

News from Dyalog – Spring 2022

Morten Kromberg



APL Germany

Agenda

- Building the Team
- The next Big Thing
- The next, next Big Thing

- Version 18.2 recap
- Version 19.0 and beyond



[Re]Building the Team

- We stand upon the shoulders of Giants



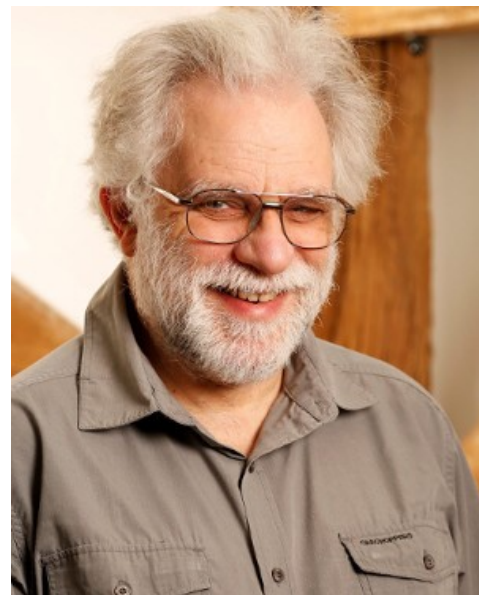
John Scholes (1948-2019)



Roger Hui (1953-2021)

Still Going Strong...






- Geoff Streeter wrote Dyalog APL v1.0 in 1981-1983 together with John Scholes
- Geoff is in good health
 - (Now working 3-4 days / week)
 - And still volunteering at night...



Still Going Strong...

- Geoff Streeter wrote Dyalog APL v1.0 in 1981-1983 together with John Scholes
- Geoff is in good health
 - (Now working 3-4 days / week)
 - And still volunteering at night...
- BUT: He has announced that he intends to retire in April'23



 <p>Gitte Managing Director (CEO)</p> <p>Based in: </p> <p>Joined Dyalog: April 2005</p> <p>Gitte discovered APL in 1983. After graduating as a research biologist, she was offered 6 months of training in EDP (Electronic Data Processing). When exposed to APL she immediately fell in love and never looked back. Since then she has worked as an APL consultant, headed development and sales of commercial software based on APL and finally in 2005 ended up where she started, at an APL language vendor – but this time as the CEO.</p>	 <p>Geoff</p> <p>Based in: </p> <p>Joined Dyalog: January 1977</p> <p>Geoff is a '60s radical who has yet to repent of his optimism. The attitude pervades his approach to Christianity, his politics and his computing.</p>	 <p>Pete</p> <p>Based in: </p> <p>Joined Dyalog: September 1977</p> <p>Pete had the amazing luck to be a member of the team, alongside John Scholes and Pauline Brand, that acquired what was then Dyalog Systems in 1990 and guided it through the formative years of Dyalog APL. His only truly wise business decision was to hand over the reins in 2005 and he now luxuriates in partial retirement on the Greek island of Alonissos. When not writing Dyalog documentation he struggles to learn to speak Greek and to play the bouzouki. He hopes one day to become semi-competent at both.</p>
 <p>Morten Chief Technical Director (CTO)</p> <p>Based in: </p> <p>Joined Dyalog: April 2005</p> <p>Morten joined Dyalog after spending about 15 years building up ideas for technical direction as a user of Dyalog – 10 as an APL Consultant and 5 as the CTO of Adaryum Software, developing a "shrink-wrapped" product which used Dyalog as the core technology. Before that, he spent about 10 years learning, using and helping clients install and maintain SHARP APL at J.P. Sharo Associates.</p>	 <p>Jason I.T. Manager</p> <p>Based in: </p> <p>Joined Dyalog: May 2007</p> <p>Jason is one of the younger members of the team though he tries to disguise his youth with hair. An "expert on everything" according to those in the know, he winces whenever anyone "www." in a URL.</p>	 <p>Rich</p> <p>Based in: </p> <p>Joined Dyalog: October 2018</p> <p>Rich discovered APL shortly after discovering Dyalog headquarters on his own doorstep in Bramley. With a degree in physics and interest in computing, he was surprised to have not heard of it before. He hopes to introduce many more people to APL so that fewer future experts have to be deprived of APL's prowess in conveying problem solutions to computation engines. He's wondering when is an appropriate time to tell them about his own personal duck collection.</p>
 <p>Adám</p> <p>Based in: </p> <p>Joined Dyalog: May 2015</p> <p>Adám prides himself on having got APL with the breast-milk, as he attended his first APL conference at the age of one. He is adamant in many of his convictions, including that APL is by far the world's most superior programming language, and the only one worthy of being used day-to-day. His work is mainly the development and maintenance of APL tools.</p>	 <p>JohnD Chief Architect</p> <p>Based in: </p> <p>Joined Dyalog: September 1991</p> <p>If memory serves, which it often doesn't, John joined Dyalog Ltd fresh faced and eager after "Computer aided Chemistry" from www.surrey.ac.uk/chemistry/. John has been responsible for the User Interface work with Dyalog, including the GUI session, COM and .NET interfaces. J the corner in the Dyalog office and can frequently be heard shouting at his computer.</p>	 <p>Richard Development Manager</p> <p>Based in: </p> <p>Joined Dyalog: June 2008</p> <p>Richard joined Dyalog with a degree in computer science from City, University of London, and 20 years' experience of compiler development and language design. He develops and maintains the Dyalog interpreter and oversees its overall development.</p>
 <p>Andy Operations Manager (COO)</p> <p>Based in: </p> <p>Joined Dyalog: July 1987</p> <p>Andy has spent most of his working life in support, supporting APL but also UNIX systems and high-end hardware. He is less the theorist and more the pragmatist. He is responsible for Dyalog's internal system as well being involved with the development and QA processes.</p>	 <p>Josh</p> <p>Based in: </p> <p>Joined Dyalog: June 2019</p> <p>Josh was exposed to Dyalog APL during an internship with The Carlisle Group. He continued and was a grand prize winner in the 2019 APL problem solving competition. A recent gradua University of Soranton with a degree in Computer Science, he now primarily serves as an AF to North American clients and works on APL tools.</p>	 <p>Rodrigo</p> <p>Based in: </p> <p>Joined Dyalog: February 2021</p> <p>Rodrigo stumbled on APL by accident in early 2020, as he posted programming challenges in an online code golf community. APL's simplicity, brevity, and affinity with mathematics got him hooked. Now Rodrigo is exercising his passions for teaching, programming, and mathematics, in an attempt to get many new people interested in APL.</p>
 <p>Björn</p> <p>Based in: </p> <p>Joined Dyalog: April 2005</p> <p>Björn has been making add-ons for APL since the start of the '90s, first at Insight Systems ApS, then at Adaryum Software and now at Dyalog Ltd. He concentrates on moving things that are hard to do in APL or that have poor execution speeds into C so that they are callable from APL. The best known examples of these are SQAPL and Conga.</p>	 <p>Karen Customer Account Manager</p> <p>Based in: </p> <p>Joined Dyalog: May 1992</p> <p>Karen has been involved with Dyalog and APL for over 20 years. With an alternative job title Herder, her administration and organisational talents ensure that everything behind the scene Ltd is running smoothly.</p>	 <p>Ron</p> <p>Based in: </p> <p>Joined Dyalog: July 2020</p> <p>Ron first encountered APL/360 in 1969, and then used it to teach Computer Science courses in the high schools of Hampton, Virginia. Over the years he was involved with several different APL implementation projects before doing a few non-APL things with Microsoft, Amazon, and an Internet Television company. During all those ventures APL has served him well as a tool for thought.</p>
 <p>Brian APL Tools Architect</p> <p>Based in: </p> <p>Joined Dyalog: March 2011</p> <p>Brian learned APL in 1975 when as a freshman in high school he discovered that the typewriter that knw how to play football was actually a computer terminal hooked up to the University of Rochester's APL timesharing system. Brian has developed APL applications across a variety of industries including financial planning, manufacturing control, legislative tracking, full text databases and risk assessment. Brian has twice been recognised by the ACM's SigAPL for innovative APL development. "I'm very blessed to be a part of the Dyalog team; building tools that make it easier for our customers to do their jobs."</p>	 <p>Karta</p> <p>Based in: </p> <p>Joined Dyalog: April 2021</p> <p>Karta completed his PhD in Theoretical Physics and stumbled upon APL entirely by happenst joined Dyalog as a C/C++ programmer, with the aim of developing its utility for scientists and</p>	 <p>Stine</p> <p>Based in: </p> <p>Joined Dyalog: May 2020</p> <p>Stine (Kirstine) grew up around APL and her first encounter with many of Dyalog's customers was at the help desk whenever the user meetings were held in Denmark. Although her experience with APL is sadly lacking, her interest in management and accounting are being put to good use.</p>
 <p>Fiona Documentation Manager</p> <p>Based in: </p> <p>Joined Dyalog: April 2013</p> <p>Fiona joined Dyalog with twelve years of technical authoring experience but a programming background limited to Pascal and FORTRAN. She wishes she'd known about APL and Dyalog earlier in her academic life...</p>	 <p>Michael</p> <p>Based in: </p> <p>Joined Dyalog: February 2017</p> <p>Michael got in contact with APL in the last century through his first job as Production Plannin supporting an APL application. The speed in which development implemented any changes curious about that strange language – and the rest is history. In his rare moments away from computer, Michael likes to spend time with his family, enjoys travelling, music and boring car games as well as board-games. He also loves British humour (claims to sometimes even un English tea and Danish cookies. Sometimes he sleeps.</p>	 <p>Vince</p> <p>Based in: </p> <p>Joined Dyalog: June 1996</p> <p>Vince enjoys helping customers. He is mainly a C programmer, but loves having Dyalog on his desktop as it's a really useful tool to do a quick calculation or write a quick program in. His computer runs Windows, and he uses command-line shells as well as the GUI.</p>

