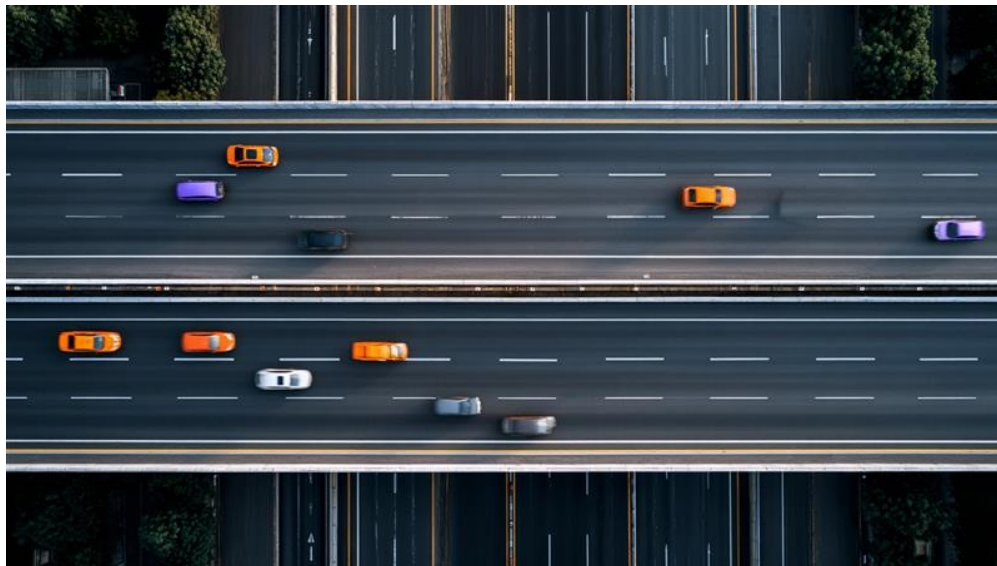
Planned for v21.0

HTMLRenderer (Chromium)
Conga (TCP / UDP)

"Later"

Cryptographic Library

Async / Parallel



Parallel / Async Operator ||

- ◆ Implement a primitive operator || which can invoke functions asynchronously
 - ◆ A future is immediately returned (as with isolates today)
 - ◆ The interpreter will block automatically if the value is used and has not materialised.
- ◆ Multiple architectures will be supported simultaneously
 - ◆ Isolates (remote processes w/TCP sockets)
 - ◆ "Green threads" (as the existing & operator)
 - ◆ Multiple interpreters running on separate O/S threads within the same process

Compilation

- The || operator will support parallelism using multiple interpreters
- Effective use of GPUs requires a compiler
- Enter co-dfns
- Or is it cod-fns ?



Static Analysis of APL Code

- Static Analysis of application code is seen as a required "best practice" by many corporations
- We are building a prototype of a tool which will
 - Detect vulnerabilities and other bad practices
 - "Lint" APL Code
- This tool will initially be licensed separately
 - A free "community edition" may follow later
- Will be tested by first two clients at the end of this year

