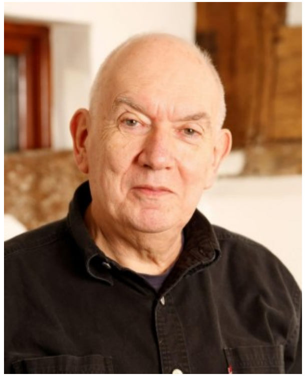
SWEDAPL

Stockholm, April 17th 2024

News from Dyalog

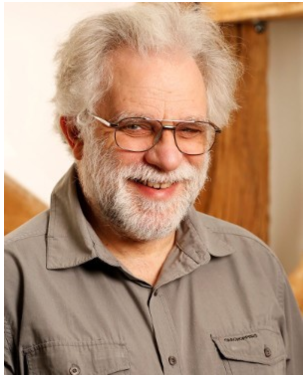
Morten Kromberg, CTO

We Stand on the Shoulders of Giants



John Scholes (1948-2019)

Roger Hui (1953-2021)

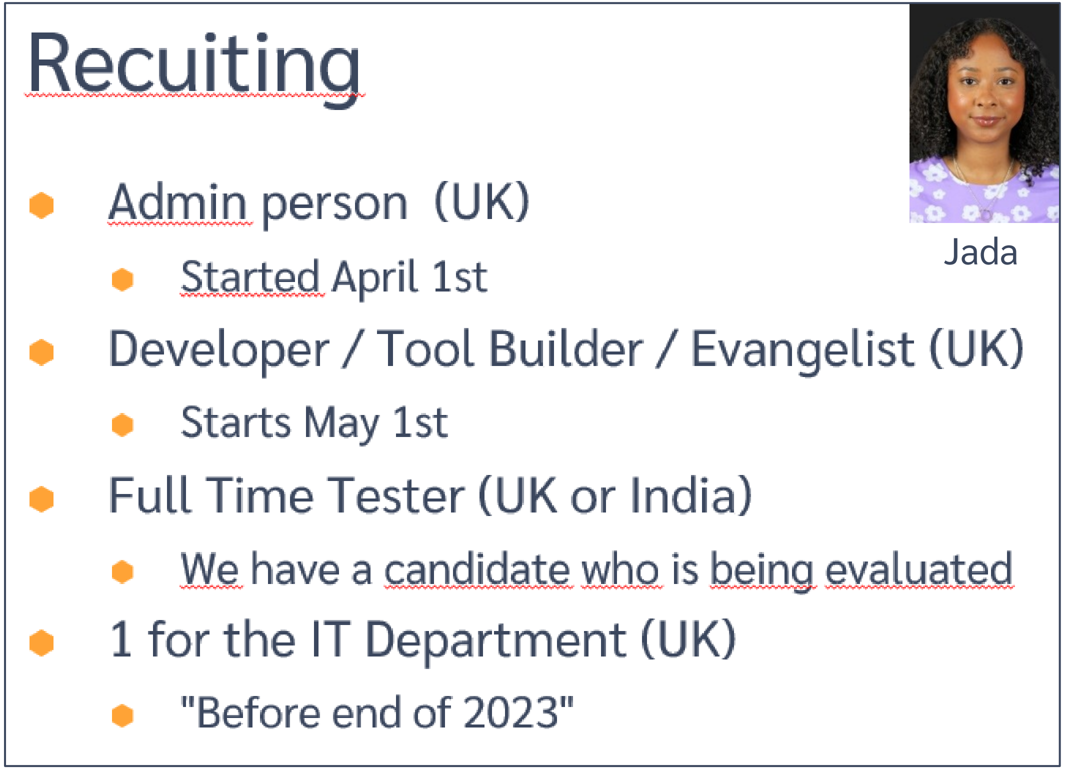


Geoff Streeter (retired 2023)

https://rogerhui.rip
https://johnscholes.rip

Ken Iverson's Blackboard arrives in Bramley (2010)

From my SWEDAPL'23 Presentation



Recuiting

- Admin person (UK)
  - Started April 1st
- Developer / Tool Builder / Evangelist (UK)
  - Starts May 1st
- Full Time Tester (UK or India)
  - We have a candidate who is being evaluated
- 1 for the IT Department (UK)
  - "Before end of 2023"

Jada

Stefan

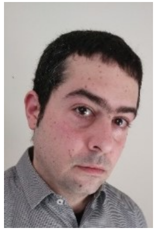
Aarush
Bangalore

Abs

Dyalog – The Next Generation

2010-2021

2022



Mike

2023

Karl

2024

2023 Summer Interns

Our 2nd Employee to Retire

Gitte Christensen

The CEO has retired!

Kirstine Kromberg

Long Live the New CEO!

Under Gitte's Reign ...

- Headcount and Turnover ~~Quadrupled~~ Quintupled
  - Headcount now ~26 full time equivalents
  - Steadily increasing R&D budget
  - First generation of interpreter developers retired; Second productive; Third recruited
  - Largest sustained investment in APL technology in history

Under Gitte's Reign...

- Trading Profit in every year since 2005. Sources:
    - Growing partnerships with major customers
    - Increasing share of APL market ("unfortunately" accelerating)
    - A small number of completely new APL systems (so far)
- Manoevering towards re-launching APL as a modern tool for prototyping and rapid application development

Financial Status

Owners	2008	2018	Present
SimCorp (Denmark)	32.33%	40%	24%
APL Italiana (Italy)	32.33%		
Management & Employees	35.33%	60%	76%

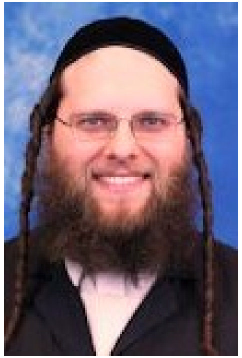
- Ownership
  - 24% owned by SimCorp (which acquired APL Italiana, and was then itself acquired by Deutsche Börse)
  - 76% owned by management and employees
- Revenue steadily increasing:

5-year period	Growth
2017-2023	35%
2012-2017	8%
2007-2012	57%

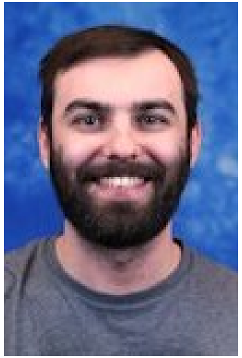
- Profitable every year since acquisition in 2005

News from the Evangelism Front

Aaron Hsu
co-dfns
compiler



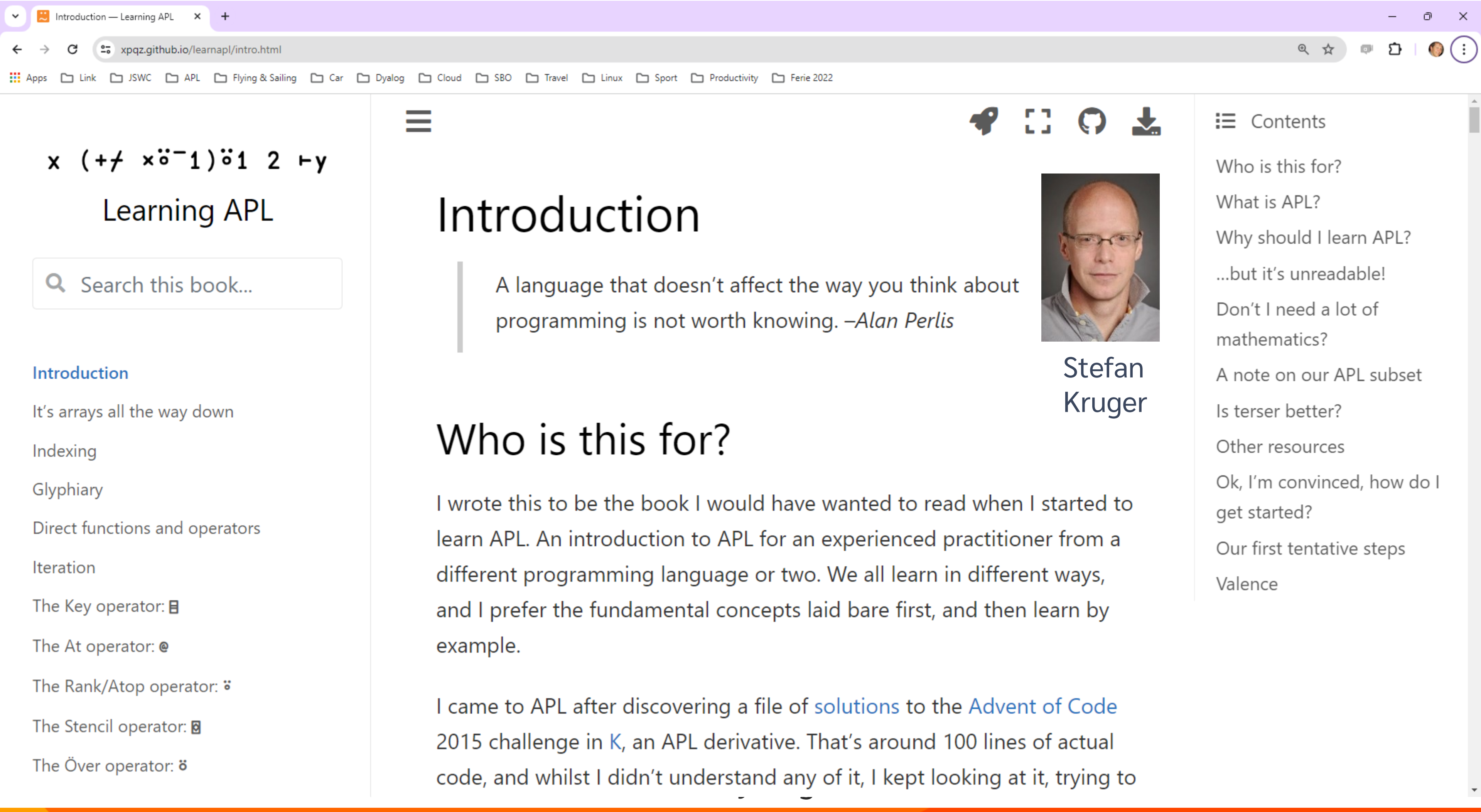
Adam Brudzewsky
Here, There and
Everywhere



Rich Park
Media, Training
and "Outreach"

Stefan Kruger
Lead Technical
Writer

Jesús López
Metallurgist
(embedded
evangelist)



x (+⌿ ×⍤¯1)⍤1 2 ⊢y

Learning APL

Search this book...

Introduction

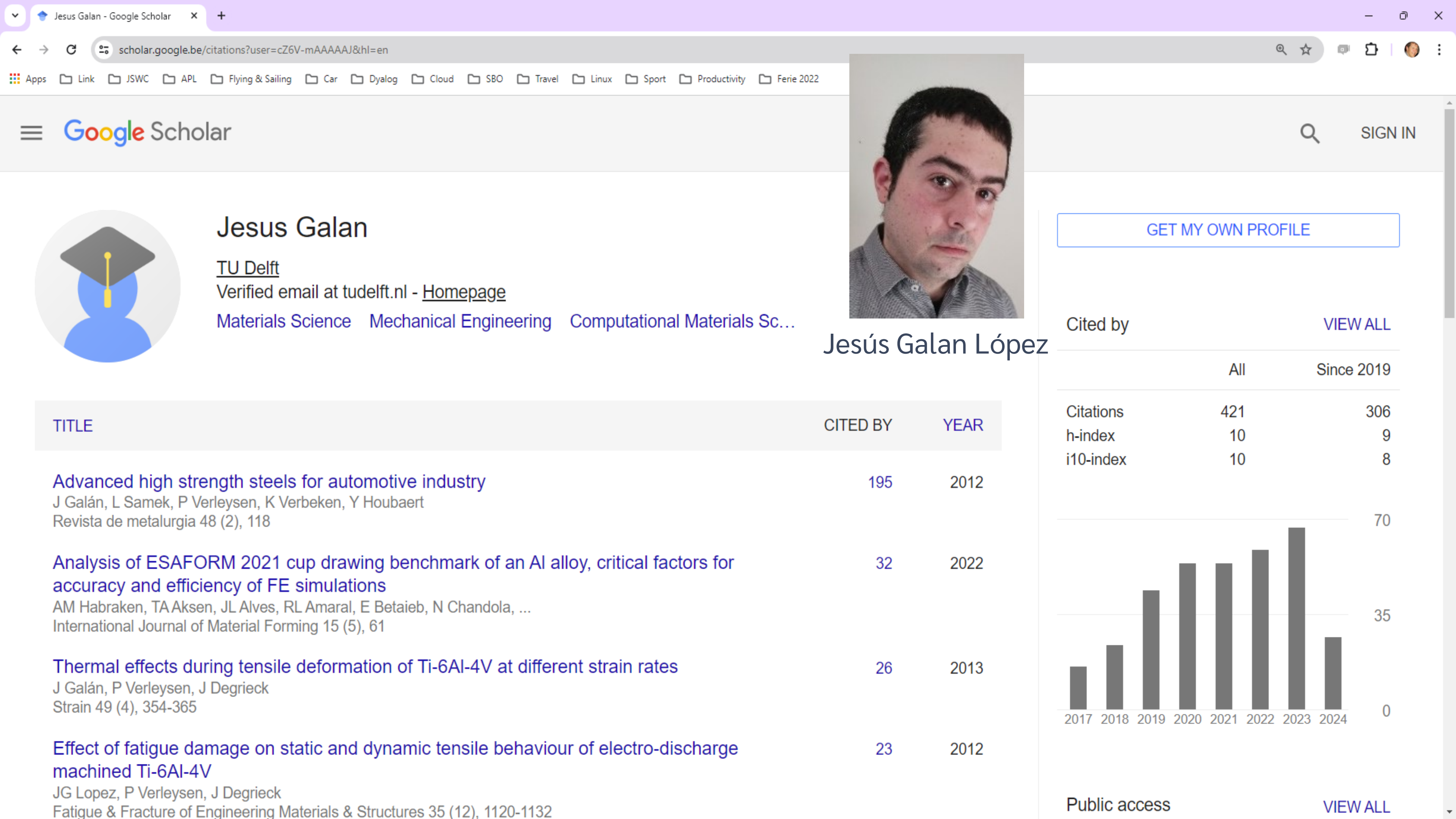
> A language that doesn't affect the way you think about programming is not worth knowing. –*Alan Perlis*

Stefan Kruger

Who is this for?

I wrote this to be the book I would have wanted to read when I started to learn APL. An introduction to APL for an experienced practitioner from a different programming language or two. We all learn in different ways, and I prefer the fundamental concepts laid bare first, and then learn by example.

I came to APL after discovering a file of solutions to the Advent of Code 2015 challenge in K, an APL derivative. That's around 100 lines of actual code, and whilst I didn't understand any of it, I kept looking at it, trying to



Jesús Galan López

Jesus Galan

TU Delft

Verified email at tudelft.nl - Homepage

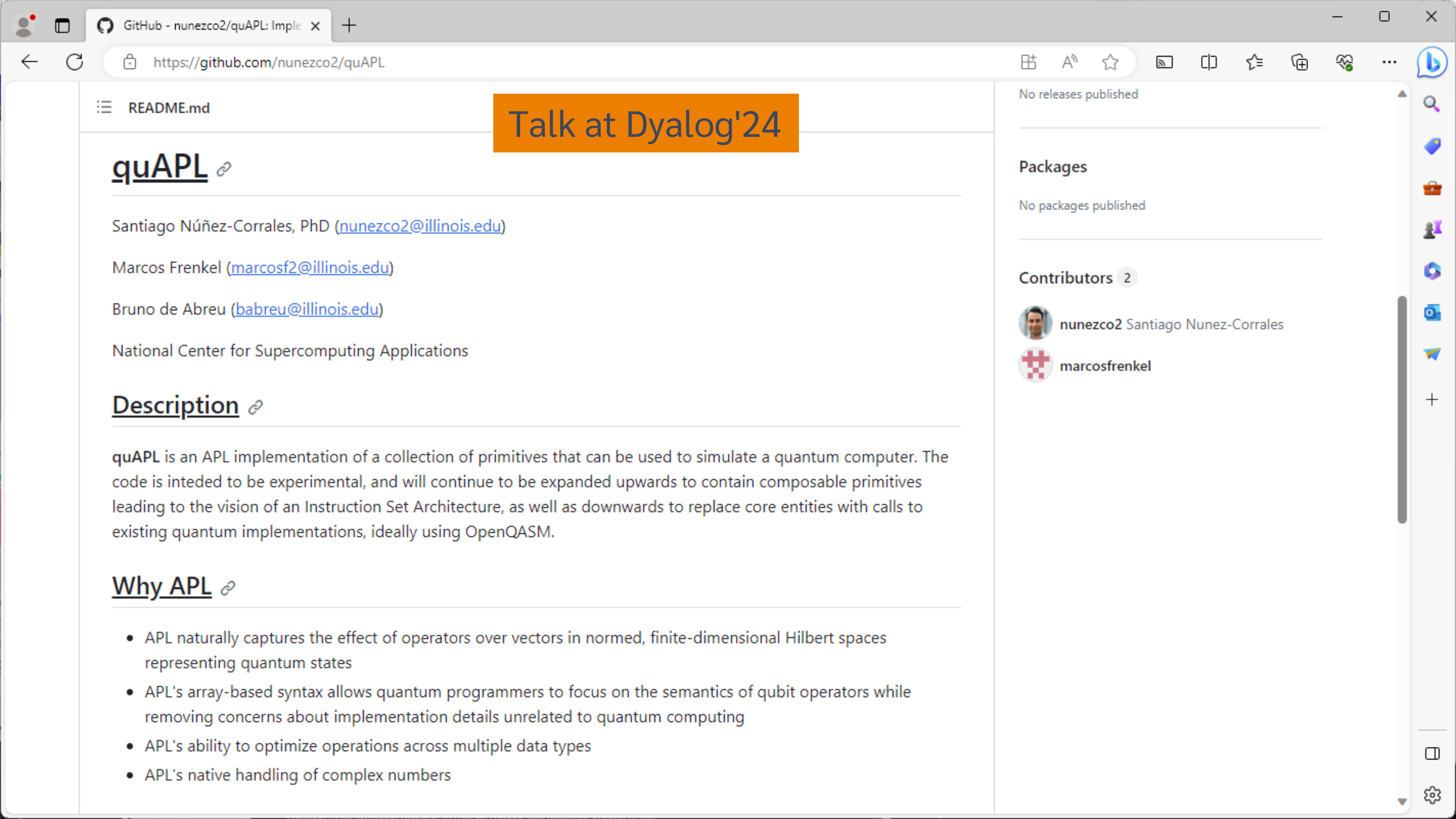
Materials Science Mechanical Engineering Computational Materials Sc…