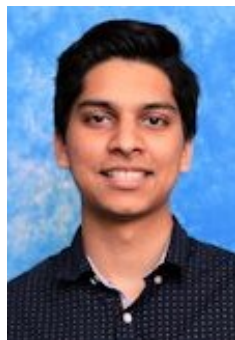
Dyalog ... The Next Generation



Adam
APL



Rich
Education



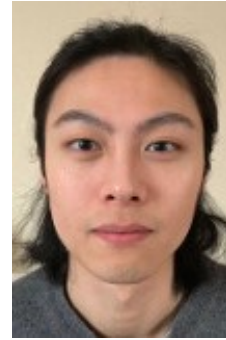
Josh
Consultant



Rodrigo
Education



Stine
Accounting



Shuhao
Interpreter



Karta
Interpreter

Dyalog ... The Next Generation



Adam
APL



Rich
Education



Josh
Consultant



Rodrigo
Education

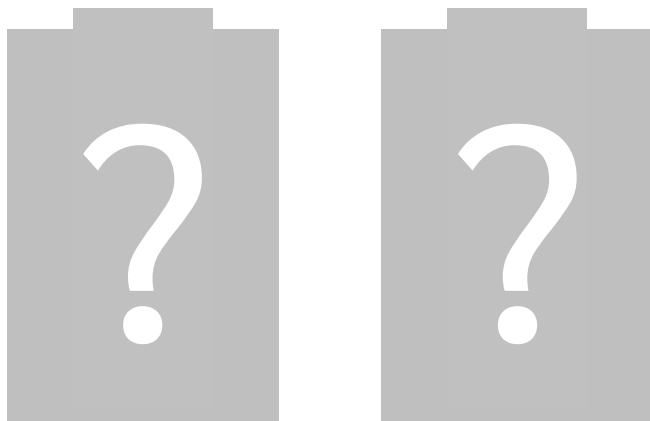


Stine
Accounting



Karta
Interpreter

Next Generation, Continued...



- We planned to recruit a tester and one more interpreter developer in 2022



Next Generation, Continued...



- Now, we plan to recruit a tester and **two** more interpreter developers in 2022



Some quotes from recent applicants...



