Dyalog North America Meetup, 11 April 2024

Dyalog
Road Map Update

Morten Kromberg, CTO

Features of Version 19.0 (March 24)

- Platform Support / Distribution
  - 64-bit ARM support
    - New Macs, Pi 4&5, AWS Graviton
  - Enhanced .NET Bridge
    - Framework vs new .NET versions
  - Bound executables on all platforms
- Building Production Systems
  - Token range reservation
  - WS FULL handling
  - NCOPY/NMOVE callbacks
- Developer Productivity / IDE
  - Source "as typed" by default
  - Multi-line input on by default
  - HTMLRenderer updates
  - Link 4.0: Support for text data
  - HttpCommand client, Jarvis web service
- Installing & Managing APL
  - Multiple session files
  - Health Monitor

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

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

tatin.dev/v1/packages

Apps | Link | JSWC | APL | Flying & Sailing | Car | Dyalog | Cloud | SBO | Travel | Linux | Sport | Productivity | Ferie 2022

apl team-SMTP	SMTP client for sending emails from within Dyalog APL	2	github.com
apl team-Snippets	Manage APL-code snippets	1	github.com
apl team-Tester2	Dyalog APL test framework	1	github.com
apl team-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
apl team-WinReg	Tools for dealing with the Windows Registry	1	github.com
apl team-WinRegSimple	Limited set of tools for dealing with the Windows Registry	1	github.com
apl team-WinSCP_NET	Interface between Dyalog and the WINSCP .NET DLL	1	github.com
apl team-WinSys	Windows-only OS calls and system infos	1	github.com
apl team-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

Tatin

tatin.dev/v1/packages

Apps | Link | JSWC | APL | Flying & Sailing | Car | Dyalog | Cloud | SBO | Travel | Linux | Sport | Productivity | Ferie 2022

apl team-SMTP	SMTP client for sending emails from within Dyalog APL	2	github.com
apl team-Snippets	Manage APL-code snippets	1	github.com
apl team-Tester2	Dyalog APL test framework	1	github.com
apl team-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
apl team-WinReg	Tools for dealing with the Windows Registry	1	github.com
apl team-WinRegSimple	Limited set of tools for dealing with the Windows Registry	1	github.com
apl team-WinSCP_NET	Interface between Dyalog and the WINSCP .NET DLL	1	github.com
apl team-WinSys	Windows-only OS calls and system infos	1	github.com
apl team-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

Tatin

tatin.dev/v1/packages

Apps | Link | JSWC | APL | Flying & Sailing | Car | Dyalog | Cloud | SBO | Travel | Linux | Sport | Productivity | Ferie 2022

apl team-SMTP	SMTP client for sending emails from within Dyalog APL	2	github.com
apl team-Snippets	Manage APL-code snippets	1	github.com
apl team-Tester2	Dyalog APL test framework	1	github.com
apl team-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
apl team-WinReg	Tools for dealing with the Windows Registry	1	github.com
apl team-WinRegSimple	Limited set of tools for dealing with the Windows Registry	1	github.com
apl team-WinSCP_NET	Interface between Dyalog and the WINSCP .NET DLL	1	github.com
apl team-WinSys	Windows-only OS calls and system infos	1	github.com
apl team-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

Multi-line Input

CLEAR WS - Dyalog APL/W-64

File Edit View Window Session Log Action Options Tools Threads Help

WS Object Tool Edit Session APL

Language Bar

```
Serial number: 000013 - pre-release software
Thu Oct 12 22:39:45 2023
      dup←{
       ⍵ ⍵
       }
      dup 42
42 42
      :If (2⍴42) ≡ dup 42
          ⎕←'Success'
      :Else
          ⎕←'Failure'
      :EndIf
Success
```

Editor

Debugger

Ready... Ins

CurObj: S (Undefined) &:1 ⎕DQ:0 ⎕TRAP ⎕SI:0 ⎕IO:1 ⎕ML:1

[Microsoft].NET History

- .NET has been around for 20+ years. The old "Framework" is being replaced by an open source, cross-platform .NET, initially known as ".NET Core".