TITLE	CITED BY	YEAR
Advanced high strength steels for automotive industry J Galán, L Samek, P Verleysen, K Verbeken, Y Houbaert Revista de metalurgia 48 (2), 118	195	2012
Analysis of ESAFORM 2021 cup drawing benchmark of an Al alloy, critical factors for accuracy and efficiency of FE simulations AM Habraken, TA Aksen, JL Alves, RL Amaral, E Betaieb, N Chandola, ... International Journal of Material Forming 15 (5), 61	32	2022
Thermal effects during tensile deformation of Ti-6Al-4V at different strain rates J Galán, P Verleysen, J Degrieck Strain 49 (4), 354-365	26	2013
Effect of fatigue damage on static and dynamic tensile behaviour of electro-discharge machined Ti-6Al-4V JG Lopez, P Verleysen, J Degrieck Fatigue & Fracture of Engineering Materials & Structures 35 (12), 1120-1132	23	2012

Cited by VIEW ALL

	All	Since 2019
Citations	421	306
h-index	10	9
i10-index	10	8

Public access VIEW ALL


GitHub - nunezco2/quAPL: Imple
https://github.com/nunezco2/quAPL
README.md

Talk at Dyalog'24

quAPL

Santiago Núñez-Corrales, PhD (nunezco2@illinois.edu)

Marcos Frenkel (marcosf2@illinois.edu)

Bruno de Abreu (babreu@illinois.edu)

National Center for Supercomputing Applications

Description

quAPL is an APL implementation of a collection of primitives that can be used to simulate a quantum computer. The code is inteded to be experimental, and will continue to be expanded upwards to contain composable primitives leading to the vision of an Instruction Set Architecture, as well as downwards to replace core entities with calls to existing quantum implementations, ideally using OpenQASM.

Why APL

- APL naturally captures the effect of operators over vectors in normed, finite-dimensional Hilbert spaces representing quantum states
- APL's array-based syntax allows quantum programmers to focus on the semantics of qubit operators while removing concerns about implementation details unrelated to quantum computing
- APL's ability to optimize operations across multiple data types
- APL's native handling of complex numbers

No releases published

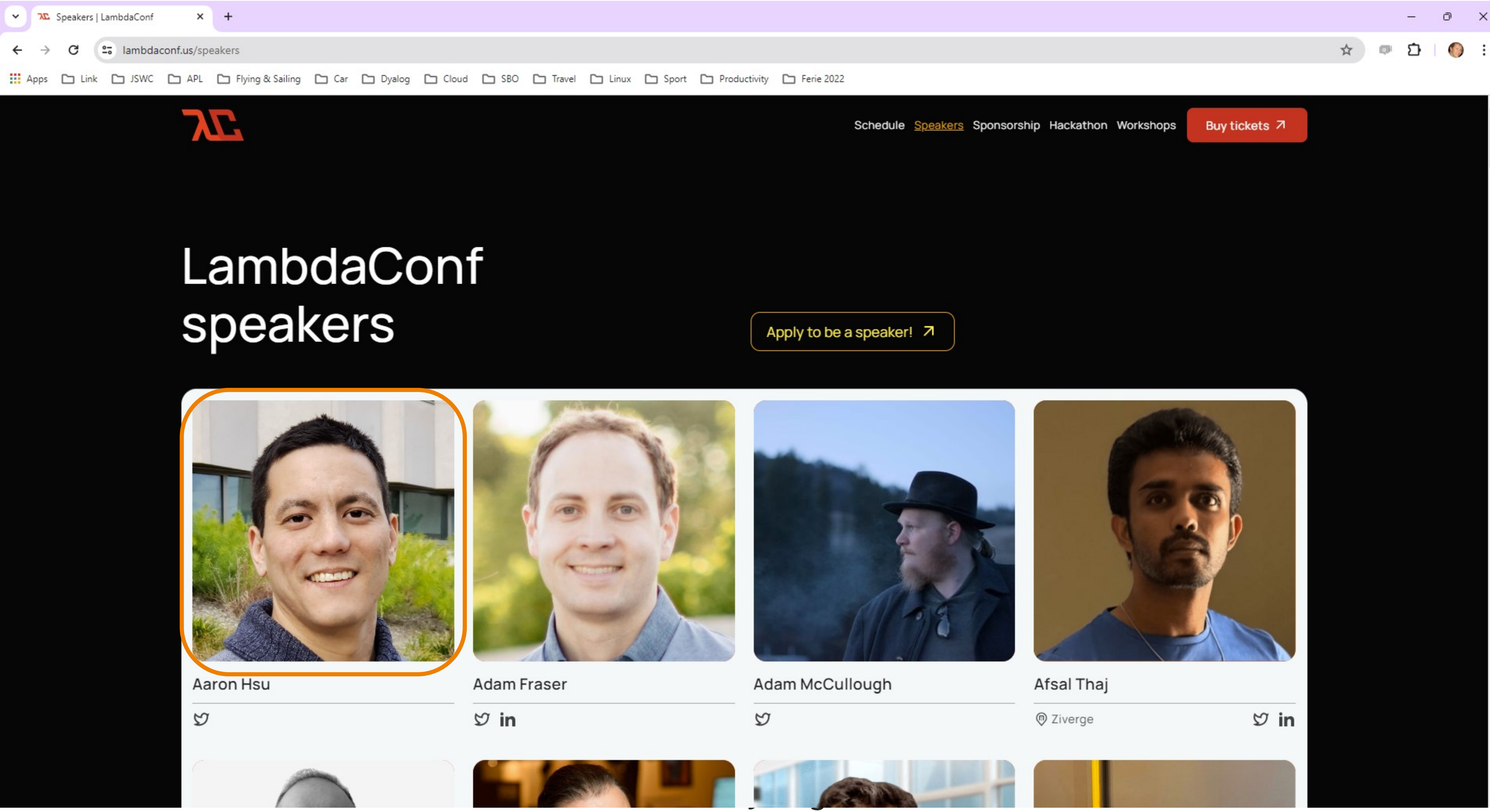
Packages

No packages published

Contributors 2

nunezco2 Santiago Nunez-Corrales

marcosfrenkel



LambdaConf speakers

Apply to be a speaker! ↗

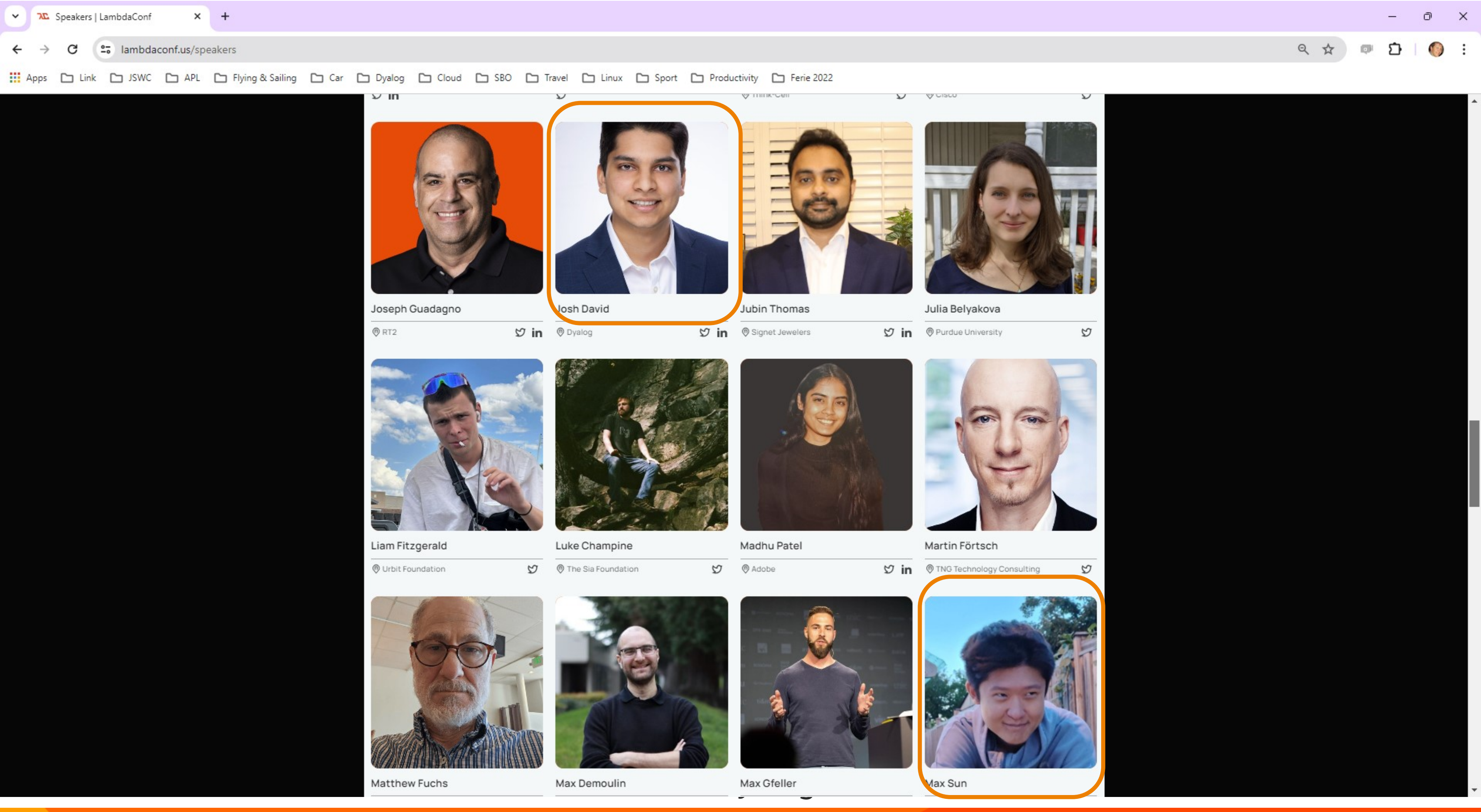
Aaron Hsu

Adam Fraser

Adam McCullough

Afsal Thaj

Ziverge



Think-Cell

Cisco

Joseph Guadagno

RT2

Josh David

Dyalog

Jubin Thomas

Signet Jewelers

Julia Belyakova

Purdue University

Liam Fitzgerald

Urbit Foundation

Luke Champine

The Sia Foundation

Madhu Patel

Adobe

Martin Förtsch

TNG Technology Consulting

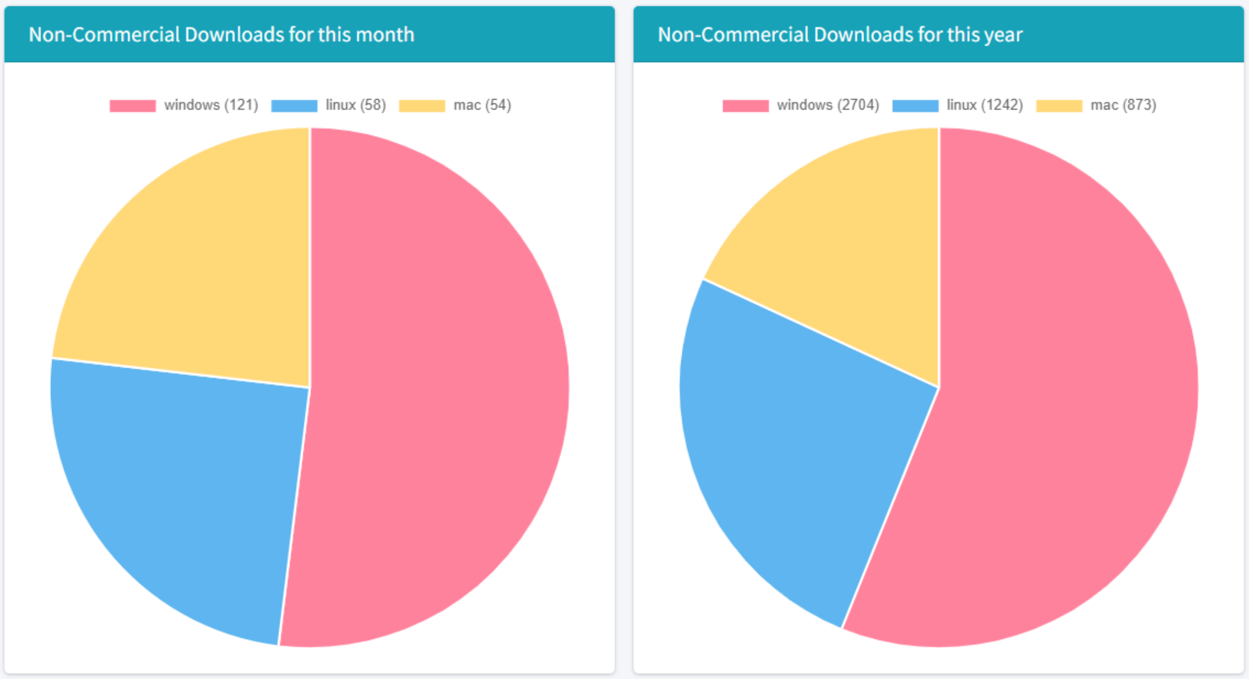
Matthew Fuchs

Max Demoulin

Max Gfeller

Max Sun

Non-Commercial Downloads (snapshot September 18th, 2023)



Non-Commercial Downloads for this month

Non-Commercial Downloads for this year

Mortens Favourite v19.0 Features

- Things that Morten has actually used since release …

Source Code Management

- **Link** 4.0 was included with v19.0. Highlights include:
  - Link a single namespace or class file
  - Default to current namespace if no namespace specified
  - Configuration files
  - Support for simple text vectors, vectors of text vectors, and character matrices in simple text files (rather than using array notation)
- The **Cider** project manager and the **Tatin** package manager will be bundled with v19.0
- Dyalog is starting to distribute tools as **Tatin** packages

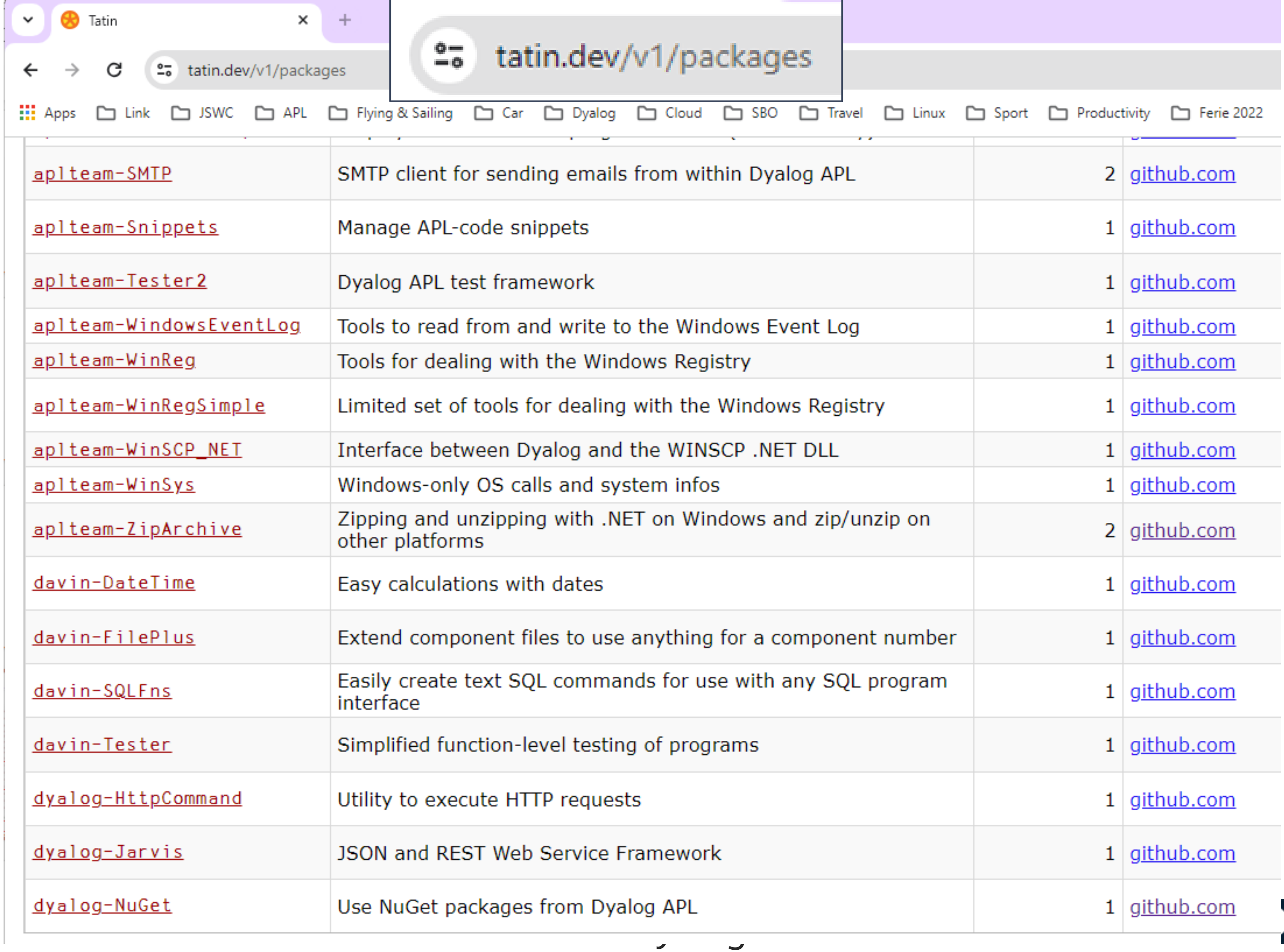
Link
(source)