Passionate about APL, able to create non-trivial applications with it. Improving my tacit, leading-axis, and array-oriented skills daily.

Active on Stack Overflow, the APL Orchard, and in the APLFarm, I enjoy teaching, helping, and problem solving with others when given the opportunity.



I've recently discovered APL and other various array languages and projects around them.

Would be amazing to work for Dyalog, implementing and improving core feature set.

I'm aware don't have great deal experience with large codebases but have enjoyed exploring Plan 9's source code and understanding how sections fit together.



The area in computer science that interests me the most is programming language design and implementation.

I have written a few interpreters in my spare time for Prolog and for APL.

I am always interested in learning new topics in computer science and learning new ways to apply my knowledge. Last year I participated in the Dyalog APL Problem Solving Competition and won a phase 1 prize, and I am also participating this year again.



All these applicants (our two most recent hires before that) approached us to inquire about work at Dyalog.

Dyalog ... The Next Generation



The Next Big Thing

No, not more product features...

- MUCH more documentation, samples and tutorials
- (there will also be lots of features too, of course)



Training Materials

🟡 mastering.dyalog.com

Rodrigo Girão Serrao



🟡 course.dyalog.com

Rich Park



🟡 tutorial.dyalog.com

Gary Bergquist + Andrew Sengul



🟡 xpqz.github.io/learnapl

Stefan Kruger (IBM)



Adam Brudzewsky
(here, there and
everywhere)



The Next, Next Big Thing

- Package Management



Tatin





List of all packages (aggregated)

Package name	Description	Major versions	Link to project
aplteam-APLGit	Git interface from Dyalog APL via Git Bash	1	github.com
aplteam-APLProcess	Start an APL process from within Dyalog APL	1	github.com
aplteam-APLTreeUtils2	General utilities required by most members of the APLTree library	1	github.com
aplteam-CodeCoverage	Monitors which parts of an application got actually executed	1	github.com
aplteam-Compare	Allows comparing and merging objects in the workspace with a file or a file with another file	1	github.com
aplteam-CompareSimple	Allows comparing objects in the workspace with a file or a file with another file	1	github.com
aplteam-DateAndTime	Utilities related to Date and Time, including doing math	1	github.com
aplteam-DotNetZip	Zippping and unzipping with .NET Core on all major platforms	1	github.com
aplteam-EventCodes	Constants with meaningful names for Dyalog error codes	1	github.com
aplteam-Execute	Start a process from within APL	1	github.com
aplteam-FilesAndDirs	Utilities for doing gymnastics with files and directories	1	github.com
aplteam-GitHubAPIv3	Utilities for dealing with GitHub repositories	1	github.com
aplteam-HandleError	Allows to catch errors on an application level; saves information that allow analyzing the error	1	github.com
aplteam-IniFiles	Allows instantiating good old INI files in APL; comes with extended syntax supporting APL-like data structures	1	github.com
aplteam-Lagunitza	Managing and displaying help pages based on markdown files	1	github.com
aplteam-Logger	Allows writing to LOG files; (almost) guaranteed to never break the application	1	github.com
aplteam-MarkAPL	Converts Markdown to HTML5	1	github.com
aplteam-OS	OS-related tools for all major platforms	1	github.com
aplteam-SMTP	SMTP client for sending emails from within Dyalog APL	1	github.com
aplteam-ServiceState	Interface between the Windows SCM (Service Control Manager) and applications running as a service	2	github.com
aplteam-SevenZip	Zip files with the Open Source tool 7Zip	1	github.com
aplteam-ShowChmHelp	Display CHM files under program control (Windows only)	1	github.com
aplteam-Tester2	Dyalog APL test framework	1	github.com
aplteam-WinReg	Tools for dealing with the Windows Registry	1	github.com
aplteam-WinRegSimple	Limited set of tools for dealing with the Windows Registry	1	github.com
aplteam-WinSys	Windows-only OS calls and system infos	1	github.com
aplteam-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
davin-DateTime	Easy calculations with dates	1	github.com
davin-FilePlus	Extend component files to use named components	1	github.com
davin-SQLFns	Easily create text SQL commands for use with any SQL program interface	1	github.com
davin-Tester	Simplified function-level testing of programs	1	github.com

Tatin

← → ↺

tatin.dev/v1/groups

📄 Apps 📄 Link 📄 APL 📄 Flying & Sailing

Tatin Registry

List of all groups

Group name	≡ ^[1]
aplteam	28
davin	4
dyalog	0
tiamatica	0
bluedolphin	0

1. Number of published packages of that group

Tatin Test Registry

List of all packages (aggregated)

You may publish any packages to the Tatin Test Registry with the API key "Tatin-Test-API-Key" except for the groups "dyalog" and "aplteam". However, note that the data of the Test Registry is occasionally reset.

Package name	Description	Major versions	Link to project
aplteam-MultipleObjectsNoAPI	Multiple objects of all types, no API ◦ must be a folder ◦ Exposes everything in the folder	1	-
aplteam-versions	This and that	2	-
dyalog-HttpCommand	Utility to execute HTTP requests	1	dyalog.github.io
dyalog-Jarvis	JSON and REST Web Service Framework	1	dyalog.github.io
example-Foo	Has three depedencies, one lives on a different server	1	-
example-Goo	Has NO dependencies but a file apl-dependencies.txt (empty)	1	-
example-Moo	Has NO dependencies and NO file apl-dependencies.txt	1	-
example-Zoo	Has NO dependencies and NO file apl-dependencies.txt	1	-
example-versions	This and that	2	-

The other Big Thing

- Package Management



Tatin

- Demo tomorrow



Recap of v18.2

(March 2022)

- Windows Patches
- Basic Licence
- Launching from Text Files:
 - Shell Scripts
 - Text Source
 - LOAD Parameter
 - Windows Explorer Integration
- Improved Docker Support
- RIDE v4.4
- Developer Productivity:
 - JSON tables/datasets
 - □ATX metadata one-stop-shop
 - Generating random numbers
 - New user commands



Windows Patches

Updates delivered as Microsoft Patch files (MSP) under Windows

- ✧ Allows patching ALL components of Dyalog installation including APL code & docs
- ✧ All components were already patched by re-installing on non-Windows platforms








My Dyalog

← → ↺ my.dyalog.com/#Home

Apps Link APL Flying & Sailing Car Dyalog Cloud SBO Travel Linux Sport Productivity Git Covid

MYDYALOG

Home Fixes Downloads Help My Account Logout



Welcome to MyDyalog

Welcome to MyDyalog. Please check the top of the page to access the part of the site that you want to view. Notices will be posted to this page when there are major security updates or for an add-on package (for example, SQAPL/Conga).