Name	Platforms	Version Numbers
Microsoft.NET Framework	Windows	1 2 3 4.0 4.8.1
".NET Core"	Windows Linux macOS	1 2 3
".NET"	Windows Linux macOS	5.0 6.0 7.0 8.0

- Dyalog v9.5 added a bridge to the Framework back in 2002
- Dyalog v18.0 added a bridge to .NET Core 3.0, v18.2 targeted 3.1
- v19.0 targets 8.0 (the current Long Term Support version)

v19.0 .NET Bridge

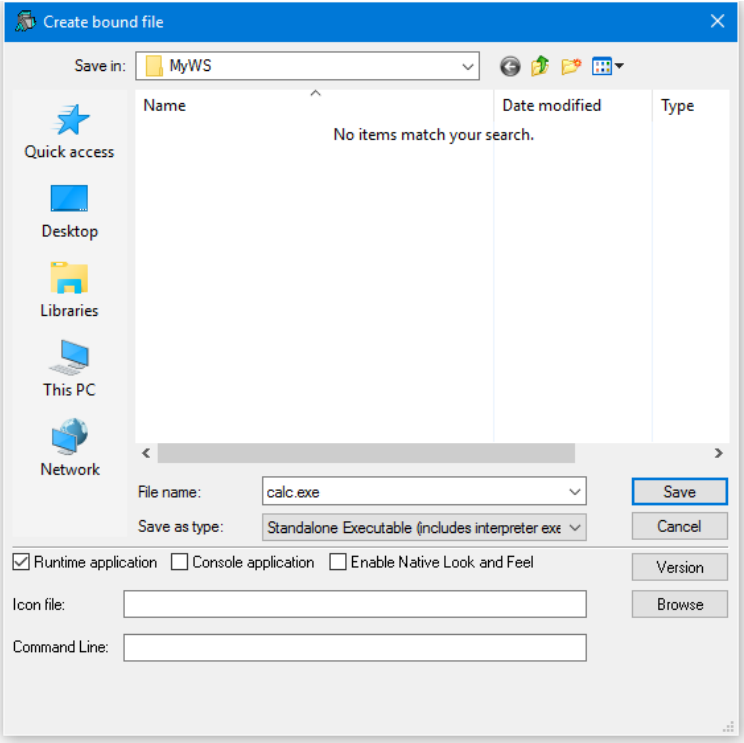
- Adds support for .NET 6, 7, 8 ...
  - Tested with 6.0 & 8.0
  - Configured for 8.0 by default
  - Also full support for 4.8 (aka ".NET Framework")
- Can export APL code as .NET assemblies
  - Will allow embedding APL code in .NET frameworks like ASP.NET Core, etc

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

Bound Executables

A bound executable is a file which combines an interpreter and a workspace into a single .exe file

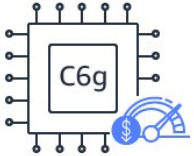
- "Always" been available under Windows
- In v19.0 also available for Linux
  - Maybe MacOS soon
- In the longer term, I expect we will look at encrypting and signing application code



Arm64

64-bit ARM chips are appearing:

- M1, M2, M3 Macs
- Raspberry Pi – 64 Bit
- Amazon Web Services "Graviton"



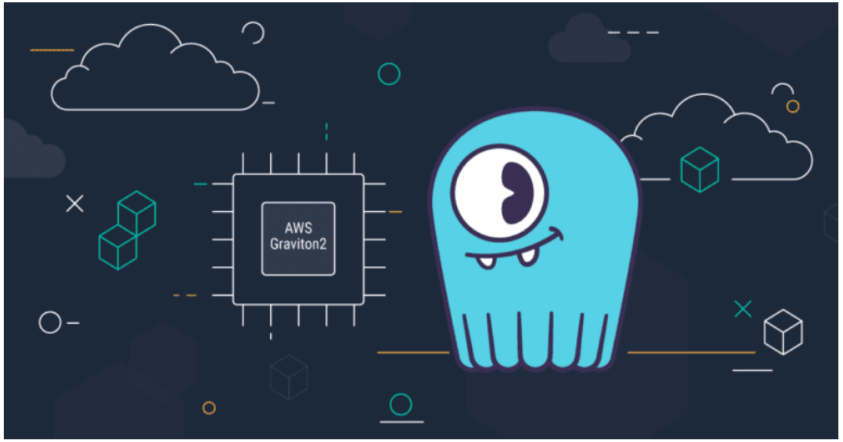
Best price performance for compute-intensive workloads