Cider
(projects)

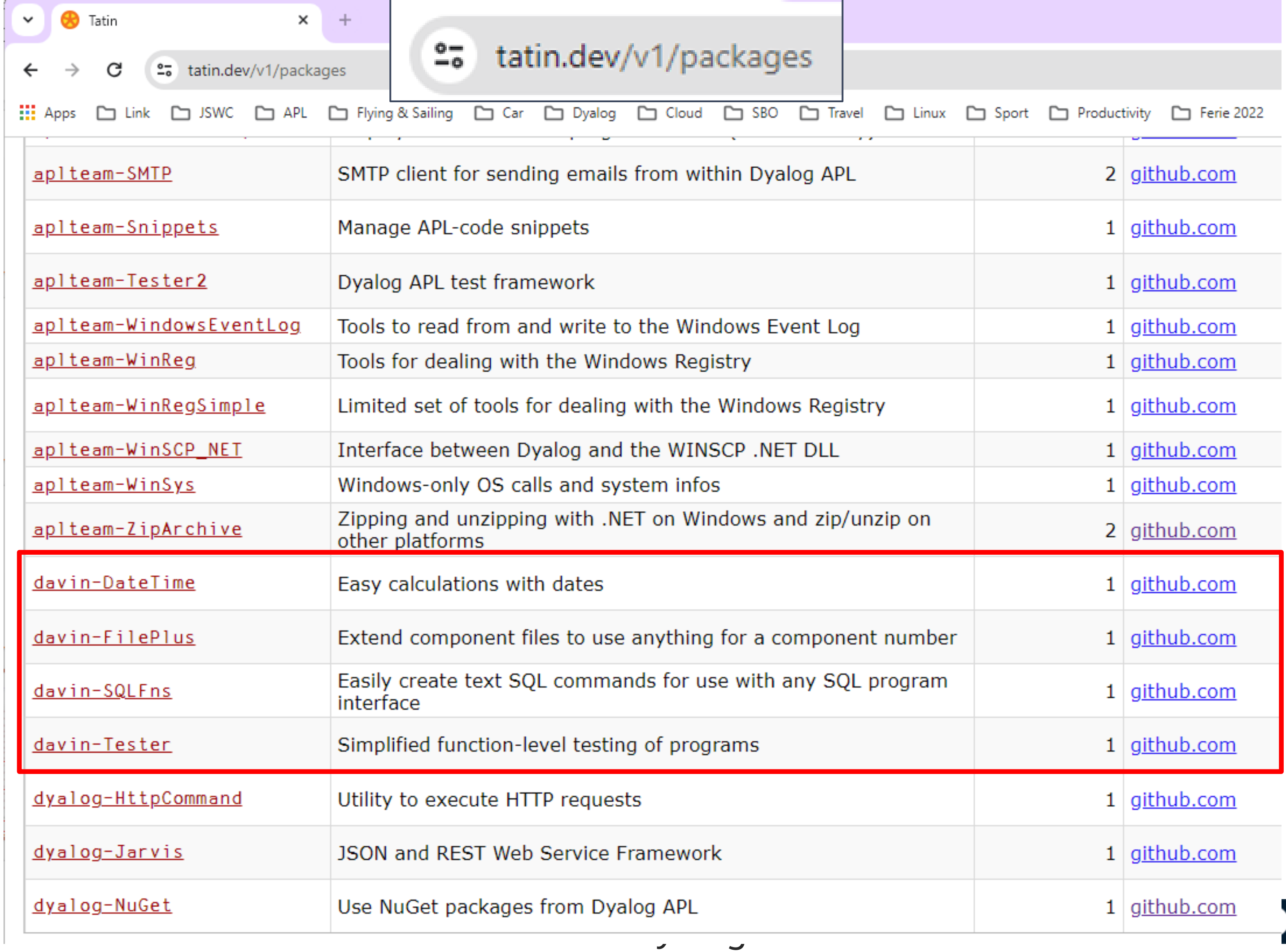
Tatin
(packages)

Tatin and Cider included...

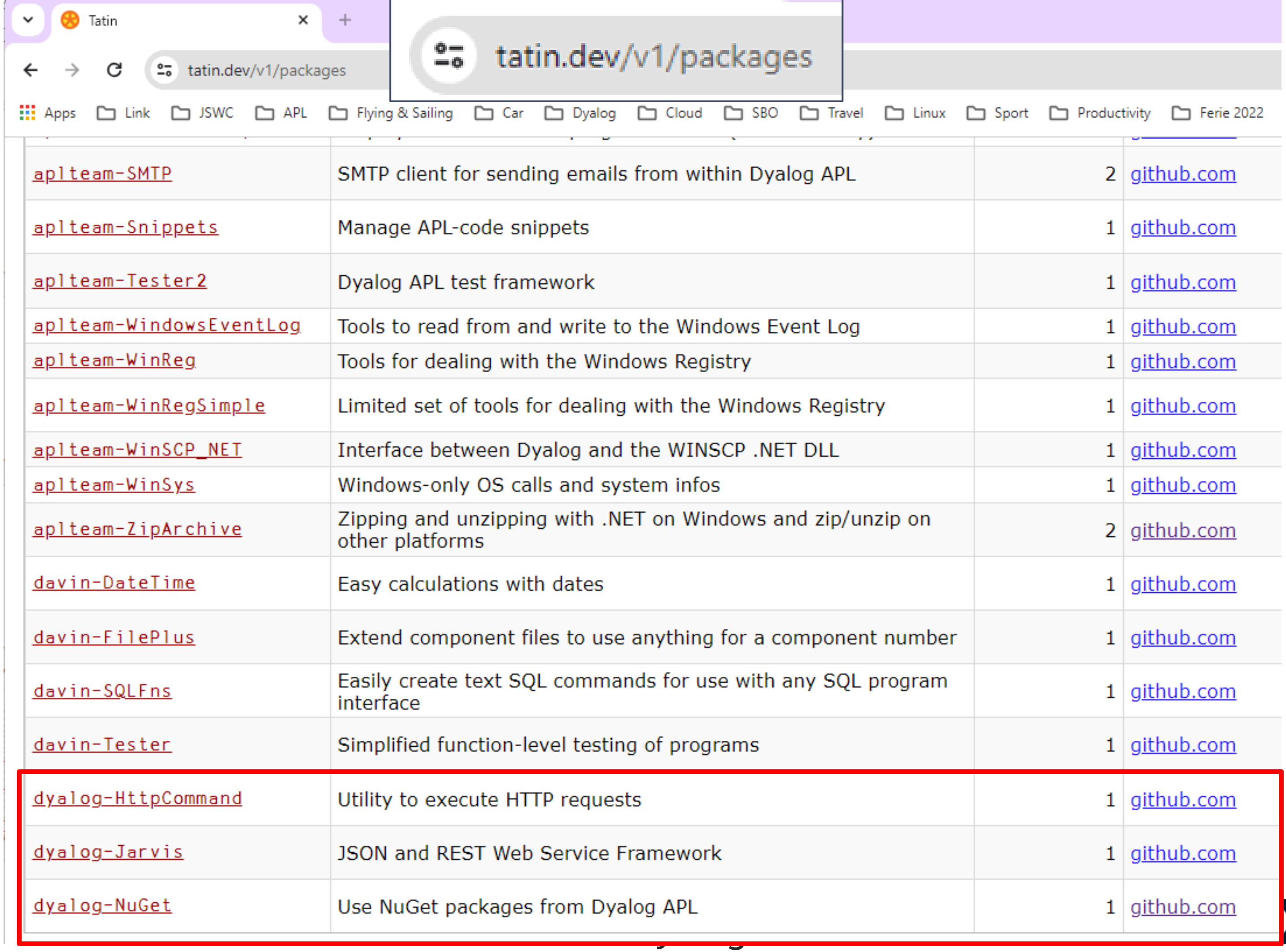
```
Dyalog APL/W Version 19.0.49074
Serial number: 000013
Tue Apr 16 13:34:28 2024

      ]tools.activate all
Rebuilding user command cache... done
Restart APL to complete activation.
cmddir set to: …;C:/…/Dyalog APL 19.0 Unicode Files/SessionExtensions/CiderTatin

      ]tatin.loadPackages dyalog-Jarvis
1 package loaded into #

      ij←⎕NEW Jarvis
      ij.⎕nl -2
AcceptFrom AllowFormData AllowGETs AppCloseFn AppInitFn AuthenticateFn …
… SessionUseCookie UseZip ValidateRequestFn WaitTimeout ZipLevel
```



aplteam-SMTP	SMTP client for sending emails from within Dyalog APL	2	github.com
aplteam-Snippets	Manage APL-code snippets	1	github.com
aplteam-Tester2	Dyalog APL test framework	1	github.com
aplteam-WindowsEventLog	Tools to read from and write to the Windows Event Log	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-WinSCP_NET	Interface between Dyalog and the WINSCP .NET DLL	1	github.com
aplteam-WinSys	Windows-only OS calls and system infos	1	github.com
aplteam-ZipArchive	Zipping and unzipping with .NET on Windows and zip/unzip on other platforms	2	github.com
davin-DateTime	Easy calculations with dates	1	github.com
davin-FilePlus	Extend component files to use anything for a component number	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
dyalog-HttpCommand	Utility to execute HTTP requests	1	github.com
dyalog-Jarvis	JSON and REST Web Service Framework	1	github.com
dyalog-NuGet	Use NuGet packages from Dyalog APL	1	github.com



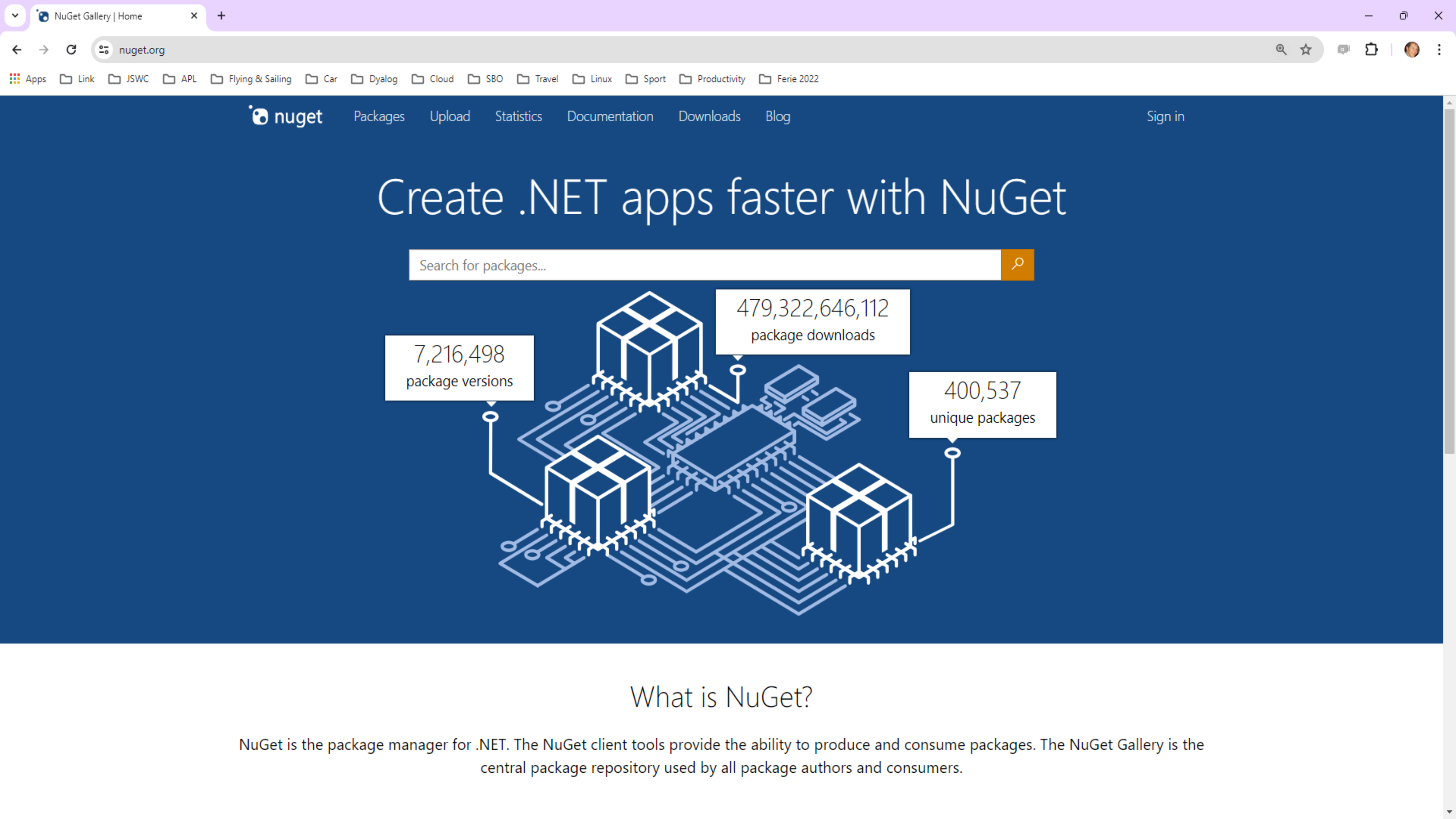
aplteam-SMTP	SMTP client for sending emails from within Dyalog APL	2	github.com
aplteam-Snippets	Manage APL-code snippets	1	github.com
aplteam-Tester2	Dyalog APL test framework	1	github.com
aplteam-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
aplteam-WinReg	Tools 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-WinSCP_NET	Interface between Dyalog and the WINSCP .NET DLL	1	github.com
aplteam-WinSys	Windows-only OS calls and system infos	1	github.com
aplteam-ZipArchive	Zipping and unzipping with .NET on Windows and zip/unzip on other platforms	2	github.com
davin-DateTime	Easy calculations with dates	1	github.com
davin-FilePlus	Extend component files to use anything for a component number	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
dyalog-HttpCommand	Utility to execute HTTP requests	1	github.com
dyalog-Jarvis	JSON and REST Web Service Framework	1	github.com
dyalog-NuGet	Use NuGet packages from Dyalog APL	1	github.com



aplteam-SMTP	SMTP client for sending emails from within Dyalog APL	2	github.com
aplteam-Snippets	Manage APL-code snippets	1	github.com
aplteam-Tester2	Dyalog APL test framework	1	github.com
aplteam-WindowsEventLog	Tools to read from and write to the Windows Event Log	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-WinSCP_NET	Interface between Dyalog and the WINSCP .NET DLL	1	github.com
aplteam-WinSys	Windows-only OS calls and system infos	1	github.com
aplteam-ZipArchive	Zipping and unzipping with .NET on Windows and zip/unzip on other platforms	2	github.com
davin-DateTime	Easy calculations with dates	1	github.com
davin-FilePlus	Extend component files to use anything for a component number	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
dyalog-HttpCommand	Utility to execute HTTP requests	1	github.com
dyalog-Jarvis	JSON and REST Web Service Framework	1	github.com
dyalog-NuGet	Use NuGet packages from Dyalog APL	1	github.com

The NuGet Tool…

NuGet is the .NET Package Manager.