DFS (Dyalog File System) cannot currently be downloaded using MyDyalog; if you have a DFS licence and want to download it, please contact sales@dyalog.com

Downloads

Dyalog
RIDE
Tools & Interfaces
Patches

Latest News

18.2 is released to all users

23rd March 2022

Dyalog APL Version 18.2 is now available for DSS users. It is available for all platforms.

The installation images have been updated to remove a problem with the LOAD= parameter; more details appear in the DSS email. The Windows patch files include the fix for the issue, so if you already have 18.2 installed you can simply download them and apply the patch. The patch wizard files do not include the entire fix, so we strongly recommend using the patchfiles.

Note that for 18.2 onwards on Windows we will be issuing patches for all the files that are part of Dyalog APL, as well as still supported the old patch wizard. These are available from the [Update Existing Windows Installations page](https://my.dyalog.com/#DownloadPatches).

<https://my.dyalog.com/#DownloadPatches>

My Dyalog

my.dyalog.com/#Home

AppsLinkAPLFlying & SailingCarDyalogCloudSBOTravelLinuxSportProductivityGitCovid

Dyalog Install Temp

New

Copy

Move

Share

Sort

View

This PC > Windows (C:) > tmp > Dyalog Install Temp

Quick access

Desktop

Downloads

Documents

Pictures

Awaiting SSG review

Design Docs

SSGMon

Name	Date modified	Type	Size
patch_18.2.45588.0_64_unicode_2022.05.02.exe	02-05-2022 23:42	Application	2.102 KB
patch_18.2.45588.0_64_unicode_2022.05.02.msdl	02-05-2022 23:42	MSD File	150.704 KB

Latest News

18.2 is released to all users

23rd March 2022

Dyalog APL Version 18.2 is now available for DSS users. It is available for all platforms.

The installation images have been updated to remove a problem with the LOAD= parameter; more details appear in the DSS email. The Windows patch files include the fix for the issue, so if you already have 18.2 installed you can simply download them and apply the patch. The patch wizard files do not include the entire fix, so we strongly recommend using the patchfiles.

Note that for 18.2 onwards on Windows we will be issuing patches for all the files that are part of Dyalog APL, as well as still supported the old patch wizard. These are available from the [Update Existing Windows Installations page](#).

My Dyalog

my.dyalog.com/#Home

AppsLinkAPLFlying & SailingCarDyalogCloudSBOTravelLinuxSportProductivityGitCovid

Dyalog Install Temp

New

SortView

This PC > Windows (C:) > tmp > Dyalog Install Temp

Quick access

DesktopDownloadsDocumentsPicturesAwaiting SSG reviewDesign DocsSSGMon

2 items

Name	Date modified	Type	Size
Dyalog APL-64 18.2 Unicode Patcher			
<div>This will patch the following components of your Dyalog installation</div> <div><div>The Dyalog development environment and related files will be updated to version 18.2.45588.0</div><div>Enable Installer Logging</div></div> <div>PatchExit</div>			

Latest News

18.2 is released to all users

23rd March 2022

Dyalog APL Version 18.2 is now available

The installation images have been updated to remove a problem with the LOAD= parameter; more details appear in the DSS email. The Windows patch files include the fix for the issue, so if you already have 18.2 installed you can simply download them and apply the patch. The patch wizard files do not include the entire fix, so we strongly recommend using the patchfiles.

Note that for 18.2 onwards on Windows we will be issuing patches for all the files that are part of Dyalog APL, as well as still supported the old patch files. These are available from the [Update Existing Windows Installations page](#).

Basic Licence

Possibly the most important v18.2 "feature"!

Intended for:

- ✧ non-commercial use
- ✧ education
- ✧ personal projects
- ✧ experiments
- ✧ sharing your experience
- ✧ proof of concepts / trials
- ✧ participating in programming competitions for cash prizes
- ✧ fun



Basic Licence

- Replaces *non-commercial licence*
- Allows distribution of Dyalog along with your work, under the default royalty licence
 - Fee is 2% of gross APL-based revenue
 - No fees due if revenue < GBP 5,000 in a calendar year
 - Multiple alternative commercial licence schemes are available
- For GBP 150 per year, you can subscribe to the Dyalog Support Service (DSS)



Launching from Text Files

Another important 18.2 theme ...

- ◆ Shell scripts
- ◆ LOAD= parameter supports starting the interpreter directly from text source
- ◆ Link properly settling down with v3.0
- ◆ Windows Explorer Integration
- ◆ Improved Docker support

... demo tomorrow ...



RIDE v4.4

- 🟡 Debug multi-threaded applications
- 🟡 Language bar and Help adapts to Dyalog version
- 🟡 Save responses to prompts
- 🟡 ... and many more tweaks and fixes, see <https://github.com/Dyalog/ride/milestone/4?closed=1>



Developer Productivity:

JSON tables/datasets

A Convert "Inverted" Matrix to JSON

```
fields ← 'Item' 'Price' 'Qty'  
items  ← 'Knife' 'Fork'  
price  ← 3 4  
qty    ← 23 45
```

fields;↑items price qty

Item	Price	Qty
Knife	3	23
Fork	4	45

```
inverted ← 4 A Data Format Code  
data ← items price qty  
table ← < inverted (data fields)  
(⌈JSON⌋'Compact'0)table
```

```
[  
  {  
    "Item": "Knife",  
    "Price": 3,  
    "Qty": 23  
  },  
  {  
    "Item": "Fork",  
    "Price": 4,  
    "Qty": 45  
  }  
]
```



Developer Productivity:

□ATX metadata one-stop-shop

✧ Unifies:

- ✧ □SIZE Size of Object
- ✧ □NC Name Classification
- ✧ □NR Nested Representation
- ✧ □SRC Source
- ✧ □AT Attributes
- ✧ 5179⊞ Loaded Object Info

✧ Additional metadata:

- ✧ Last edit UTC
- ✧ Source "as typed"
- ✧ Normalised source
- ✧ Best available source
- ✧ More to come...