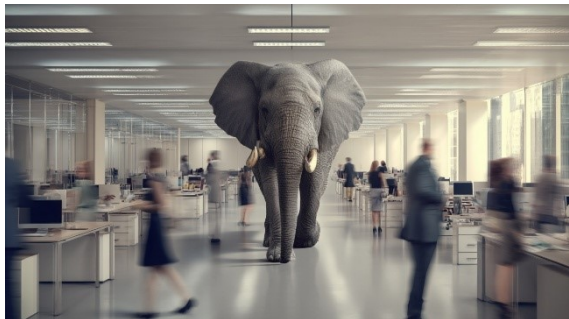


Large Language Models / AI



No presentation can be complete without mention of Artificial Intelligence

Large Language Models / AI

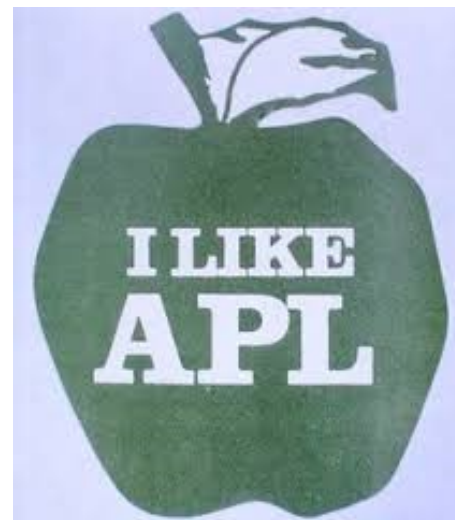


No presentation can be complete without mention of Artificial Intelligence

- Mark Wolfson will talk about how they can be used to add functionality to APL systems
- Stefan Kruger is up next to talk about his journey, investigating the potential of current LLMs to help us develop APL code

Data Science

- APL used to be the best tool for DS



Data Science

- ◆ APL used to be the best tool for DS
- ◆ Today, Python has more, better libraries



Data Science

- ◆ APL used to be the best tool for DS
- ◆ Today, Python has more, better libraries
- ◆ We are building a team to understand
 - ◆ Where the opportunities are for APL
 - ◆ How to make ourselves more visible
 - ◆ Where we need to improve to be competitive
 - ◆ How to share tools & data with Python



Data Science Findings

We need to work on:

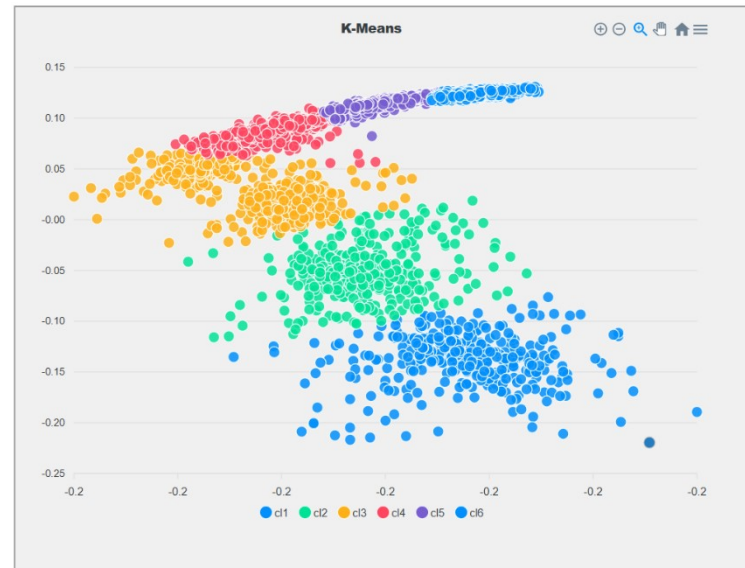
- ❖ Faster Data Loading
- ❖ Faster Matrix Multiplication
- ❖ Easy access to LLMs



Data Science Findings

We need to work on:

- ❖ Faster Data Loading
- ❖ Faster Matrix Multiplication
- ❖ Easy access to LLMs
- ❖ New Visualisation Tools



Data Science Findings

We need to work on:

- ❖ Faster Data Loading
- ❖ Faster Matrix Multiplication
- ❖ Easy access to LLMs
- ❖ New Visualisation Tools
- ❖ Arrow & Parquet Data Formats
- ❖ Python Integration
 - ❖ Use Python packages from APL
 - ❖ Use APL as a package from Python



[About the APL Forge](#)[Previous Winners](#)[Submissions](#)[Ideas](#)[FAQs](#)[Contact Us](#)[SUBMIT YOUR PROJECT](#)

Shaping tomorrow with APL

Forge your ideas into prize-winning APL tools

Do you have a problem and are looking for the right language in which to solve it? Do you have an idea for a library that you think other APL users might benefit from? Do you have an APL-based application that you want to share? The APL Forge is where we reward you for using Dyalog APL to solve problems and develop libraries, applications, and tools.

An APL Competition

The APL Forge

This annual event is designed to promote the use and development of APL by challenging participants to create innovative open-source libraries and commercial applications. Whether you're an individual, a group, or a company, if you have a passion for problem-solving in APL, this competition is for you.