Create .NET apps faster with NuGet

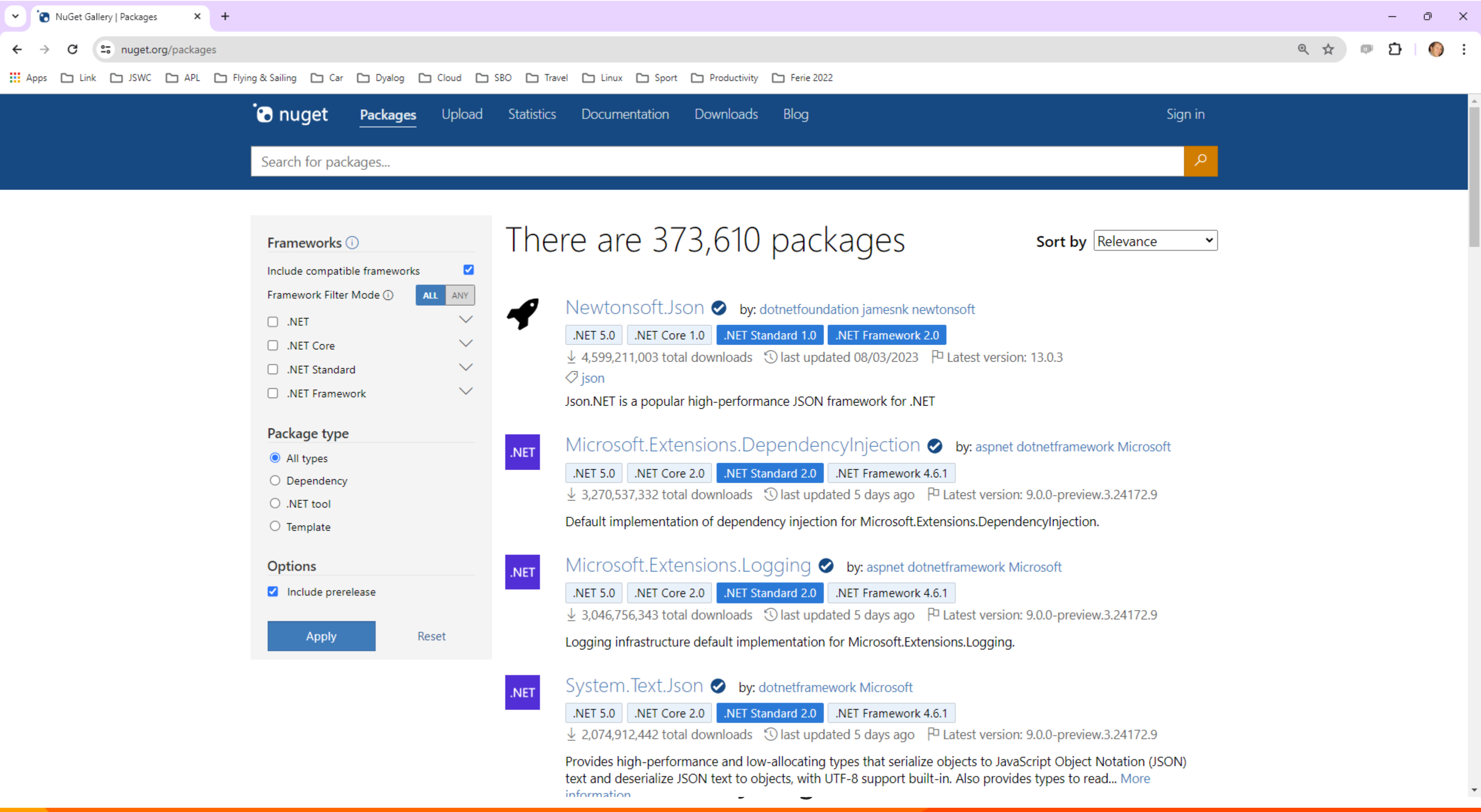
7,216,498
package versions

479,322,646,112
package downloads

400,537
unique packages

What is NuGet?

NuGet is the package manager for .NET. The NuGet client tools provide the ability to produce and consume packages. The NuGet Gallery is the central package repository used by all package authors and consumers.



There are 373,610 packages

Sort by Relevance

Frameworks

Include compatible frameworks ☑

Framework Filter Mode ALL ANY

□ .NET

□ .NET Core

□ .NET Standard

□ .NET Framework

Package type

● All types

○ Dependency

○ .NET tool

○ Template

Options

☑ Include prerelease

Apply Reset

Newtonsoft.Json ✓ by: dotnetfoundation jamesnk newtonsoft

.NET 5.0 .NET Core 1.0 .NET Standard 1.0 .NET Framework 2.0

4,599,211,003 total downloads last updated 08/03/2023 Latest version: 13.0.3

json

Json.NET is a popular high-performance JSON framework for .NET

Microsoft.Extensions.DependencyInjection ✓ by: aspnet dotnetframework Microsoft

.NET 5.0 .NET Core 2.0 .NET Standard 2.0 .NET Framework 4.6.1

3,270,537,332 total downloads last updated 5 days ago Latest version: 9.0.0-preview.3.24172.9

Default implementation of dependency injection for Microsoft.Extensions.DependencyInjection.

Microsoft.Extensions.Logging ✓ by: aspnet dotnetframework Microsoft

.NET 5.0 .NET Core 2.0 .NET Standard 2.0 .NET Framework 4.6.1

3,046,756,343 total downloads last updated 5 days ago Latest version: 9.0.0-preview.3.24172.9

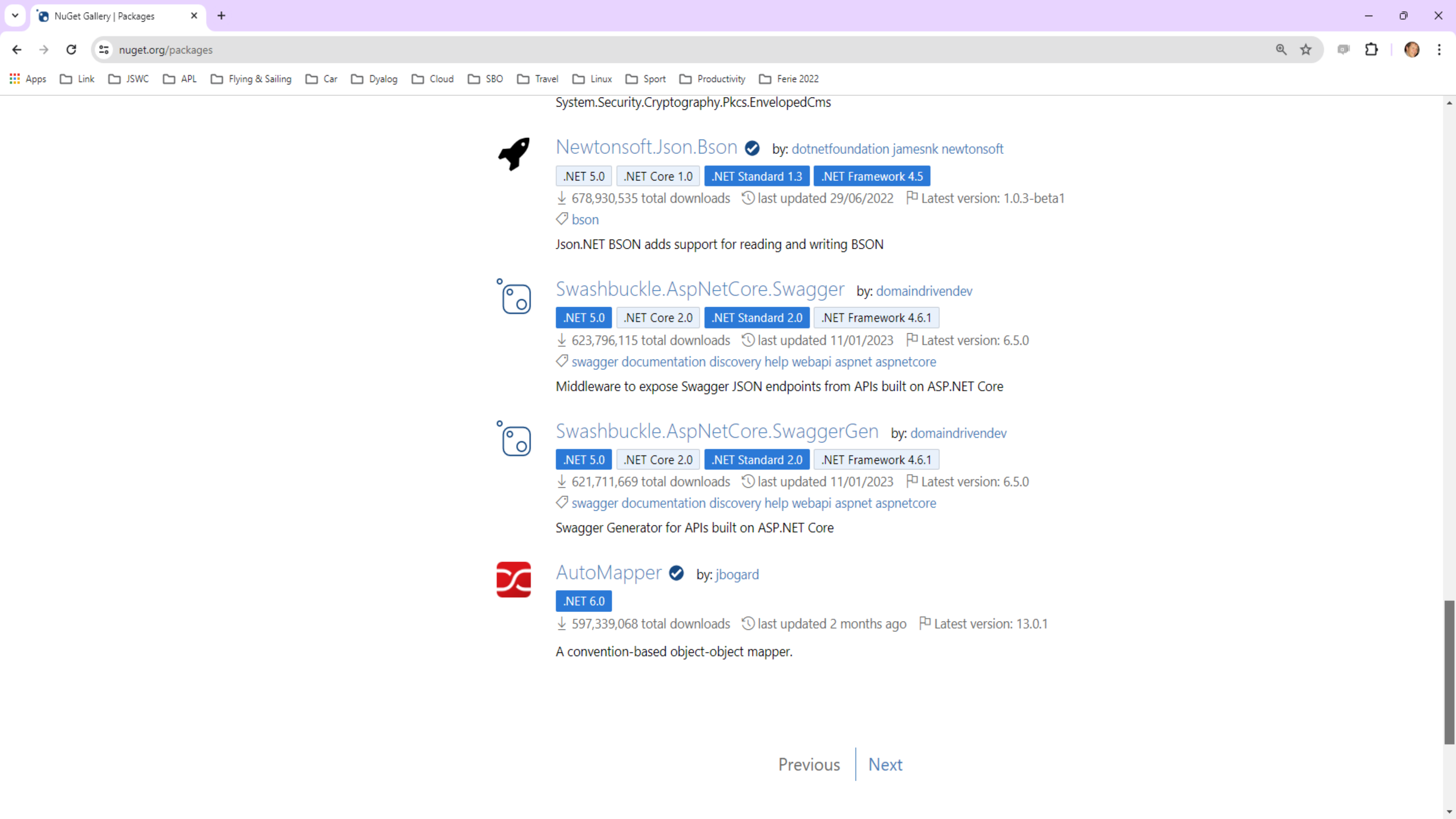
Logging infrastructure default implementation for Microsoft.Extensions.Logging.

System.Text.Json ✓ by: dotnetframework Microsoft

.NET 5.0 .NET Core 2.0 .NET Standard 2.0 .NET Framework 4.6.1

2,074,912,442 total downloads last updated 5 days ago Latest version: 9.0.0-preview.3.24172.9

Provides high-performance and low-allocating types that serialize objects to JavaScript Object Notation (JSON) text and deserialize JSON text to objects, with UTF-8 support built-in. Also provides types to read... More information



System.Security.Cryptography.Pkcs.EnvelopedCms

Newtonsoft.Json.Bson ✓ by: dotnetfoundation jamesnk newtonsoft

.NET 5.0 | .NET Core 1.0 | .NET Standard 1.3 | .NET Framework 4.5

↓ 678,930,535 total downloads last updated 29/06/2022 Latest version: 1.0.3-beta1

bson

Json.NET BSON adds support for reading and writing BSON

Swashbuckle.AspNetCore.Swagger by: domaindrivendev

.NET 5.0 | .NET Core 2.0 | .NET Standard 2.0 | .NET Framework 4.6.1

↓ 623,796,115 total downloads last updated 11/01/2023 Latest version: 6.5.0

swagger documentation discovery help webapi aspnet aspnetcore

Middleware to expose Swagger JSON endpoints from APIs built on ASP.NET Core

Swashbuckle.AspNetCore.SwaggerGen by: domaindrivendev

.NET 5.0 | .NET Core 2.0 | .NET Standard 2.0 | .NET Framework 4.6.1

↓ 621,711,669 total downloads last updated 11/01/2023 Latest version: 6.5.0

swagger documentation discovery help webapi aspnet aspnetcore

Swagger Generator for APIs built on ASP.NET Core

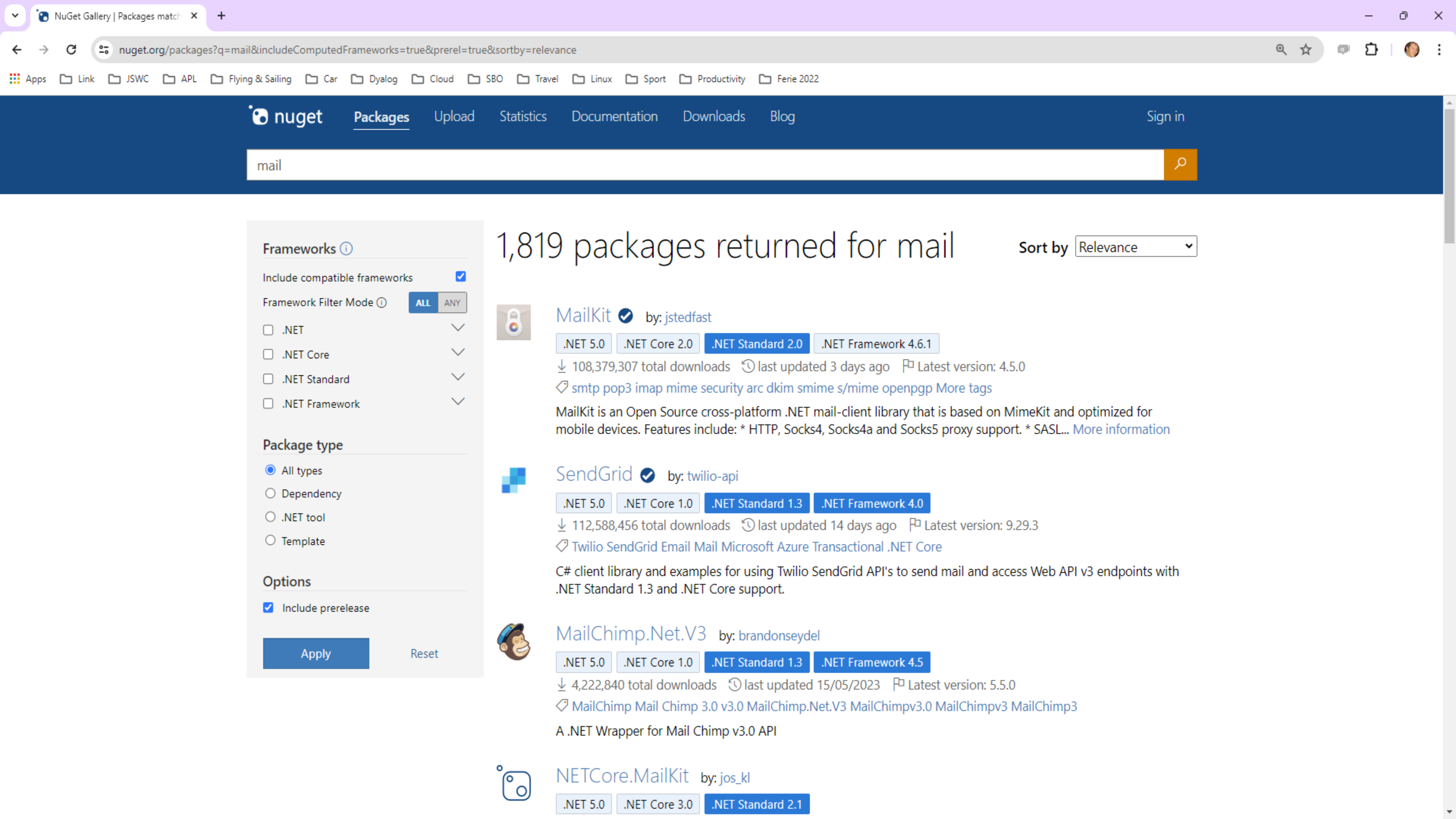
AutoMapper ✓ by: jbogard

.NET 6.0

↓ 597,339,068 total downloads last updated 2 months ago Latest version: 13.0.1

A convention-based object-object mapper.

Previous | Next



mail

Frameworks

☑ Include compatible frameworks

Framework Filter Mode [✓ ALL] ANY

□ .NET

□ .NET Core

□ .NET Standard

□ .NET Framework

Package type

● All types

○ Dependency

○ .NET tool

○ Template

Options

☑ Include prerelease

Apply   Reset

1,819 packages returned for mail

Sort by Relevance

MailKit ✓ by: jstedfast

.NET 5.0 | .NET Core 2.0 | .NET Standard 2.0 | .NET Framework 4.6.1

108,379,307 total downloads last updated 3 days ago Latest version: 4.5.0

smtp pop3 imap mime security arc dkim smime s/mime openpgp More tags

MailKit is an Open Source cross-platform .NET mail-client library that is based on MimeKit and optimized for mobile devices. Features include: * HTTP, Socks4, Socks4a and Socks5 proxy support. * SASL... More information

SendGrid ✓ by: twilio-api

.NET 5.0 | .NET Core 1.0 | .NET Standard 1.3 | .NET Framework 4.0

112,588,456 total downloads last updated 14 days ago Latest version: 9.29.3

Twilio SendGrid Email Mail Microsoft Azure Transactional .NET Core

C# client library and examples for using Twilio SendGrid API's to send mail and access Web API v3 endpoints with .NET Standard 1.3 and .NET Core support.

MailChimp.Net.V3 by: brandonseydel

.NET 5.0 | .NET Core 1.0 | .NET Standard 1.3 | .NET Framework 4.5

4,222,840 total downloads last updated 15/05/2023 Latest version: 5.5.0

MailChimp Mail Chimp 3.0 v3.0 MailChimp.Net.V3 MailChimpv3.0 MailChimpv3 MailChimp3

A .NET Wrapper for Mail Chimp v3.0 API

NETCore.MailKit by: jos_kl

.NET 5.0 | .NET Core 3.0 | .NET Standard 2.1

The NuGet Tool...