?	Welcome
?	Release Notes V18.2
?	Introduction
?	Shell Scripts
?	Language Reference Changes
?	Extended Attributes
?	JSON Extension
?	Fix Script
?	Sample Probability Distribution
?	Set Shell Script Debug Options
?	Object Reference Changes
?	Windows Installation and Configuration Gu
?	Configuring the IDE
?	Programming Reference Guide
?	Language Reference Guide
?	Object Reference
?	Windows UI Guide
?	Interface Guide
?	.NET Framework Interface Guide
?	Old Release Notes
?	Licences for third-party components
?	Trademarks

Extended Attributes

R+X □ATX Y

This function provides information about a name in a workspace, including its usage, history, restrictions, role and origin.

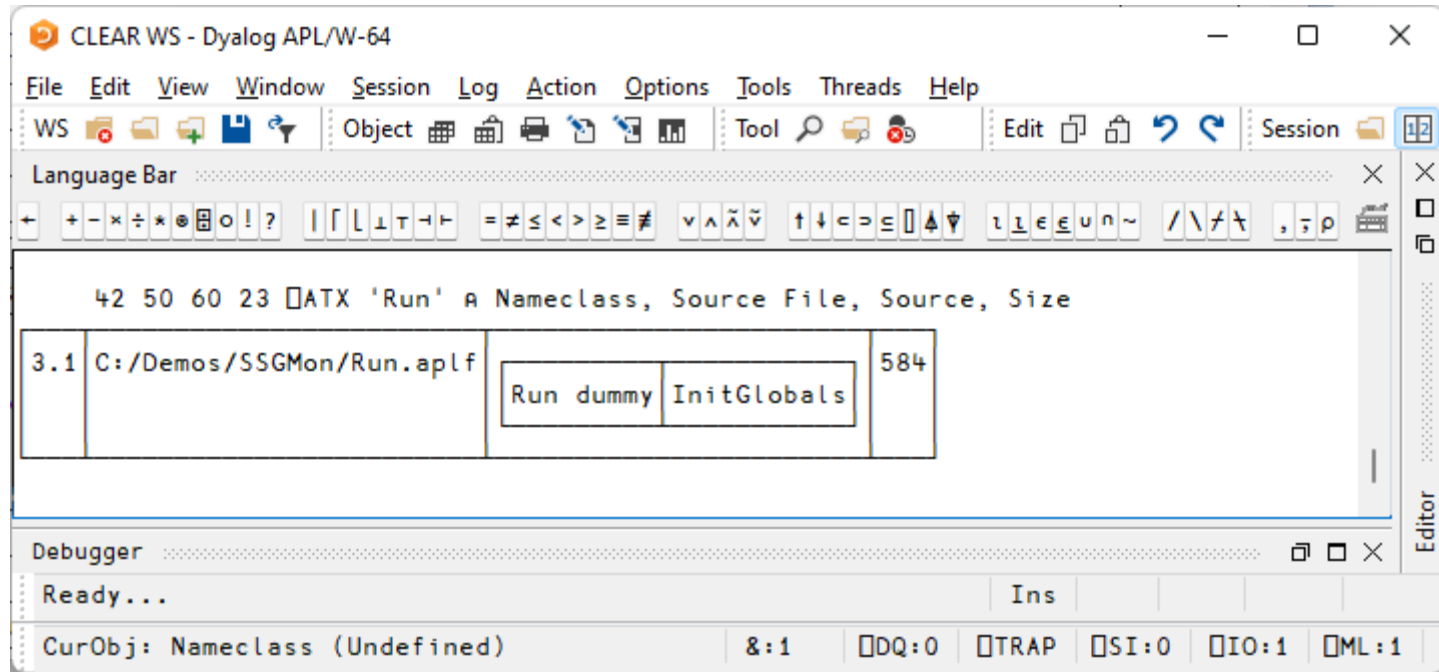
Note: To retrieve this information for an unnamed value, wrap `□ATX` in a dfn and use the name `u`, for example `{60□ATX'u'}`

Y can be a simple character scalar, a simple or enclosed character vector, or a vector of character scalars and vectors (as least one must be a character vector) of the name(s) for which information is required.

X can be a scalar or a vector indicating the information required:

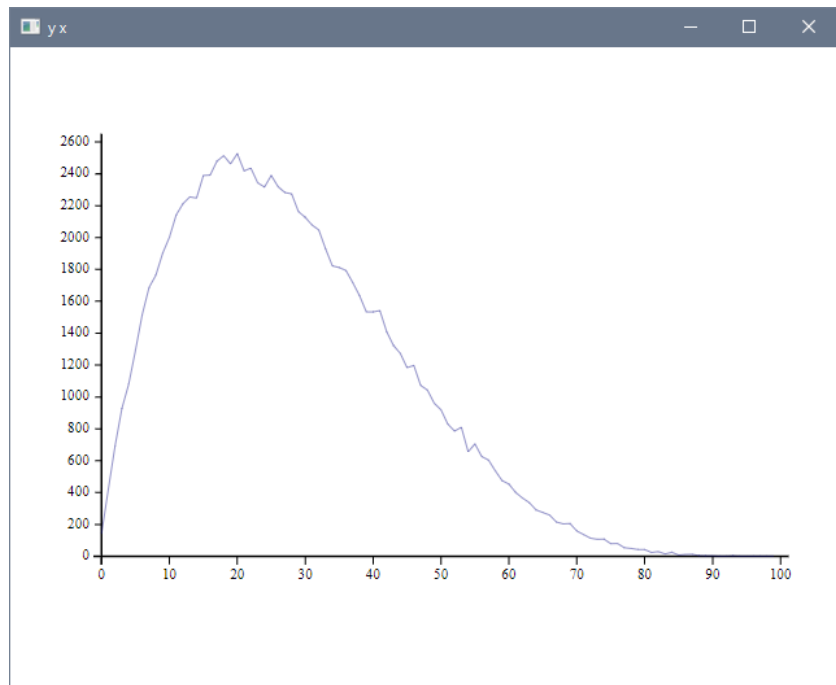
Group	X	Meaning	Default
Identity	0	Name	' '
Syntax	10	Function result (0: none or not a function, 1: explicit, ~1: shy)	0
	11	Function valence (0: niladic, 1: monadic, 2: dyadic, ~2: ambivalent)	0
	12	Operator valence: (0: not an operator, 1: monadic, 2: dyadic)	0
Last edit	20	Author of last edit	' '
	21	Number of days passed between 1899-12-31 at 00:00 UTC and last edit (includes fractional days)	0
	22	Local timestamp at last edit (format is the 7-item vector described by DTS . See Timestamp .)	9
	23	Number of bytes required for storage without sharing	0
Restrictions	30	Source can be displayed	~1
	31	Execution can be suspended mid-execution	~1
	32	Responds to weak interrupt	~1
Class*	40	Syntactic supra-class (~1: invalid name, 0: undefined, 1: label, 2: variable, 3: function, 4: operator, 8: event, 9: object)	~1
	41	Syntactic sub-class (0: none, 1: traditional/plain, 2: field/dynamic/instance, 3: property/derived/primitive, 4: class, 5: interface, 6: external, 7: external interface)	0
	42	Full syntactic class (sum of supra- and sub-class)	~1
Source	50	File name	' '
	51	File encoding	' '
	52	File checksum	' '
	53	File line separators (13: Carriage Return, 10: Line Feed, 13 10: Carriage Return followed by Line Feed, 133: New Line, 11: Vertical Tab, 12: Form Feed, 8232: Line Separator, 8233: Paragraph Separator)	9
	54	Definition's offset from top	0
	55	Number of lines in definition	0
Definition	60	Verbatim source (as typed)	0p<' '
	61	Normalised source (with AUTOFORMAT=1 and TAB_STOPS=4)	0p<' '
	62	Most precise available source (verbatim with fallback to normalised)	0p<' '





Developer Productivity:

Generating random numbers



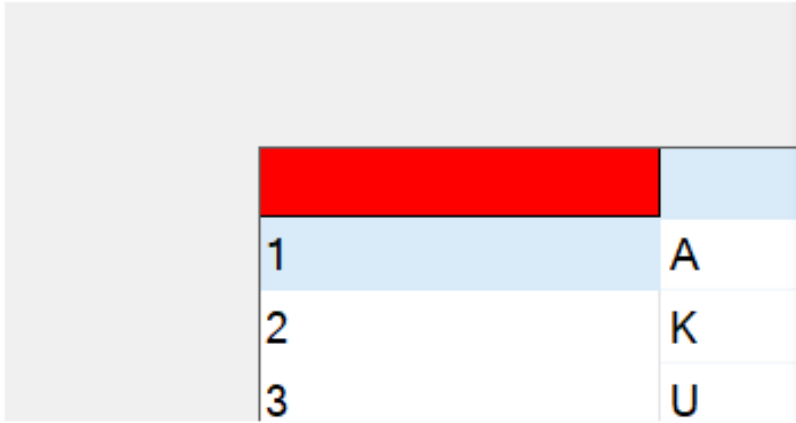
```
Dist ← 16808I  
val ← 2 5 Dist 'Beta' 1e5  
res ← 100  
bin ← [res×val  
x ← -1+ires  
y ← {≠ω}⊖x,bin  
]plot y x
```



New Win32 GUI (WIC) features!!!

- CornerTitleBCol on Grid
- FireOnce on Timer

f.g.CornerTitleBCol+<255 0 0



1	A
2	K
3	U



Developer Productivity:

New user commands



Welcome to Dyalog Videos Page

dialogs.tv/Webinar?v=A6cNLh52Bkl

Apps

Link

APL

Flying & Sailing

Car

Dyalog

Cloud

SBO

Travel

Linux

Sport

Productivity

Git

Covid

DYALOG

DYALOG

WEBINAR

Webinar

APL Seeds '22

Dyalog '21

APL Seeds '21

Dyalog '20

Dyalog '19

Dyalog '18

Dyalog '17

Dyalog '16

Dyalog '15

Dyalog '14

Dyalog '13

Dyalog '12

Dyalog '11

APL Berlin 2010

Dyalog '09

Dyalog '08

New User Commands in Dyalog v18.2 – Dyalog Webinar with Rich Park

Overview

]

OUTPUT.View

]

WS.Names

]

TOOLS.APLCart

]

OUTPUT.Repr

]

EXPERIMENTAL.Get

MORE VIDEOS

1

0:18 / 29:44

New User Commands in Dyalog v18.2

Watch Later

Share

Youtube

Coming up

New User Commands in Dyalog v18.2

Richard Park

Previous Webinars

Introducing Dyalog version 18.2

Morten Kromberg

Data Visualisation

Richard Park

Computing Check Digits, Fast

Adam Brudzewsky

Error Handling - Part 5

Adam Brudzewsky

TryAPL v3

Richard Park

38

Service Orientation

A rapidly increasing share of new APL code (and some old code) is exposed as services

- **Jarvis** wraps APL code as HTTP/JSON or RESTful services on any platform
 - <https://github.com/dyalog/jarvis>
- Version 18.2 is easier to deploy and debug as a “headless” service
 - Allows improved docker containers

RESTful API
GET PUT POST DELETE

HTTP(s) / JSON



Summary – Version 18.2

- ✧ Windows Patches
- ✧ Basic Licence
- ✧ Launching from Text Files:
 - ✧ Shell Scripts
 - ✧ Text Source
 - ✧ LOAD Parameter
 - ✧ Windows Explorer Integration
- ✧ Improved Docker Support
- ✧ RIDE v4.4
- ✧ Developer Productivity:
 - ✧ JSON tables/datasets
 - ✧ □ATX metadata one-stop-shop
 - ✧ Generating random numbers
 - ✧ New user commands



Version 19.0 and Beyond

Everything mentioned in the following is a statement of intent...