aws

See all blog posts

AWS Graviton2: Arm Brings Better Price-Performance than Intel

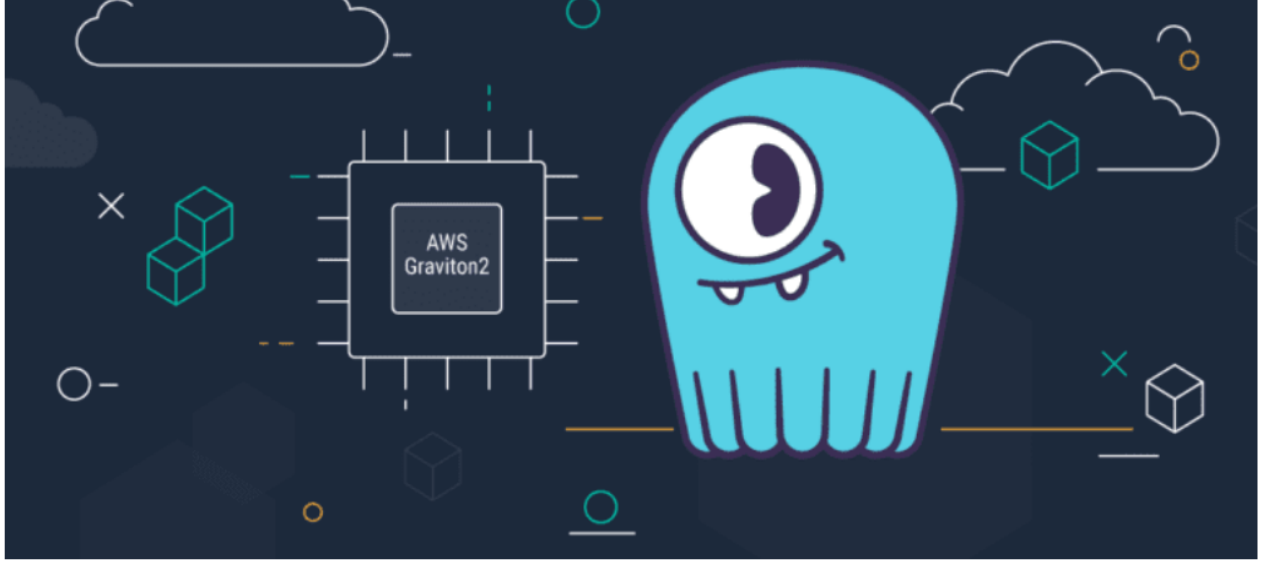
By Michał Chojnowski
September 16, 2021



Since the last time we took a look at ScyllaDB's performance on Arm, its expansion into the desktop and server space has continued: Apple introduced its M1 CPUs, Oracle Cloud added Ampere Altra-based instances to its offerings, and AWS expanded its selection of Graviton2-based machines. So now's a perfect time to test Arm again — this time with SSDs.

Summary

We compared ScyllaDB's performance on m5d (x86) and m6gd (Arm) instances of AWS. We found that **Arm instances provide 15%-25% better price-performance**, both for CPU-bound and disk-bound workloads, with similar latencies.



Since the last time we took a look at ScyllaDB's performance on Arm, its expansion into the desktop and server space has continued: Apple introduced its M1 CPUs, Oracle Cloud added Ampere Altra-based instances to its offerings, and AWS expanded its selection of Graviton2-based machines. So now's a perfect time to test Arm again — this time with SSDs.

Summary

We compared ScyllaDB's performance on m5d (x86) and m6gd (Arm) instances of AWS. We found that **Arm instances provide 15%-25% better price-performance**, both for CPU-bound and disk-bound workloads, with similar latencies.