APLForge Winners



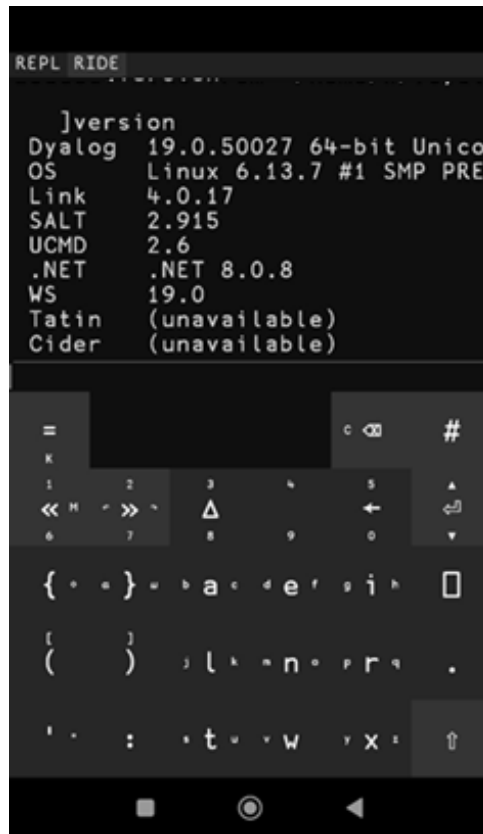
2024 Holden Hoover
Radar Ingest System



2025 Borna Ahmadzadeh
APLearn – Machine Learning

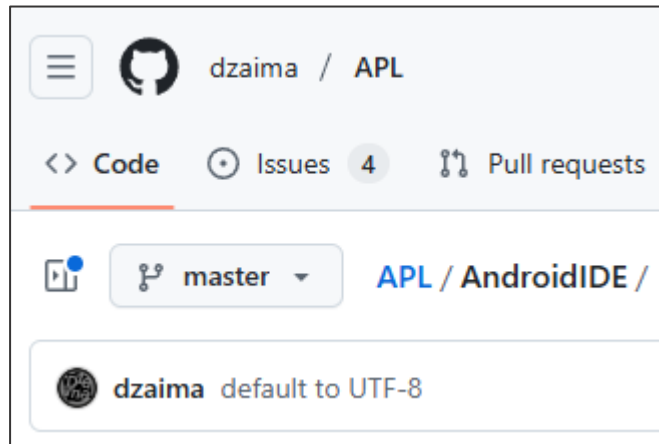
Open-Source helps Drive Engagement

- The Remote IDE (Ride) and the Ride protocol are open
- "dzaima" is a former Problem-Solving Contest winner and has his own APL interpreter
- Suddenly one day ...



Open-Source helps Drive Engagement

- The Remote IDE (Ride) and the Ride protocol are open
- "dzaima" is a former Problem-Solving Contest winner and has his own APL interpreter
- Suddenly one day ...



Building the Team

- Recruit in good time
 - New recruits provide much-needed inspiration to go after new business
 - But need to learn from the experienced crew
- Offer Consulting & Training
 - Stay in close contact with existing users
 - Help new users build modern APL systems



Recent Hires



Asher
APL+C
UK Oct 25
(Intern in 23 & 24)



Brandon
APL
Japan Jan 25



Martina
APL+C
Denmark Aug 24



Andrea
CEO Assist.
Denmark Oct 24



Neil
JS+APL
Germany Sep 24

60 FPS

Interns!



Asher



Kamila



Brian

raylibAPL



The Plan is Unchanged



Continue to grow the revenue base

Customers use of APL is growing
Migrants keep arriving



Grow and replenish the team

Hire new people in good time
before old-timers retire



Hunt the new generation of users

Get back into data science & education
Continue to position APL in the functional community



A big THANK YOU to ...

- Mike Mingard provided invaluable help with the images.
- A ton of Admin work behind the scenes by Karen Shaw and Jada Andrade



Behind and on
the scene:
Brian Becker