- Version 19.0 release target = 1 year after 18.2, or March 2023



Planned Features of v19.0

- .NET 6 bridge
- I/O re-implementation
- Improve Installers
- Keyboarding on all platforms
- Arm64 (M1 Mac & Pi)
- Monitoring / Health Checks
- Optimisation work
- Portable File Functions
- HTMLRenderer improvements
- Session initialisation
- Configuration



.NET 6 Bridge

- Export APL code as .NET assemblies
- Create executables under
 - Linux: Amd/Intel x64 and Pi on Arm64
 - macOS (x64 and M1)
 - Windows (x64 – maybe Arm64)
 - (maybe Android on Arm64 soon)



V19.0 should also support .NET 3.1, 5 and 7



.NET 6: Asynchronous Features

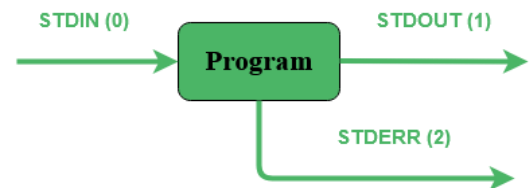
- Asynchronous function calls have appeared in many languages
 - C# / .NET use is widespread
- V19.0 work on the .NET bridge will hopefully include Async support
- "futures" may be a natural solution
 - Already used by isolates for asynch processing



I/O re-implementation

(completion of support for shell scripts)

- Allow control over whether I/O streams are connected to RIDE or redirected files
 - (allows RIDE debugging of APL shell scripts)



aplscript.apls <data



Installation Issues

- ✧ Linux-style installation for macOS
 - ✧ (most Mac APL users are really using the Mac as a very cool UNIX workstation)
- ✧ Remove "termcap" dependency
 - ✧ (Varies from one Linux to the next)
- ✧ Make HTMLRenderer easily upgrade-able
 - ✧ (new version of Chromium without reinstalling APL)



Keyboarding on all platforms

Quite simply: We need to review how APL symbols are entered into APL - and other applications

- ✧ The way Windows handles keyboards is changing
 - IME no longer works with all Windows 10 apps
- ✧ "ctrl" as the APL key is more and more problematic
- ✧ We need to provide more easily configurable options, and much better documentation



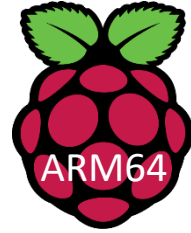
Arm64 (M1 Mac & Pi)

64-bit ARM chips are appearing in important places that Dyalog should support:

- New "M1" Macs
- Raspberry Pi – 64 Bit

We will also look at, but are unlikely to add official support in v19.0

- ARM based Windows
- Android



Health Checks

Tools to monitor the state of APL processes

- ✧ CPU consumption
- ✧ Memory usage, Compaction counts, etc
- ✧ Are any threads suspended?
- ✧)SI and Error information
- ✧ Available via API and or protocols like SNMP

Important as APL becomes more widely used to implement multi-process services



Optimisation work

- Identify and carefully implement most valuable (disabled) v18.0 optimisations
- Continue work on adding new idioms, special cases of rank and other operators
- (resume steady performance work, following new process)
- **PLEASE** send us your application benchmarks!



[Portable] File Functions

- `NCOPY` to allow a regular callback, so you can cancel long-running ops
- `NATTRIBUTES` to set e.g. timestamps, read-only/hidden and other attributes
- `NINFO` recursion depth option



HTMLRenderer improvements

- Find a way to easily upgrade the version of the Chromium Embedded Framework
 - Possibly turn the HTMLRenderer into an Open Source project
- Multiple Modal Instances
 - (only allow access to one window if required)
- Thread switching in OnHTTPReceive and OnWebSocketReceive to allow threaded GUI apps



Session initialisation mechanism

- Re-implement the way that the session object (`⌈SE`) is populated at startup
- Completely separate Dyalog content from user content with no risk of conflict
 - (and no need to reinstall own content when APL is upgraded)
- Design a declarative way to add user-defined menu[item]s and widgets
- User Command implementation will also be reviewed



Configuration

Move towards portable configuration

- Remove need for a startup script under 'NIXes
 - Good defaults for all settings
 - All settings configurable via text files
- Remove need for the Windows Registry
 - (Windows APL already has good defaults)
 - Move all "interpreter settings" to config files
 - Leave IDE settings in the Registry (as RIDE settings are in a JSON repository)

This might not make 19.0, but remains an important short term goal.



Recap - Planned Features of v19.0

- .NET 6 bridge
- I/O re-implementation
- Improve Installers
- Keyboarding on all platforms
- Arm64 (M1 Mac & Pi)
- Monitoring / Health Checks
- Optimisation work
- Portable File Functions
- HTMLRenderer improvements
- Session initialisation
- Configuration



Improving "Cross Platform" capability & similarity

- .NET 6 bridge
- I/O re-implementation
- Improve Installers
- Keyboarding on all platforms
- Arm64 (M1 Mac & Pi)
- Monitoring / Health Checks
- Optimisation work
- Portable File Functions
- HTMLRenderer improvements
- Session initialisation
- Configuration



Beyond 19.0

- ✧ Array Notation
- ✧ Multiple numeric towers
- ✧ Under $\ddot{\vee}$
- ✧ Simple cross-platform user interfaces



Array Notation

```
[ 'Jan'           'Feb'           'Mar'
  (101 102 103 104) (201 202 203 204) (301 302 303 304) ]
```



Final Community
review of
spec before
implementation

Under $\ddot{\vee}$

Computational Under: Mathematical definition

$$\alpha \text{ f } \ddot{\vee} \text{ g } \omega \quad \leftrightarrow \quad (\text{g } \ddot{*}^{-1}) (\text{g } \alpha) \text{ f } (\text{g } \omega)$$

$$\begin{array}{ccccccc} & 2 & 3 & 4 & +\ddot{\vee} & \otimes & 2 \\ 4 & 6 & 8 & & & & \end{array}$$

Structural Under: Less strict, but "more useful"

```
1 ⌈C $\ddot{\vee}$ (>`) 'adam' 'karta' 'nathaniel'
Adam  Karta  Nathaniel
```

Also to come...