Health Monitor

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"
}]
```

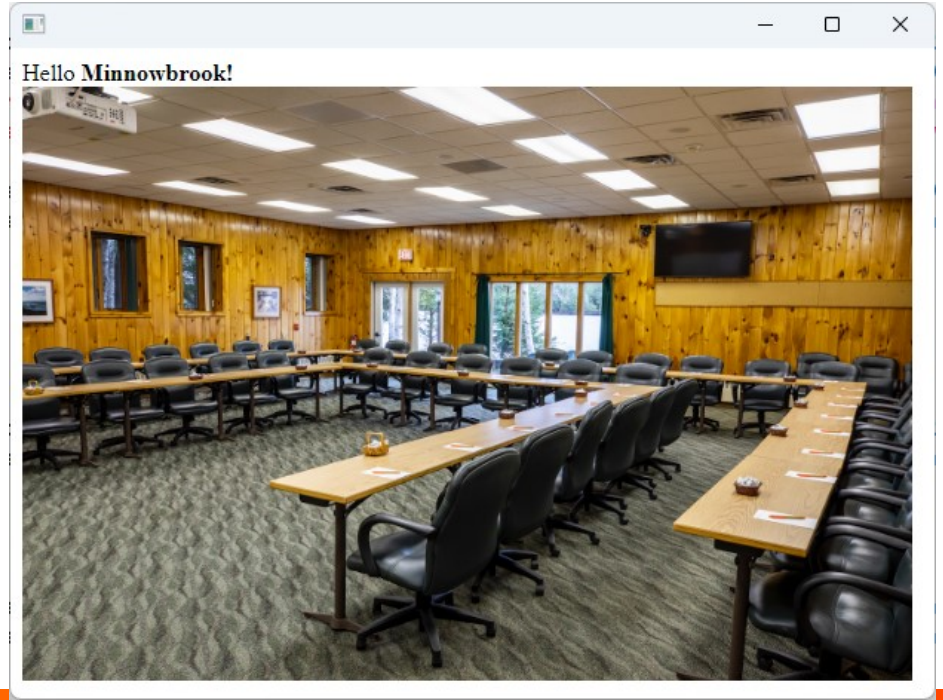
HTMLRenderer updates

New features include:

- AllowContextMenu
- 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>')
```



Chromium Embedded Framework (CEF)

HTMLRenderer updates

New features include:

- AllowContextMenu
- Get/SetZoomLevel
- IsLoading + LoadEnd event

Version 19.0 (March 2024)

- Platform Support / Distribution
  - 64-bit ARM support
    - New Macs, Pi 4&5, AWS Graviton
  - Enhanced .NET Bridge
    - Framework vs new .NET versions
  - Bound executables on all platforms
- Building Production Systems
  - Token range reservation
  - WS FULL handling
  - NCOPY/NMOVE callbacks
- Developer Productivity / IDE
  - Source "as typed" by default
  - Multi-line input on by default
  - HTMLRenderer updates
  - Link 4.0: Support for text data
  - HttpCommand client, Jarvis web service
- Installing & Managing APL
  - Multiple session files
  - Health Monitor

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'
```

```
('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]
```

```
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
```

```
([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])
```

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

```
[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

Token-by-token Debugging

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

Token-by-token Debugging

Debugger

Tools View

Left Argument

```
<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'
```

Function Pos: 4/24,0

Current Function

```
⊢ ⍪ ⌿ ÷ ≢
+
```

Right Argument

```
 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
```

SIstack (Tid: Tid:0)

Welcome to Dyalog Videos Page

dyalog.tv/Dyalog23/?v=CohsPaCIh4s

Apps | Link | APL | Flying & Sailing | Car | Dyalog | Cloud | SBO | Travel | Linux | Sport | Productivity | Ferie 2022

DYALOC

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

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

Watch Later | Share

More videos

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

John Daintree

Dyalog Version 20.0 – Part 1 — John Daintree

Getting and Setting 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 (design only, implementation probably not until v21)
- Strategy: Support .NET well, but DO NOT depend on it!

HTMLRenderer Enhancements

Suggested:

- File Upload
- Modal HTMLRenderers
- Other changes driven by EWC project

Easy Web Creator - EWC

- A JavaScript emulation of our Win32 layer (□WC, □WG, □WS ...)