Let's try MailKit....

```
      ]tatin.loadpackages dyalog-nuget
1 package loaded into #

      ]link.import c:\devt\nuget\APLSource\Tests
Imported: #.Tests ← c:\devt\nuget\APLSource\Tests

      Tests.test 'mailkit' ⍝ Trace me
```

Cider Project Manager

- Cider uses Link to manage source code in a folder…
- Along with any packages that it uses
  - Tatin and Nuget both supported
- (I have not had cause to use Cider)

Token Allocation

JSWC needs a token for each thread doing DQ or NQ:

```
allocateTokens;z;t;i

t←'JSWC Synchronisation'

:Trap 2 ⍝ ⎕TALLOC will SYNTAX ERROR in v18.2
   z←↑⎕TALLOC 0                            ⍝ List all allocated ranges
   :If (≢z)≥i←z[;2]⍳⊂t                     ⍝ Do we already have a range?
      TOKENRANGE←z[i;1]                    ⍝ Yes: use it
   :Else
      TOKENRANGE←⎕TALLOC 1 t               ⍝ No: Allocate a range
   :EndIf
   TOKENSTEP←10*-⌈10⍟MAXSESSIONS+1         ⍝ e.g. step by 0.001 if MAXSESSIONS is 100
   (WG_TOKENBASE DQ_TOKENBASE)←TOKENRANGE+0.1 0.2 ⍝ Start at n.101 and n.2.01
:Else ⍝ Version 18.2 or earlier
   TOKENSTEP←1            ⍝ Step by 1, and separate the ranges out
   (WG_TOKENBASE DQ_TOKENBASE)←(0 1×1+MAXSESSIONS)+30⊥⎕A⍳'JSWC' ⍝ Think of a number
:EndTrap
```

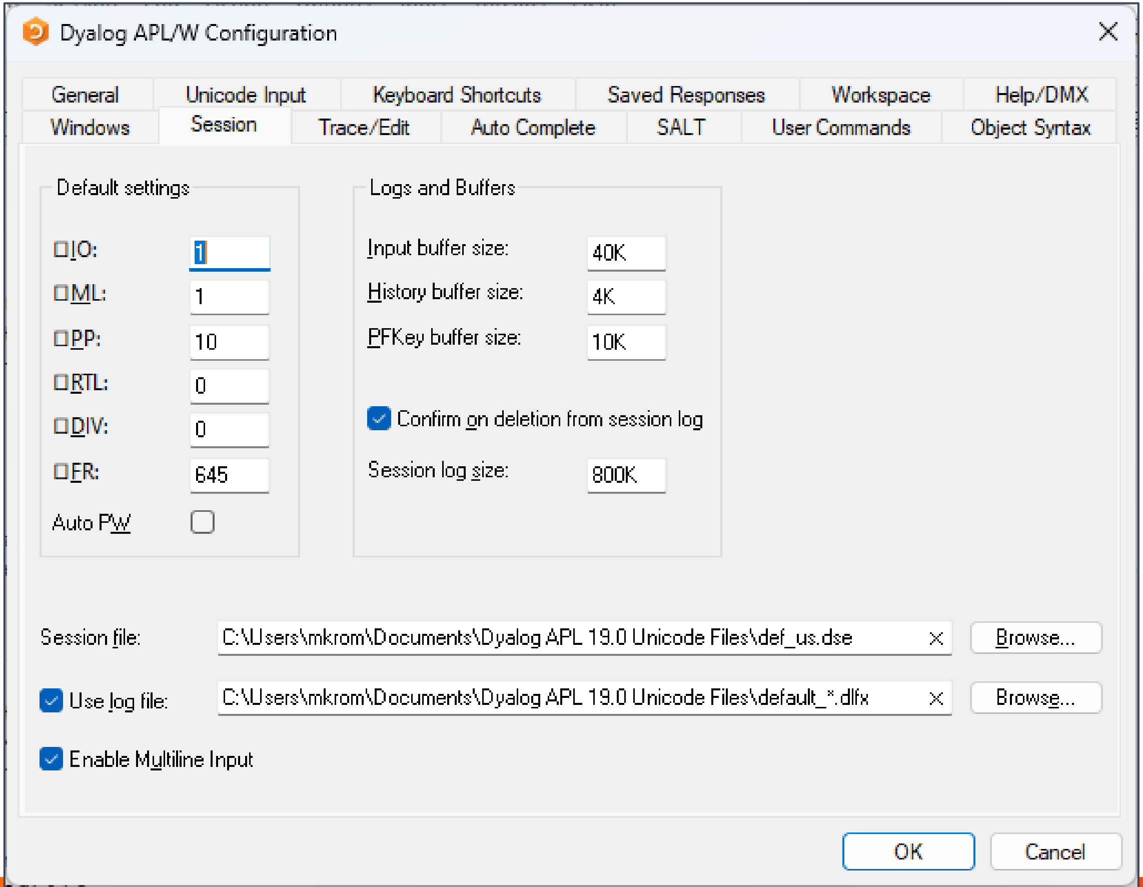
Multiple Session Log Files

- Before v19.0, the default would be for all Dyalog processes to use the same log file
- This sometimes caused loss of useful log file data
- In the worst case, rare crashes when two processes tried to update the file at the same time
  - (typically when starting or stoppping many processes at once)
- Version 19.0 locks the log file and subsequent processes will open or create a new one

Log Files - Intended Usage Pattern

- Users who always start the same set of interpreters at the start of a working session
  - Development
  - Test
  - Prod
- … will get the same log files each day
- Alternatively, create shortcuts without * in the log file name

Multiple Session Log Files...



Default settings

⎕IO: 1

⎕ML: 1

⎕PP: 10

⎕RTL: 0

⎕DIV: 0

⎕FR: 645

Auto PW □

Logs and Buffers

Input buffer size: 40K

History buffer size: 4K

PFKey buffer size: 10K

☑ Confirm on deletion from session log

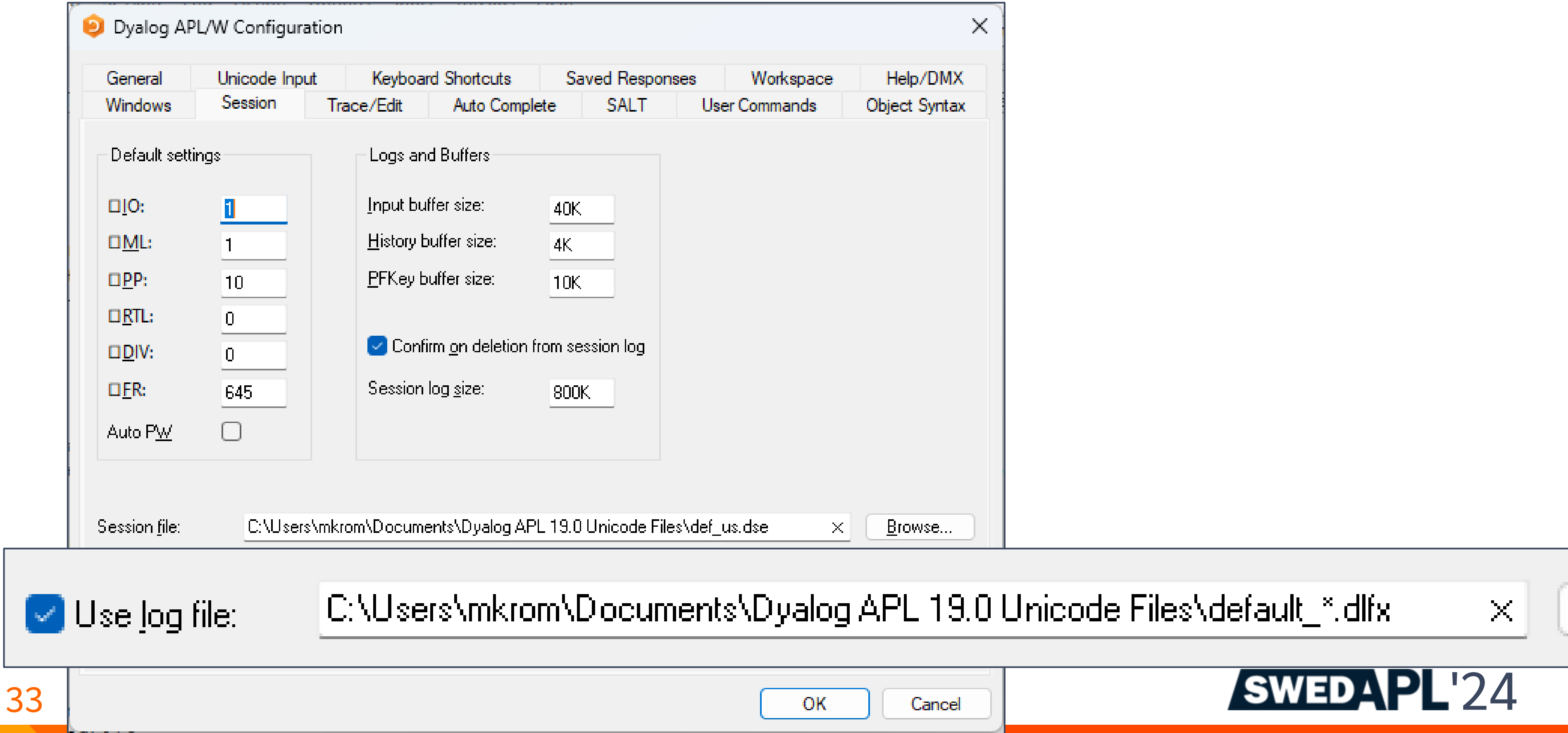
Session log size: 800K

Session file: C:\Users\mkrom\Documents\Dyalog APL 19.0 Unicode Files\def_us.dse

☑ Use log file: C:\Users\mkrom\Documents\Dyalog APL 19.0 Unicode Files\default_*.dlfx

☑ Enable Multiline Input

Multiple Session Log Files...



Default settings

⎕IO: 1

⎕ML: 1

⎕PP: 10

⎕RTL: 0

⎕DIV: 0

⎕FR: 645

Auto PW ☐

Logs and Buffers

Input buffer size: 40K

History buffer size: 4K

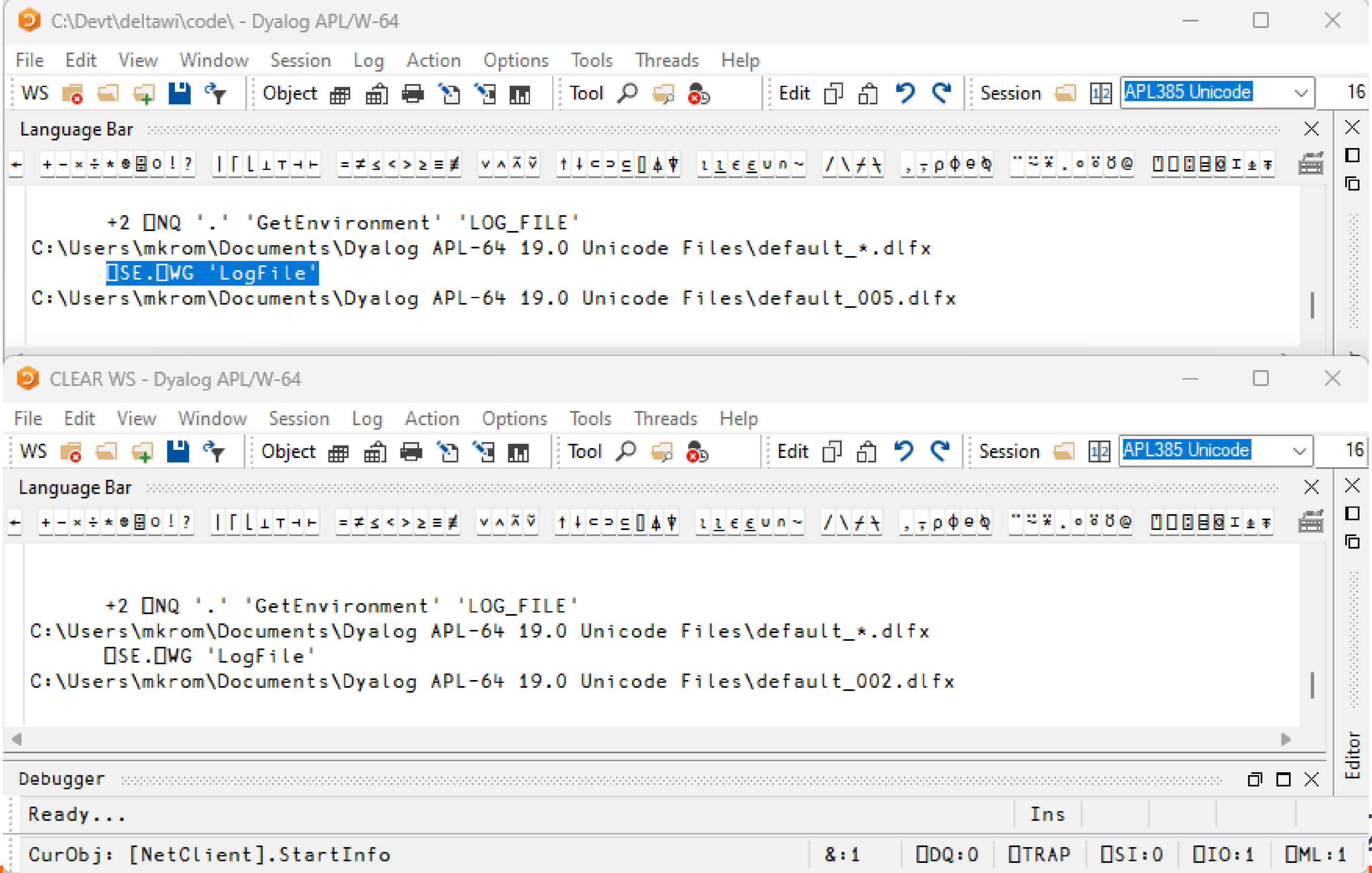
PFKey buffer size: 10K

☑ Confirm on deletion from session log

Session log size: 800K

Session file: C:\Users\mkrom\Documents\Dyalog APL 19.0 Unicode Files\def_us.dse

☑ Use log file: C:\Users\mkrom\Documents\Dyalog APL 19.0 Unicode Files\default_*.dlfx



C:\Devt\deltawi\code\ - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL385 Unicode 16
Language Bar

```
      +2 ⎕NQ '.' 'GetEnvironment' 'LOG_FILE'
C:\Users\mkrom\Documents\Dyalog APL-64 19.0 Unicode Files\default_*.dlfx
      ⎕SE.⎕WG 'LogFile'
C:\Users\mkrom\Documents\Dyalog APL-64 19.0 Unicode Files\default_005.dlfx
```

CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL385 Unicode 16
Language Bar

```
      +2 ⎕NQ '.' 'GetEnvironment' 'LOG_FILE'
C:\Users\mkrom\Documents\Dyalog APL-64 19.0 Unicode Files\default_*.dlfx
      ⎕SE.⎕WG 'LogFile'
C:\Users\mkrom\Documents\Dyalog APL-64 19.0 Unicode Files\default_002.dlfx
```

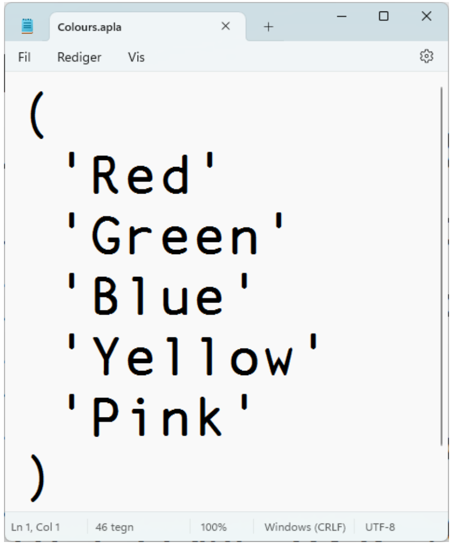
)ed foo[123]

In version 19.0, if you add a line number after the a name, the editor will open on that line

Arrays in "Plain" Text Files

- Some character data doesn't map nicely to APL Array Notation:
  - Simple character vectors
  - Simple character matrices
  - Vectors of character vectors

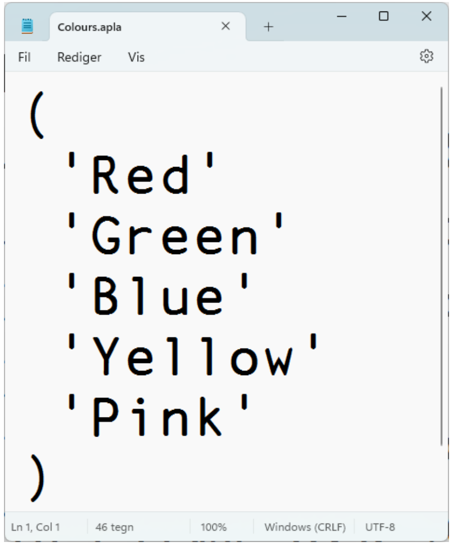
Link v4.0 adds experimental support for storing such arrays in "plain" format



```
(
 'Red'
 'Green'
 'Blue'
 'Yellow'
 'Pink'
)
```

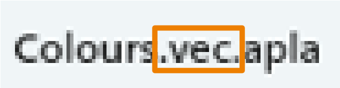
-text=aplan (default)

```
       )ns MyApp
#.MyApp
      #.MyApp.Colours←'Red' 'Green' 'Blue' 'Yellow' 'Pink'
      ]link.create MyApp c:\tmp\myapp -arrays
Linked: #.MyApp ←→ c:\tmp\myapp [directory was created]
      ]open c:\tmp\myapp\Colours.apla
c:\tmp\myapp\Colours.apla
```

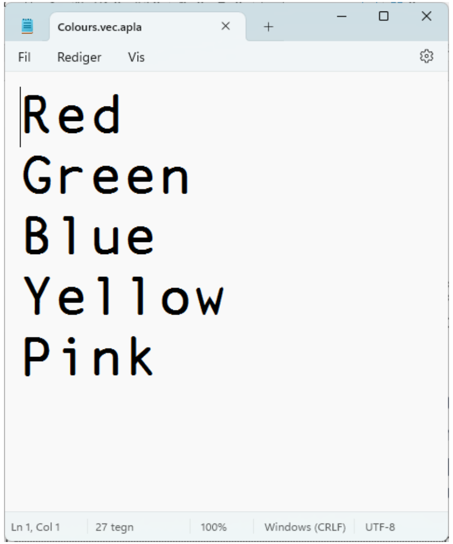


```
(
 'Red'
 'Green'
 'Blue'
 'Yellow'
 'Pink'
)
```