Obverse: $\text{FFT}\ddot{\vee}\text{IFT}$
(declared inverse)

Depth: $\text{f}\ddot{\text{o}}\text{n}$
(f on depth-n sub-arrays)



Enhancing APL: Multiple Towers

Goals:

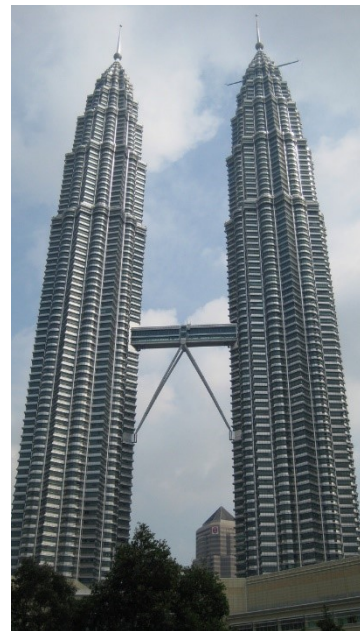
- Better control of DECIMAL data type
- Accommodate INT64 and RATIONAL types

Model: Three Towers, selectable **per array**

- Fast: BOOL, INT 8-16-32, DOUBLE, COMPLEX
- Decimal: BOOL, INT 8-16-32-**64**, DECF
- Unlimited: **RATIONAL**

Significant design remains to be done on notation, rules for comparison and promotion. What is:

$(\cup 0.1D, (1 \div 10), 1R10)$

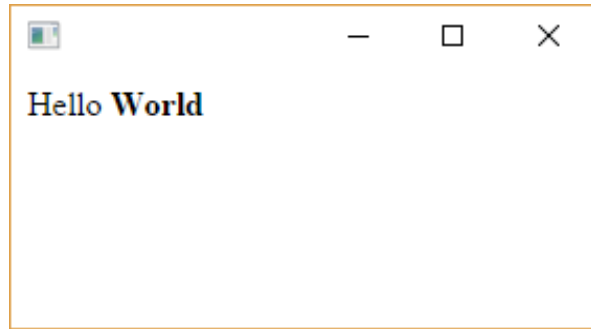


Quest for the Holy Grail



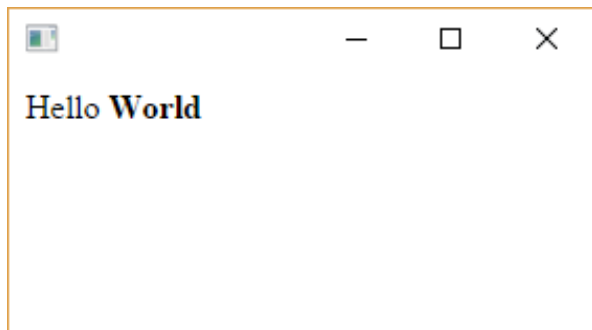
Simple Cross-Platform User Interfaces

```
'hr' WC 'HTMLRenderer' 'Hello <b>World</b>'
```



Simple Cross-Platform User Interfaces

```
'hr' WC 'HTMLRenderer' 'Hello <b>World</b>'
```



- The challenge is generating the HTML/JS...
 - Our "DUI" has not yet caught on
- Perhaps the community can help?



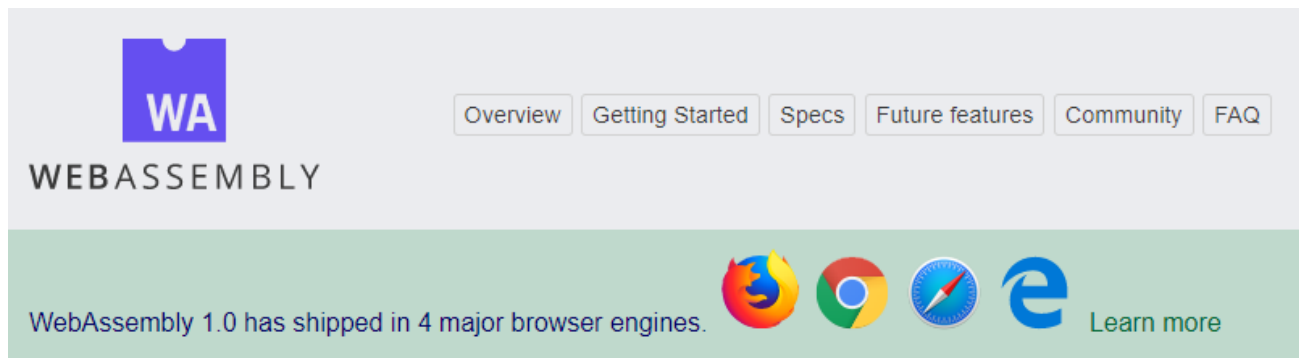
GitHub: the-carlisle-group / Abacus

- Project to generate HTML from simple APL code

```
Run←{  
  □THIS.H←##.Main  
  p←□NS''  
  p.Caption←'Sample Form'  
  p.Title←'A Sample Dialog Box'  
  p.Subtitle←'This is a resizable, modal dialog box.'  
  p.OnOK←'OnOK'  
  p.OnCancel←'OnCancel'  
  p.Content←##.Grid.New GridDataLarge 0  
  ##.DialogBox.Run p  
}
```



Web Assembly (WASM)



We could compile
Dyalog APL to run
in the browser.

(some features
would need to be
disabled)

WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.



Conclusion

- ✧ Although improving documentation, samples and training materials is at least as important
- ✧ ... and Packages ...
- ✧ ... we are not about to run out of ideas for ways to improve Dyalog APL and it's tooling