JavaScript JWC

Function Table

File Colours

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

Average Score: 12.5

- Q1
- Q2
  - Apr
  - May
  - Jun

10 × Calc ********

	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	

JSWC

localhost:22322

Initialise

File Colours

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

Average Score: 12.5

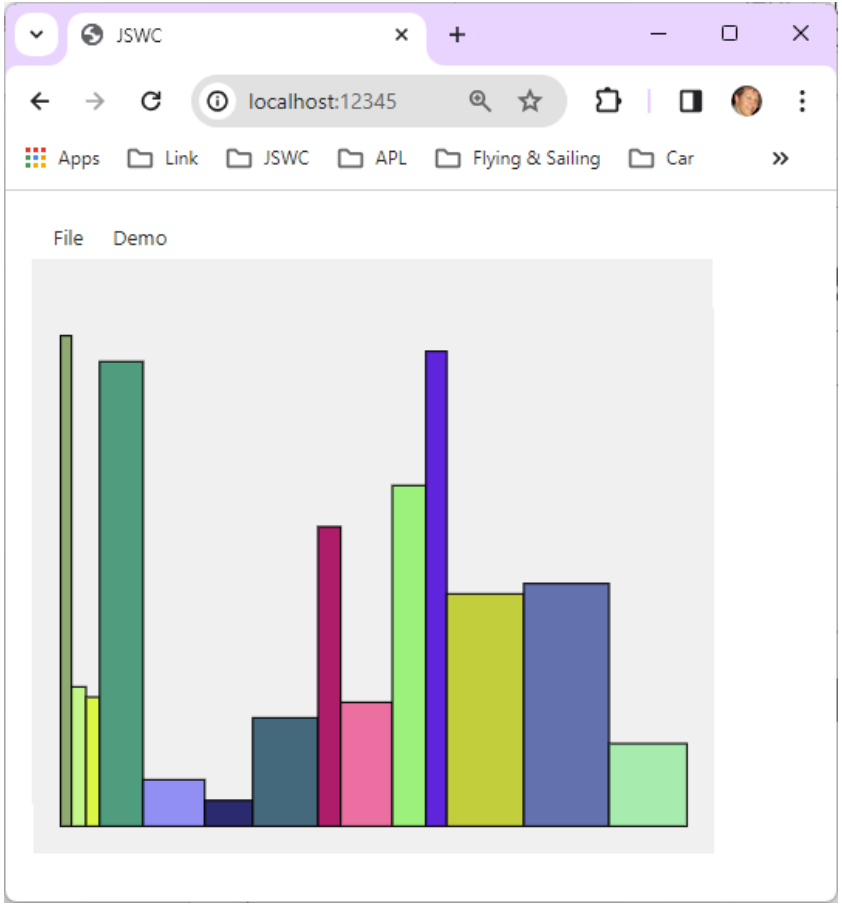
- Q1
- Q2
  - Apr
  - May
  - Jun

10 × Calc ********

	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 + ΔWI

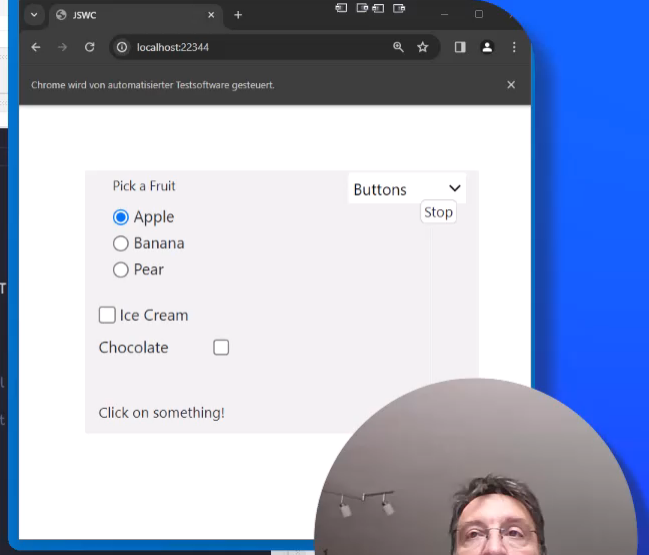
- EWC will support all ▯WC features used by ΔWI
- The resulting GUI will run under Linux, macOS
  - … and in a Browser on any platform



Automated Testing w/ Selenium

```
    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::0 ⋄ c←#.S.Find ⍵ ⋄ c.Selected∘c.Click}¨addon_ids  ⍝ make sure all options are unselected
        :If fruit≠''
            z←#.S.Click fruit
            ⎕DL 1   ⍝ a 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::0 ⋄ 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   ⍝ a 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
```



Set and Get 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

⎕NS: Name Set

```
      FirstName←'Jill' ⋄ Age←32

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

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

      FirstName Age
Jill  32

⍝ Left argument can be a namespace reference
```

⎕NG: Name Get

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

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

⍝ Left argument can be a namespace reference
```

Welcome to Dyalog Videos Page

dyalog.tv/Dyalog23/?v=eya2FlHinHE

Apps | Link | APL | Flying & Sailing | Car | Dyalog | Cloud | SBO | Travel | Linux | Sport | Productivity | Ferie 2022

DYALOC

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

Watch Later | Share

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

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

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

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

avg.apls

Fil Rediger Vis

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

Ln 2, Col 19 100% Unix (LF) UTF-8

Windows PowerShell

```
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>
```

⎕SHELL to replace existing ⎕SH

Invoke OS commands from APL

- Interruptible
- Optionally return data as an asynchronous "Stream"
- Manage stdin, stdout & stderr
- 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

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