-text=plain



Colours.vec.apla

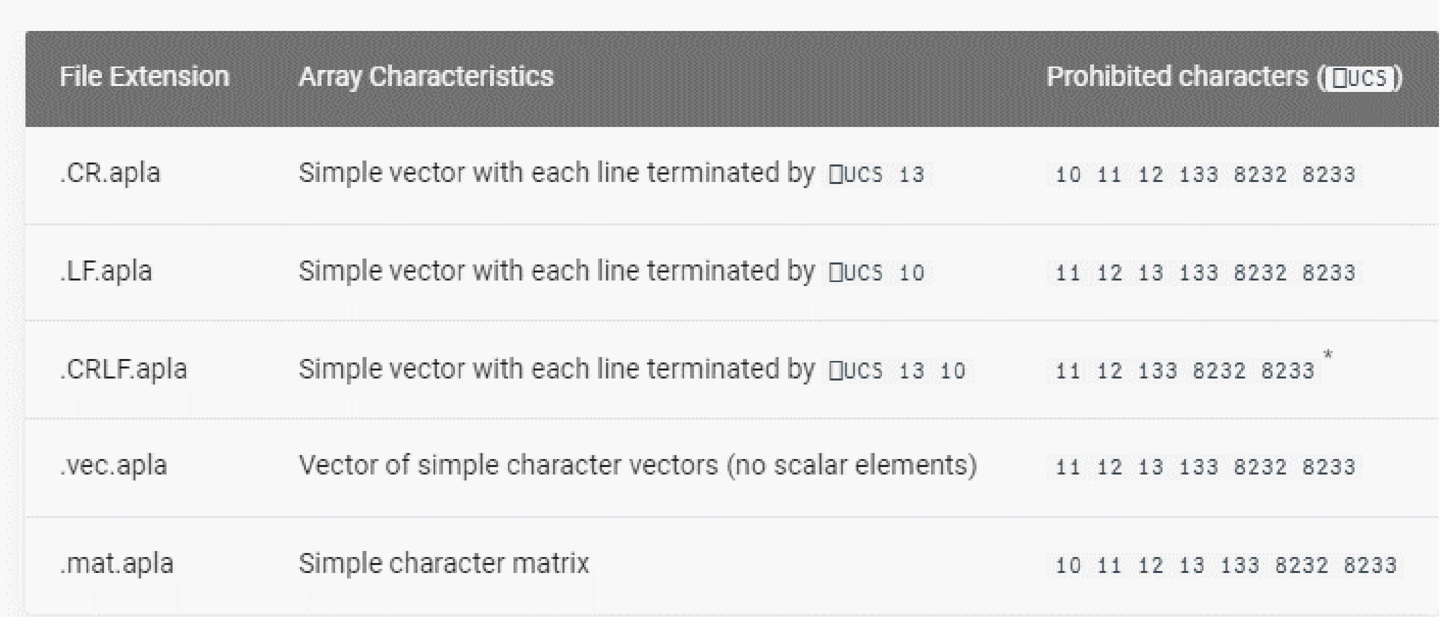


```
Red
Green
Blue
Yellow
Pink
```

```
      )ns MyApp
#.MyApp
      #.MyApp.Colours←'Red' 'Green' 'Blue' 'Yellow' 'Pink'
      ]link.create MyApp c:\tmp\myapp -arrays -text=plain
Linked: #.MyApp ←→ c:\tmp\myapp [directory was created]
      ]open c:\tmp\myapp\Colours.apla
c:\tmp\myapp\Colours.apla
```

text=plain is experimental

- Design may be refined in next release cycle



File Extension	Array Characteristics	Prohibited characters (⎕UCS)
.CR.apla	Simple vector with each line terminated by ⎕UCS 13	10 11 12 133 8232 8233
.LF.apla	Simple vector with each line terminated by ⎕UCS 10	11 12 13 133 8232 8233
.CRLF.apla	Simple vector with each line terminated by ⎕UCS 13 10	11 12 133 8232 8233 *
.vec.apla	Vector of simple character vectors (no scalar elements)	11 12 13 133 8232 8233
.mat.apla	Simple character matrix	10 11 12 13 133 8232 8233

v19.0 .NET Bridge

- Adds support for .NET 6, 7, 8 ...
  - Tested with 6.0 & 8.0
  - Configured for 8.0 by default
  - Also full support for 4.8 (aka ".NET Framework")

- Can export APL code as .NET assemblies
  - Will allow embedding APL code in .NET frameworks like ASP.NET Core, etc

.NET 8.0 is the Long Term Support version (LTS)

Service Orientation

A rapidly increasing proportion of new APL code is delivered as services

- ***Jarvis*** wraps APL code as HTTP/JSON or RESTful services on any platform
  - https://github.com/dyalog/jarvis
- Off-the-shelf docker containers containing Dyalog APL (optionally with Jarvis)
- ***HttpCommand*** is our HTTP client

RESTful API

GET PUT POST DELETE

HTTP(s) / JSON

docker

Health Monitor

Installing & Managing

Experimental TCP-based monitor:

- Interpreter can broadcast regular updates on (e.g.):
  - CPU consumption, Memory statistics
  -)SI stack and Error information
- Notifications on
  - untrapped errors
  - ws compaction
- Exact execution location is available as an option (with runtime cost)
- Also broadcasts details of how to connect Remote IDE if that is possible

Health Monitor Example

```
["PollFacts",{"Facts":["AccountInformation","Workspace","ThreadCount"],"Interval":5000,"UID":"1 1"}]
```

```
["Facts",
{"Facts": [ {
  "ID": 2, "Name": "AccountInformation",
  "Value": {
  "ComputeTime": 438,
  "ConnectTime": 46973,
  "KeyingTime": 0,
  "UserIdentification": 0
}},{
  "ID": 3, "Name": "Workspace",
  "Value": {
  "Allocation": 33882064,
  "AllocationHWM": 33882064,
  "Available": 2144207480,
  "Compactions": 2,
  "FreePockets": 186682,
  "GarbageCollections": 0,
  "GarbagePockets": 0,
  "Sediment": 2120,
  "Used": 3276168,
  "UsedPockets": 23209,
  "WSID": "CLEAR WS"
```

```
}},{
  "ID": 6, "Name": "ThreadCount",
  "Value": {
  "Suspended": 1,
  "Total": 2
}}
],
"Interval": 5000,
"UID": "1 1"
}]
```

Health Monitor Example

```
["PollFacts",{"Facts":["AccountInformation","Workspace","ThreadCount"],"Interval":5000,"UID":"1 1"}]
```

```
["Facts",
{"Facts": [ {
  "ID": 2, "Name": "AccountInformation",
  "Value": {
  "ComputeTime": 438,
  "ConnectTime": 46973,
  "KeyingTime": 0,
  "UserIdentification": 0
}},{
  "ID": 3, "Name": "Workspace",
  "Value": {
  "Allocation": 33882064,
  "AllocationHWM": 33882064,
  "Available": 2144207480,
  "Compactions": 2,
  "FreePockets": 186682,
  "GarbageCollections": 0,
  "GarbagePockets": 0,
  "Sediment": 2120,
  "Used": 3276168,
  "UsedPockets": 23209,
  "WSID": "CLEAR WS"
```

```
}},{
  "ID": 6, "Name": "ThreadCount",
  "Value": {
  "Suspended": 1,
  "Total": 2
}}
],
"Interval": 5000,
"UID": "1 1"
}]
```

Health Monitor Example

```
["PollFacts",{"Facts":["AccountInformation","Workspace","ThreadCount"],"Interval":5000,"UID":"1 1"}]
```

```
["Facts",
{"Facts": [ {
  "ID": 2, "Name": "AccountInformation",
  "Value": {
  "ComputeTime": 438,
  "ConnectTime": 46973,
  "KeyingTime": 0,
  "UserIdentification": 0
}},{
  "ID": 3, "Name": "Workspace",
  "Value": {
  "Allocation": 33882064,
  "AllocationHWM": 33882064,
  "Available": 2144207480,
  "Compactions": 2,
  "FreePockets": 186682,
  "GarbageCollections": 0,
  "GarbagePockets": 0,
  "Sediment": 2120,
  "Used": 3276168,
  "UsedPockets": 23209,
  "WSID": "CLEAR WS"
}},{
  "ID": 6, "Name": "ThreadCount",
  "Value": {
  "Suspended": 1,
  "Total": 2
}}
],
"Interval": 5000,
"UID": "1 1"
}]
```

Health Monitor Example

```
["PollFacts",{"Facts":["AccountInformation","Workspace","ThreadCount"],"Interval":5000,"UID":"1 1"}]
```

```
["Facts",
{"Facts": [ {
  "ID": 2, "Name": "AccountInformation",
  "Value": {
  "ComputeTime": 438,
  "ConnectTime": 46973,
  "KeyingTime": 0,
  "UserIdentification": 0
}},{
  "ID": 3, "Name": "Workspace",
  "Value": {
  "Allocation": 33882064,
  "AllocationHWM": 33882064,
  "Available": 2144207480,
  "Compactions": 2,
  "FreePockets": 186682,
  "GarbageCollections": 0,
  "GarbagePockets": 0,
  "Sediment": 2120,
  "Used": 3276168,
  "UsedPockets": 23209,
  "WSID": "CLEAR WS"
```

```
}},{
  "ID": 6, "Name": "ThreadCount",
  "Value": {
  "Suspended": 1,
  "Total": 2
}}
],
"Interval": 5000,
"UID": "1 1"
}]
```

Health Monitor Example

```
["PollFacts",{"Facts":["AccountInformation","Workspace","ThreadCount"],"Interval":5000,"UID":"1 1"}]
```

```
["Facts",
{"Facts": [ {
  "ID": 2, "Name": "AccountInformation",
  "Value": {
  "ComputeTime": 438,
  "ConnectTime": 46973,
  "KeyingTime": 0,
  "UserIdentification": 0
}},{
  "ID": 3, "Name": "Workspace",
  "Value": {
  "Allocation": 33882064,
  "AllocationHWM": 33882064,
  "Available": 2144207480,
  "Compactions": 2,
  "FreePockets": 186682,
  "GarbageCollections": 0,
  "GarbagePockets": 0,
  "Sediment": 2120,
  "Used": 3276168,
  "UsedPockets": 23209,
  "WSID": "CLEAR WS"
```

```
}},{
  "ID": 6, "Name": "ThreadCount",
  "Value": {
  "Suspended": 1,
  "Total": 2
}}
],
"Interval": 5000,
"UID": "1 1"
}]
```

Health Monitor Example

```
["PollFacts",{"Facts":["AccountInformation","Workspace","ThreadCount"],"Interval":5000,"UID":"1 1"}]
```

```
["Facts",
{"Facts": [ {
  "ID": 2, "Name": "AccountInformation",
  "Value": {
  "ComputeTime": 438,
  "ConnectTime": 46973,
  "KeyingTime": 0,
  "UserIdentification": 0
}},{
  "ID": 3, "Name": "Workspace",
  "Value": {
  "Allocation": 33882064,
  "AllocationHWM": 33882064,
  "Available": 2144207480,
  "Compactions": 2,
  "FreePockets": 186682,
  "GarbageCollections": 0,
  "GarbagePockets": 0,
  "Sediment": 2120,
  "Used": 3276168,
  "UsedPockets": 23209,
  "WSID": "CLEAR WS"
```

```
}},{
  "ID": 6, "Name": "ThreadCount",
  "Value": {
  "Suspended": 1,
  "Total": 2
}}
],
"Interval": 5000,
"UID": "1 1"
}]
```

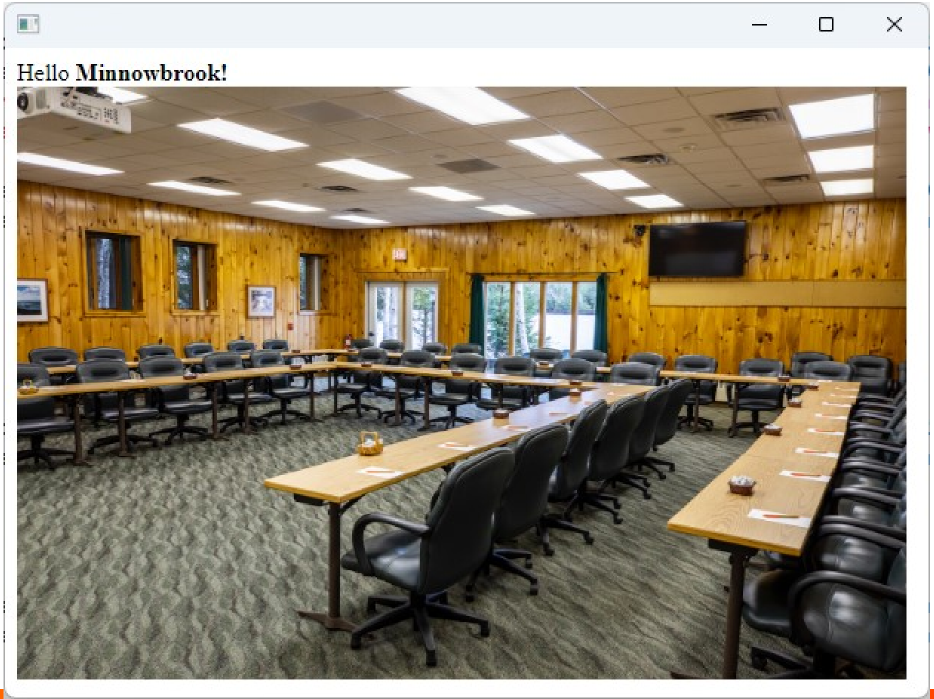
HTMLRenderer updates

New features include:

- AllowContextMenu
- ExecuteJavascript
- Get/SetZoomLevel
- IsLoading + LoadEnd event

HTMLRenderer – what's that?

```
pic←'https://minnowbrook.org/wp-content/uploads/2022/04/
                  Screen-Shot-2022-04-18-at-2.24.30-PM.png'
'MyForm' ⎕WC 'HTMLRenderer' ('Hello <b>Minnowbrook!</b><br/>
                  <img src="',pic,'" height=400 width=600>')
```



Hello **Minnowbrook!**

Chromium Embedded Framework (CEF)

HTMLRenderer updates

New features include:

- AllowContextMenu
- ExecuteJavascript
- Get/SetZoomLevel
- IsLoading + LoadEnd event

Sketch of Version 20.0 (Q2/2025)

Transferred from v19.0

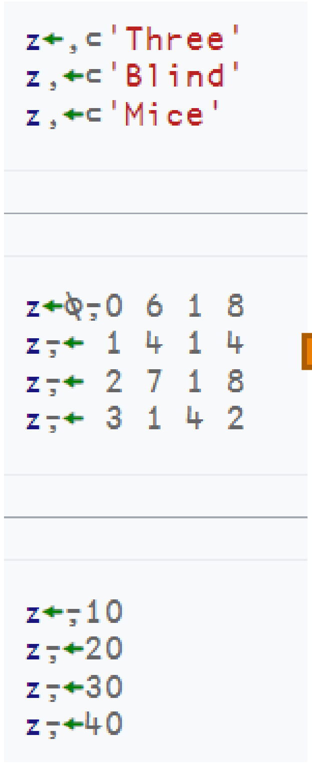
- Resume Optimisation Work
- .NET Bridge "enhancements"
  - Support "Generic" methods & classes
- More HTMLRenderer improvements
- Health Monitor
- Script Engine Support
- ⎕NATTRIBUTES

Next Set of Projects

- Relax Interpreter Limits
- Array Notation
- Token-by-token Debugging
- Query Platform Features
- New "Shell" System Command
- Open-Source HTMLRenderer & Conga
- JavaScript emulation of ⎕WC

Array Notation

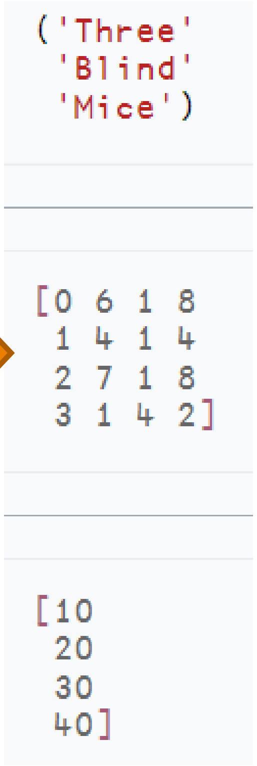
https://aplwiki.com/wiki/Array_notation



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

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

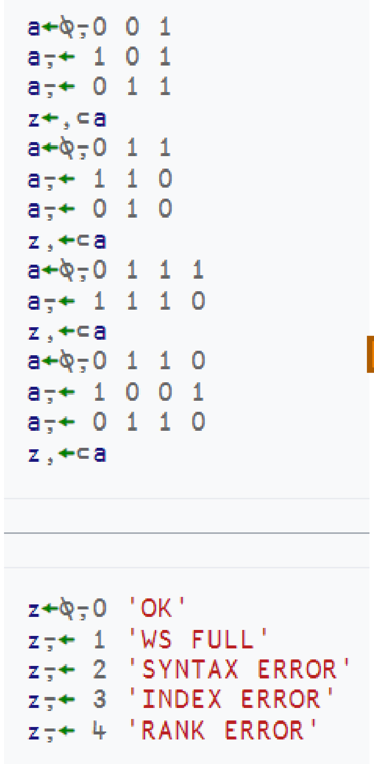
```
z←⍪10
z⍪←20
z⍪←30
z⍪←40
```



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

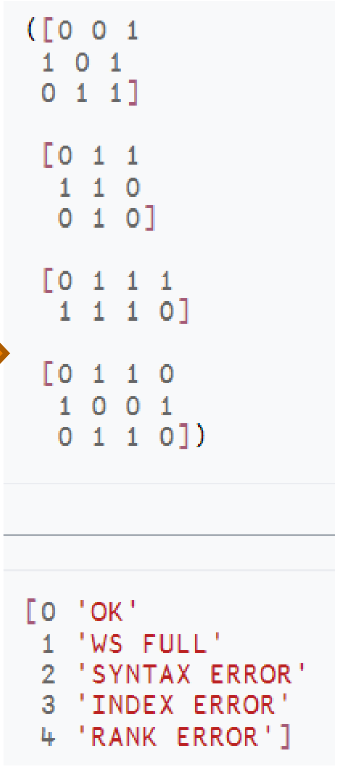
```
[0 6 1 8
 1 4 1 4
 2 7 1 8
 3 1 4 2]
```

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



```
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
```

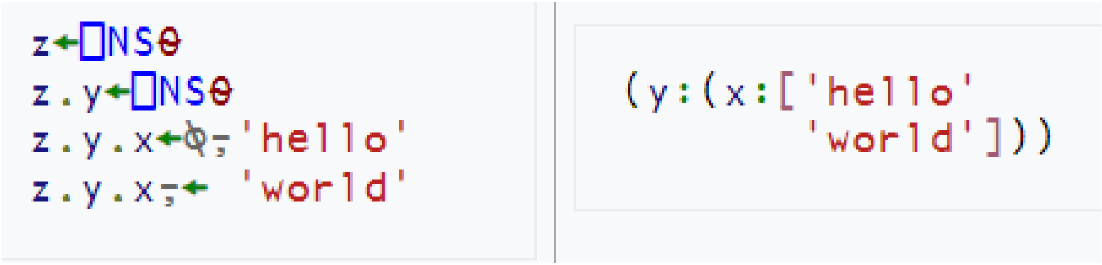
```
z←⍉⍪0 'OK'
z⍪← 1 'WS FULL'
z⍪← 2 'SYNTAX ERROR'
z⍪← 3 'INDEX ERROR'
z⍪← 4 'RANK ERROR'
```



```
([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])
```

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

Array Notation



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

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

- Public proposal published
  - https://aplwiki.com/wiki/Array_notation
- APL model exists in Link since v3.0 and `⎕SE.Dyalog.Array.Serialise|Deserialise`
- Will be integrated with interpreter in v20.0

Get and Set values without Execute

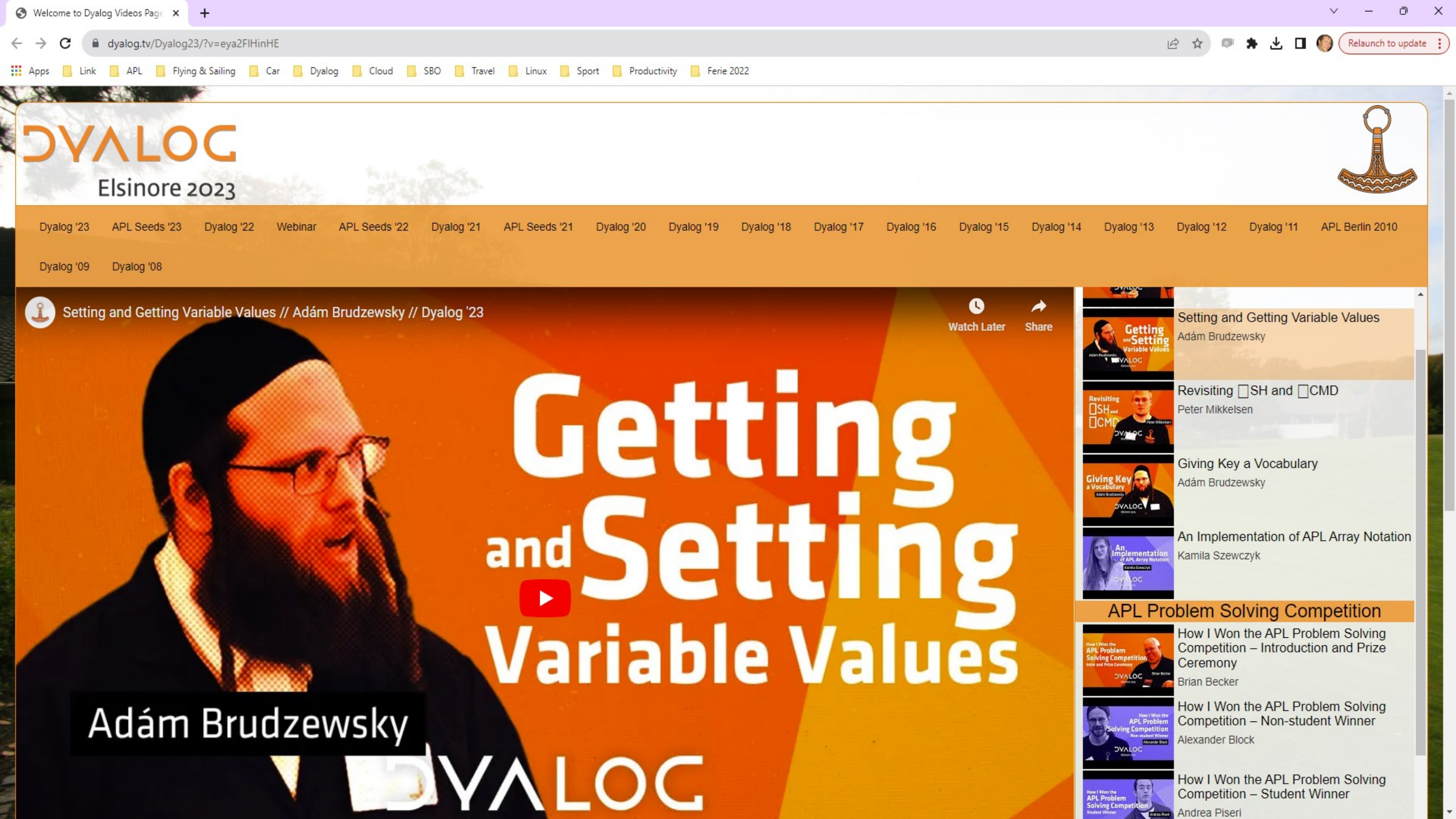
- Working with data where the names of variables are held in an array
- A common requirement when writing tools
- Should have been done 20 years ago

⎕NG: Name Get

```
      FirstName←'Jill' ⋄ Age←32 ⍝ Some data

      ⎕NG 'Age' 'FirstName' ⍝ Avoid ⍎
32  Jill

      ⎕NG ('Age' 0) ('LastName' 'Unknown')
      ⍝ (Name Default) pairs
32  Unknown

⍝ Left argument can be a namespace reference
```

⎕NS: Name Set

```
      ⎕NS ('FirstName' 'Jill') ('Age' 32)
      ⍝ (Name Value) pairs

 ⍝ But what if you have a vector of values?

      ⎕NS (↑'FirstName' 'Age') ('Jill' 32)
      ⍝ (Matrix of Names) (Vector of Values)

 ⍝ Left argument can be a namespace
reference
```



DYALOG
Elsinore 2023

Dyalog '23 APL Seeds '23 Dyalog '22 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

Setting and Getting Variable Values // Adám Brudzewsky // Dyalog '23

Setting and Getting Variable Values
Adám Brudzewsky

Revisiting ⎕SH and ⎕CMD
Peter Mikkelsen

Giving Key a Vocabulary
Adám Brudzewsky

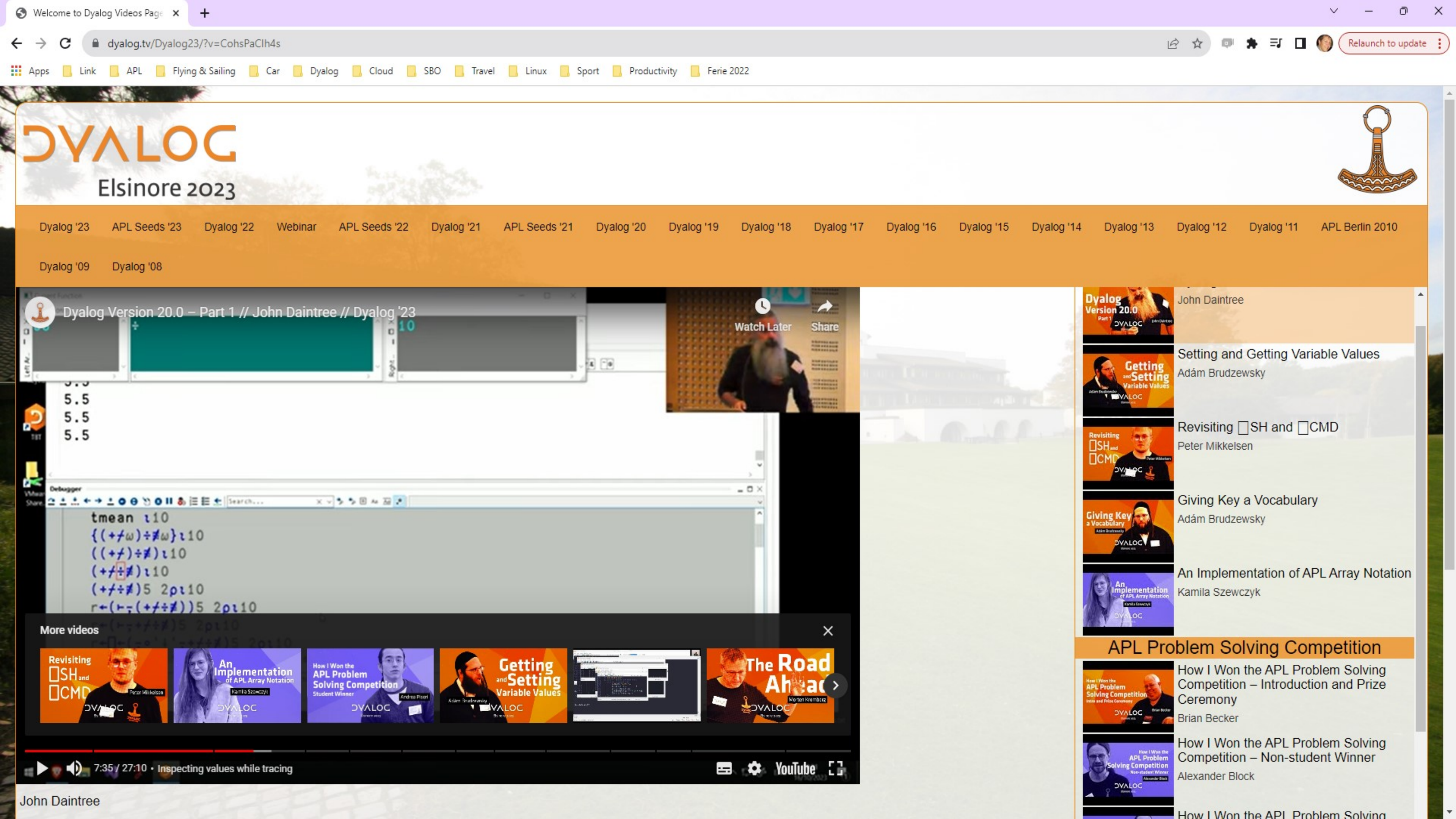
An Implementation of APL Array Notation
Kamila Szewczyk

APL Problem Solving Competition

How I Won the APL Problem Solving Competition – Introduction and Prize Ceremony
Brian Becker

How I Won the APL Problem Solving Competition – Non-student Winner
Alexander Block

How I Won the APL Problem Solving Competition – Student Winner
Andrea Piseri



DYALOG

Elsinore 2023

Dyalog Version 20.0 – Part 1 // John Daintree // Dyalog '23

```
5.5
5.5
5.5
```

Debugger

```
tmean ⍳10
{(+⌿⍵)÷≢⍵}⍳10
((+⌿)÷≢)⍳10
(+⌿÷≢)⍳10
(+⌿÷≢)5 2⍴⍳10
r←(⊢⍪(+⌿÷≢))5 2⍴⍳10
```

More videos

Revisiting ⎕SH and ⎕CMD
An Implementation of APL Array Notation
How I Won the APL Problem Solving Competition
Getting and Setting Variable Values
The Road Ahead

7:35 / 27:10 • Inspecting values while tracing

John Daintree

Dyalog Version 20.0 Part 1
John Daintree

Setting and Getting Variable Values
Adám Brudzewsky

Revisiting ⎕SH and ⎕CMD
Peter Mikkelsen

Giving Key a Vocabulary
Adám Brudzewsky

An Implementation of APL Array Notation
Kamila Szewczyk

APL Problem Solving Competition

How I Won the APL Problem Solving Competition – Introduction and Prize Ceremony
Brian Becker

How I Won the APL Problem Solving Competition – Non-student Winner
Alexander Block

How I Won the APL Problem Solving

.NET Bridge Enhancements

- The v19.0 bridge to .NET 6/7/8 is roughly on par with the Framework bridge
- In v20.0, the new bridge may move ahead, adding:
  - Generics
  - Delegates
  - Async
- Some of these will only be designed / modelled in v20

HTMLRenderer Enhancements

Suggested:

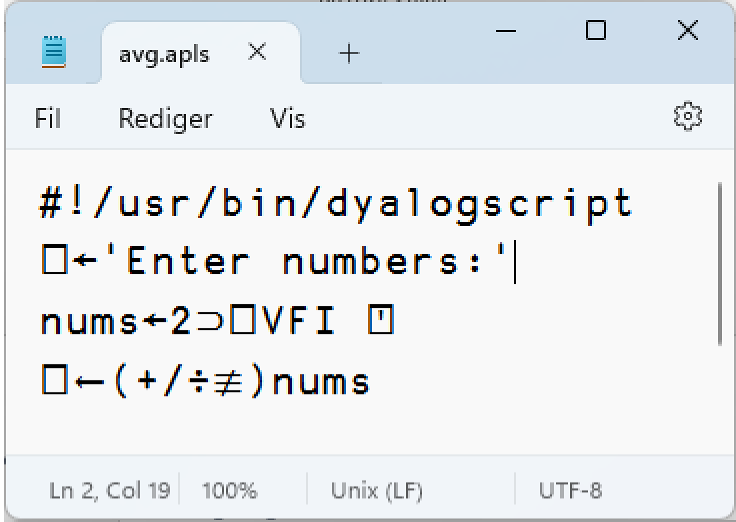
- File Upload
- Modal HTMLRenderer Windows
- More control over "Chrome"
- Other changes driven by JSWC project

HTMLRenderer & Conga "Open Source"

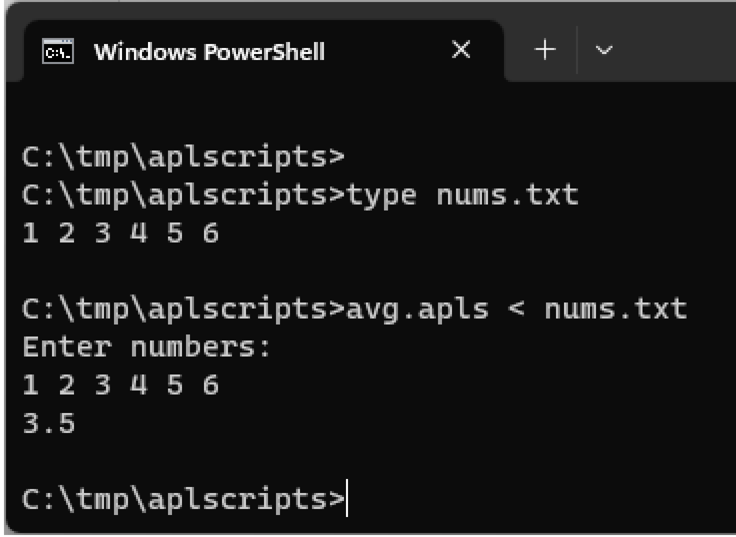
- Allow users to compile own versions of "plugins" and make pull requests
  - Allow users to move a bit faster when Dyalog is slow to e.g. use a new version of Chromium
  - Help grow the community by open sourcing more things
- This might include the Dyalog Cryptographic Library and user-defined plugins
- A modern replacement for Auxiliary Processors
- Conga may also be loaded and initialised at startup, avoiding current issues with initialisation

Script-Engine Support

- #! (hash bang) scripting
- We think the script engine is critical for attracting new users
- Still a bit of a prototype
  - Will be hardened for v20.0
  - Need to be able to debug scripts via RIDE



```
#!/usr/bin/dyalogscript
⎕←'Enter numbers:'
nums←2⊃⎕VFI ⍞
⎕←(+/÷≢)nums
```



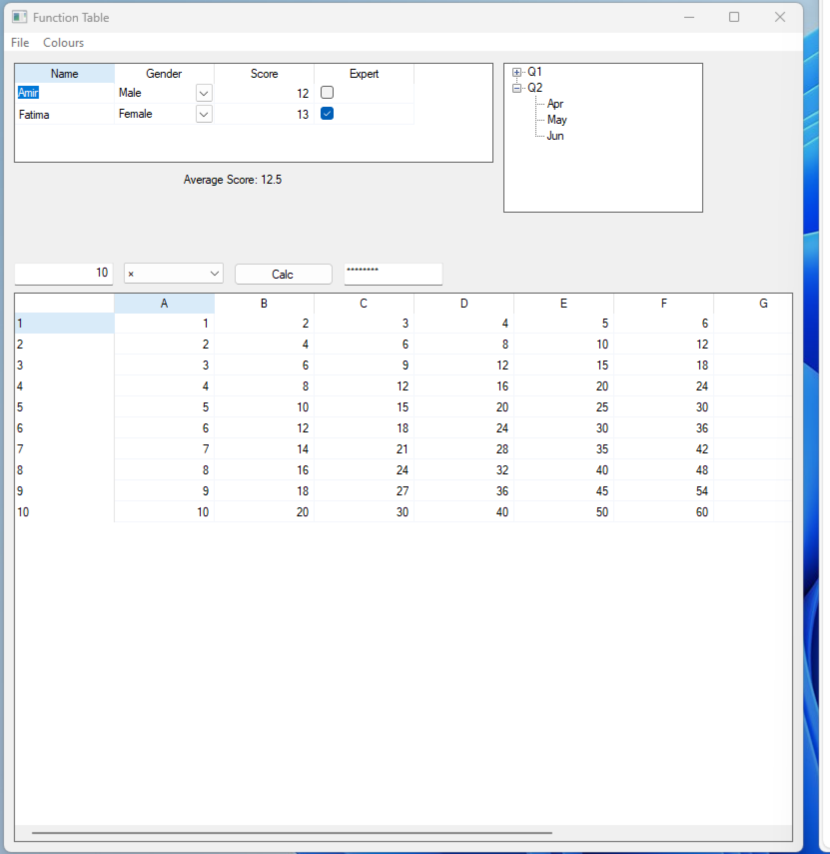
```
C:\tmp\aplscripts>
C:\tmp\aplscripts>type nums.txt
1 2 3 4 5 6

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

C:\tmp\aplscripts>
```

Easy Web Creator - EWC

- A JavaScript emulation of our Win32 layer (⎕WC, ⎕WG, ⎕WS ...)
- Still mostly referred to as JSWC (JavaScript Window Create)

Demo of JSWC

JavaScript □WC

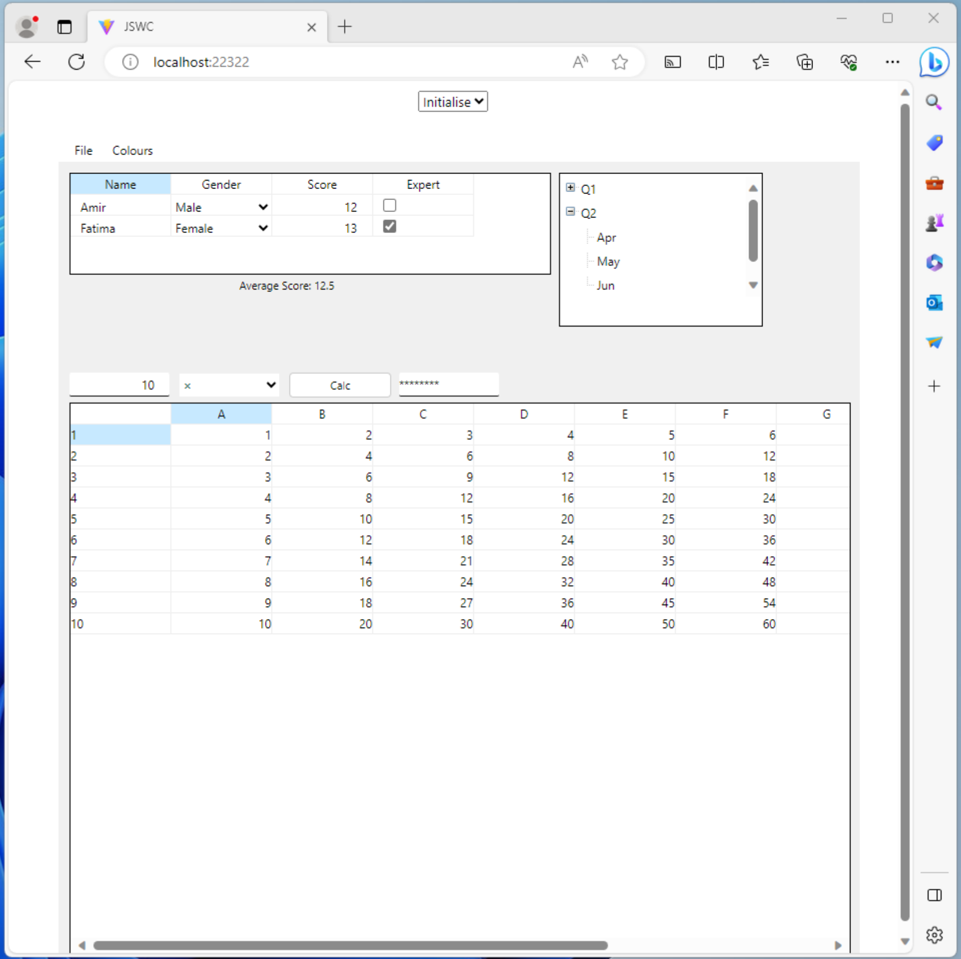


Name	Gender	Score	Expert
Amir	Male	12	□
Fatima	Female	13	☑

Average Score: 12.5

10 × ********

	A	B	C	D	E	F	G
1	1	2	3	4	5	6	
2	2	4	6	8	10	12	
3	3	6	9	12	15	18	
4	4	8	12	16	20	24	
5	5	10	15	20	25	30	
6	6	12	18	24	30	36	
7	7	14	21	28	35	42	
8	8	16	24	32	40	48	
9	9	18	27	36	45	54	
10	10	20	30	40	50	60	



Name	Gender	Score	Expert
Amir	Male	12	□
Fatima	Female	13	☑

Average Score: 12.5

10 × ********

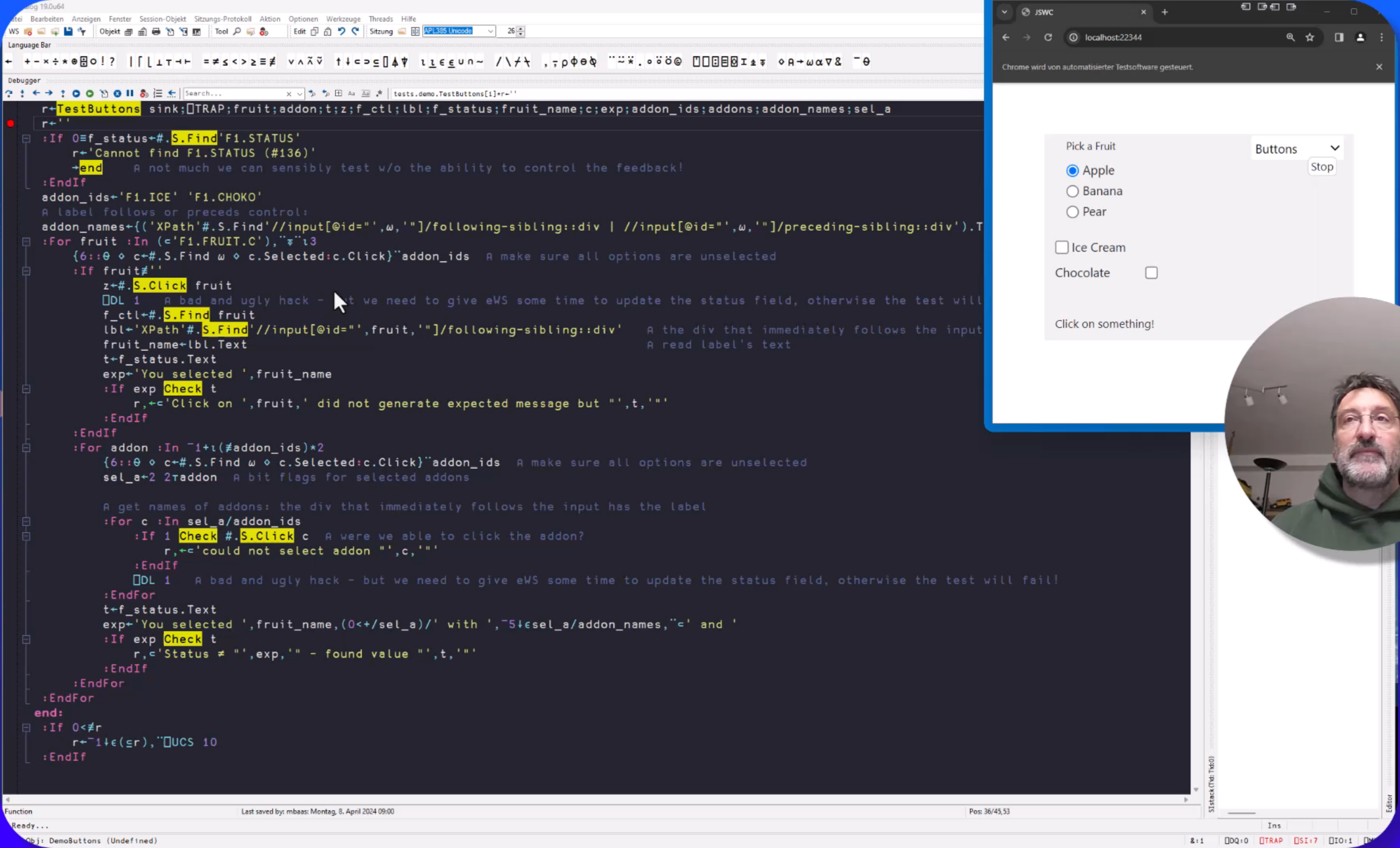
	A	B	C	D	E	F	G
1	1	2	3	4	5	6	
2	2	4	6	8	10	12	
3	3	6	9	12	15	18	
4	4	8	12	16	20	24	
5	5	10	15	20	25	30	
6	6	12	18	24	30	36	
7	7	14	21	28	35	42	
8	8	16	24	32	40	48	
9	9	18	27	36	45	54	
10	10	20	30	40	50	60	

EWC also Open Source

- Both the server-side APL code and the client-side JavaScript
- Users will be able to create their own widgets
- We are planning

Automated Testing w/ Selenium

- One important benefit of HTML rendering is that Selenium can be used for automated testing


Datei Bearbeiten Anzeigen Fenster Session-Objekt Sitzungs-Protokoll Aktion Optionen Werkzeuge Threads Hilfe
Language Bar
Debugger
Search...
tests.demo.TestButtons[1]*r←''
JSWC
localhost:22344
Chrome wird von automatisierter Testsoftware gesteuert.
Pick a Fruit
Buttons
Stop
◉ Apple
○ Banana
○ Pear
☐ Ice Cream
Chocolate ☐
Click on something!
Function
Ready...
Ins

```
r←TestButtons sink;⎕TRAP;fruit;addon;t;z;f_ctl;lbl;f_status;fruit_name;c;exp;addon_ids;addons;addon_names;sel_a
r←''
:If 0≡f_status←#.S.Find'F1.STATUS'
    r←'Cannot find F1.STATUS (#136)'
    →end    ⍝ not much we can sensibly test w/o the ability to control the feedback!
:EndIf
addon_ids←'F1.ICE' 'F1.CHOKO'
⍝ label follows or preceds control:
addon_names←{('XPath'#.S.Find'//input[@id="',⍵,'"]/following-sibling::div | //input[@id="',⍵,'"]/preceding-sibling::div').T
:For fruit :In (⊂'F1.FRUIT.C'),¨⍕¨⍳3
    {6::⍬ ⋄ c←#.S.Find ⍵ ⋄ c.Selected:c.Click}¨addon_ids  ⍝ make sure all options are unselected
    :If fruit≢''
        z←#.S.Click fruit
        ⎕DL 1   ⍝ bad and ugly hack - but we need to give eWS some time to update the status field, otherwise the test will
        f_ctl←#.S.Find fruit
        lbl←'XPath'#.S.Find'//input[@id="',fruit,'"]/following-sibling::div'   ⍝ the div that immediately follows the input
        fruit_name←lbl.Text                                                     ⍝ read label's text
        t←f_status.Text
        exp←'You selected ',fruit_name
        :If exp Check t
            r,←⊂'Click on ',fruit,' did not generate expected message but "',t,'"'
        :EndIf
    :EndIf
    :For addon :In ¯1+⍳(≢addon_ids)*2
        {6::⍬ ⋄ c←#.S.Find ⍵ ⋄ c.Selected:c.Click}¨addon_ids  ⍝ make sure all options are unselected
        sel_a←2 2⊤addon  ⍝ bit flags for selected addons

        ⍝ get names of addons: the div that immediately follows the input has the label
        :For c :In sel_a/addon_ids
            :If 1 Check #.S.Click c  ⍝ were we able to click the addon?
                r,←⊂'could not select addon "',c,'"'
            :EndIf
            ⎕DL 1   ⍝ bad and ugly hack - but we need to give eWS some time to update the status field, otherwise the test will fail!
        :EndFor
        t←f_status.Text
        exp←'You selected ',fruit_name,(0<+/sel_a)/' with ',¯5↓∊sel_a/addon_names,¨⊂' and '
        :If exp Check t
            r,⊂'Status ≠ "',exp,'" - found value "',t,'"'
        :EndIf
    :EndFor
:EndFor
end:
:If 0<≢r
    r←¯1↓∊(⊆r),¨⎕UCS 10
:EndIf
```

Last saved by: mbaas: Montag, 8. April 2024 09:00

Pos: 36/45,53

Obj: DemoButtons (Undefined)

Health Monitor

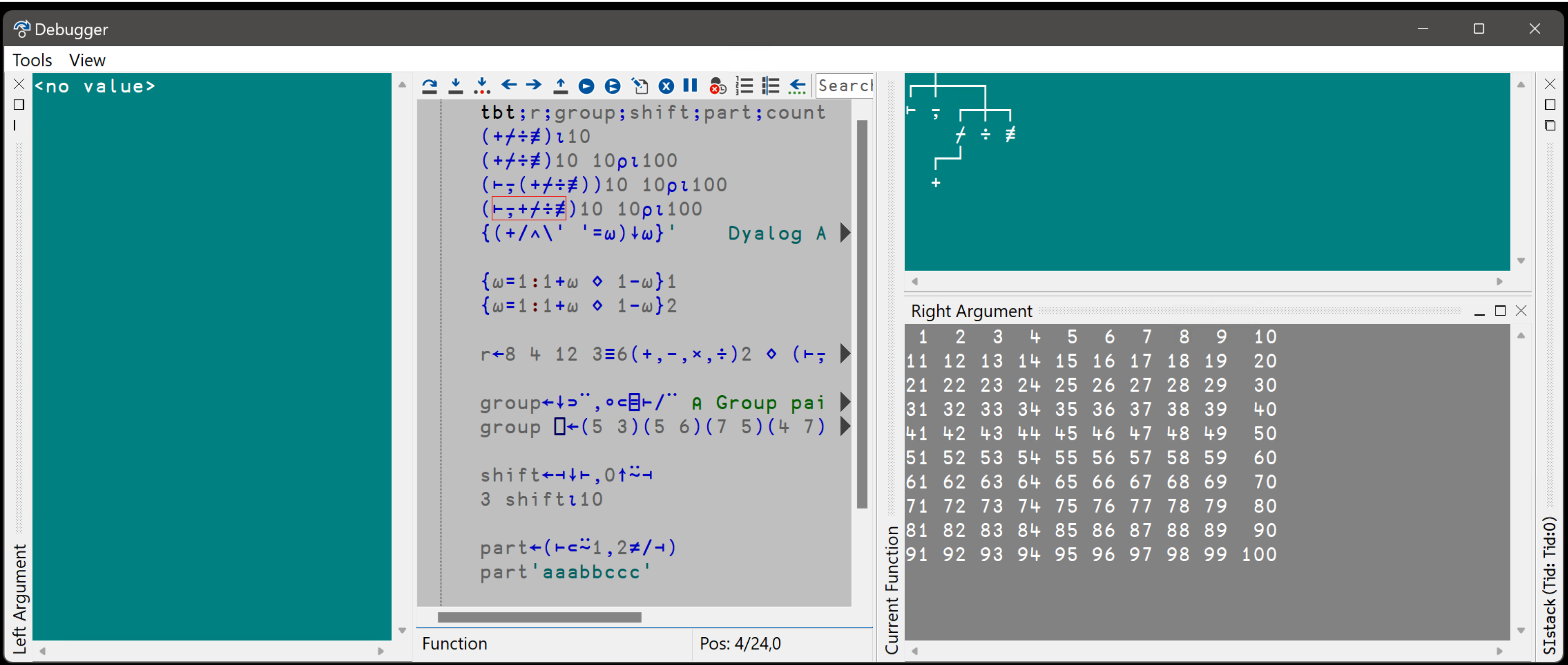
Version 19.0 contains a prototype. Ideas for v20.0 include:

- Add feature to find last known location of a "hanging" interpreter
- Sending signals to interrupt or terminate tasks
- Discoverability: allow APL process to broadcast services that it provides
- Switch ⎕PROFILE on and off; collect data

Token-by-token Debugging

- See John's presentations at Dyalog'22 – and Dyalog'23

Token-by-token Debugging



```
<no value>
```

```
tbt;r;group;shift;part;count
(+⌿÷≢)⍳10
(+⌿÷≢)10 10⍴⍳100
(⊢⍮(+⌿÷≢))10 10⍴⍳100
(⊢⍮+⌿÷≢)10 10⍴⍳100
{(+/∧\' '=⍵)↓⍵}'      Dyalog A

{⍵=1:1+⍵ ⋄ 1-⍵}1
{⍵=1:1+⍵ ⋄ 1-⍵}2

r←8 4 12 3≡6(+,-,×,÷)2 ⋄ (⊢⍮

group←↓⊃¨,∘⊂⌸⊢/¨ ⍝ Group pai
group ⎕←(5 3)(5 6)(7 5)(4 7)

shift←⊣↓⊢,0↑⍨⊣
3 shift⍳10

part←(⊢⊂⍨1,2≠/⊣)
part'aaabbccc'
```

Pos: 4/24,0

```
┌──┴───┐
⊢ ⍮  ┌──┼──┐
     ⌿  ÷  ≢
  ┌──┘
  +
```

```
 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
```

Static Analysis of APL Code

- Static Analysis of application code is seen as a required "best practice" by some corporations
- We are building a prototype of a tool which will
  - Detect vulnerabilities
  - "Lint" APL Code
  - Compute readability and other metrics
- This tool will initially be licensed separately
- A free "community edition" may follow

⎕SHELL to replace existing ⎕SH

Invoke OS commands from APL

- Interruptible
- Optionally return data as an asynchronous Stream
- Manage stdin, stdout & stderr independently
- Handle variety of data encodings

No New Primitives in v20.0

- There is a small set of APL primitives that are "missing"
- See Adam Brudzewsky's presentation "Filling the Core Language Gaps" at Dyalog'22
- These were expected in v20
- However, we are going to spend the time relaxing limitations in the interpreter
  - Max Rank, # of lines in a function, tokens on a line, more primitives, …
- New primitives will appear in v21

Core Language

Data Transformation	Select	Y[X;;]	X⊇Y
Function Application	Depth	X f¨¨⊂⊂Y	X f⍥k Y
Function Composition	Behind	(f X)g Y	X f⍛g Y

What about...

- HTTP2
- Kafka

Sketch of Version 20.0 (Q2/2025)

Transferred from v19.0

- Resume Optimisation Work
- .NET Bridge "enhancements"
  - Support "Generic" methods & classes
- More HTMLRenderer improvements
- Health Monitor
- Script Engine Support
- ⎕NATTRIBUTES

Next Set of Projects

- Relax Interpreter Limits
- Array Notation
- Token-by-token Debugging
- Query Platform Features
- New "Shell" System Command
- Open-Source HTMLRenderer & Conga
- JavaScript emulation of ⎕WC