

DYALOG

2020



## News from Dyalog

*Gitte Christensen, CEO Dyalog Ltd  
APL Germany. November 2020*

## Interesting (and busy) year

- Working from home
- Virtual meetings
- Business as usual

On a personal level:

- Haven't travelled since January
- Seen a lot of my grandchildren
- ... but I miss my Friends

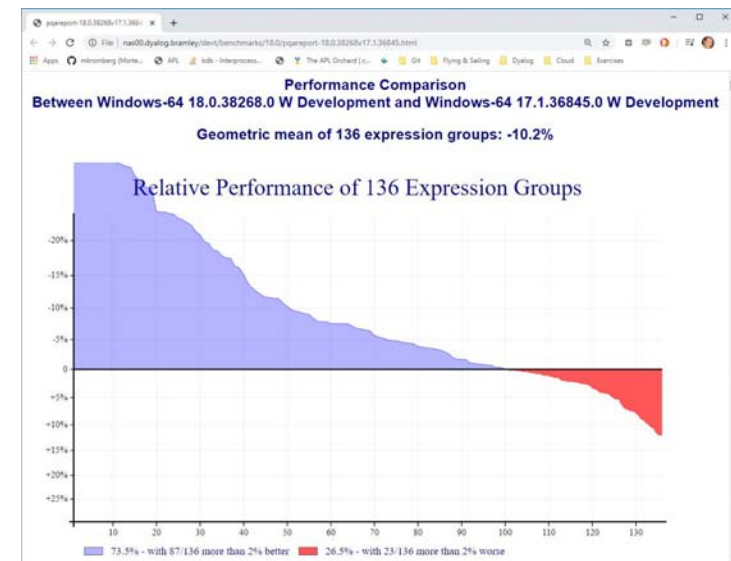
## Business as usual

- ◆ APL – the Tool of Thought
- ◆ - and the mountain



# Version 18.0 Language Engine

- New Primitive Operators
  - ∞ Constant
  - ∞ Atop
  - ∞ Over
- New Primitive Function
  - ≠ Unique Mask
- Integer arguments to
  - l Where
  - c Partition
- Significant Speed-ups



## New

- `□C` Case convert
- `fög` Over
- `fög` Atop
- `≠Y` Unique mask
- `A~` Constant
- `□DT` Date-time
- `1200⌈` Format date-time

## Improved

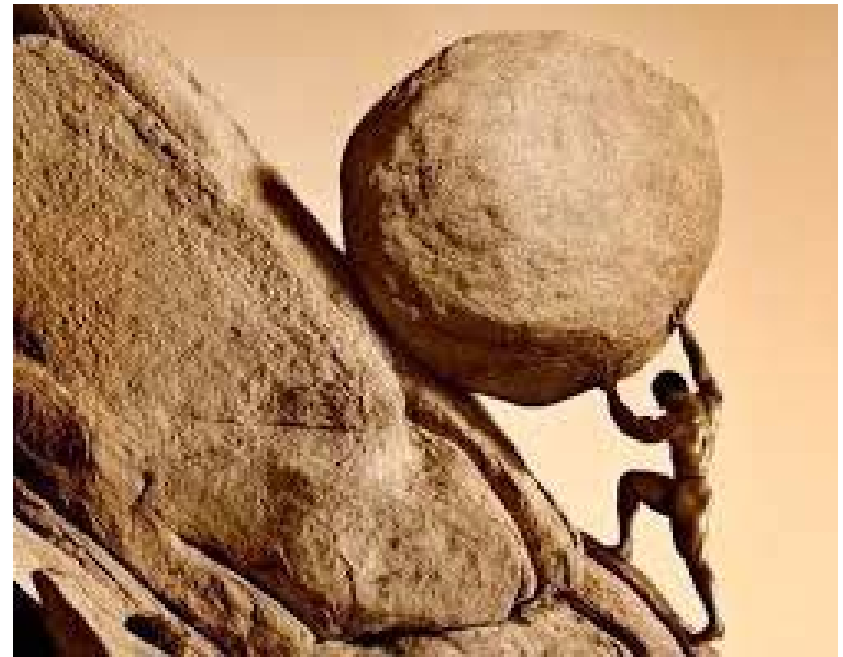
- `□JSON⌈ 'HighRank'`
- `□JSON⌈ 'Dialect'`
- `□INPUT⌈ 'NEOL'`
- `⌊Y`
- `X<Y`
- `↑[k]Y`

See Adams five excellent webinars on version 18.0 Language features



## Revisiting the Mountain

- ◆ Many of our paying customers are professional software developers with commercial applications
- ◆ - and we still insist that it be possible for knowledgeable people with good ideas to develop applications in APL



## Revisiting the Mountain

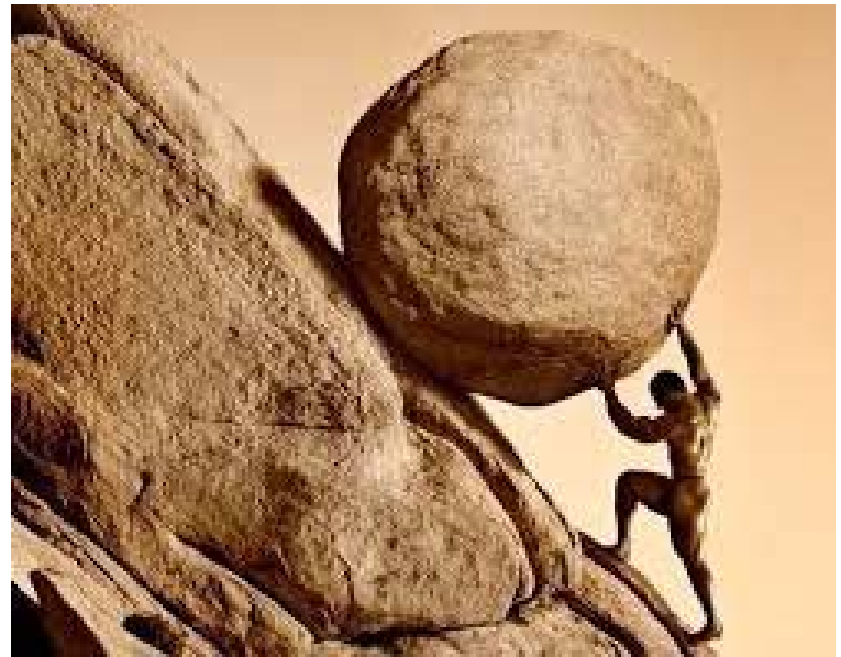
- There is a major paradigm shift right now
- Cloud deployment and Browser-based UI are winning
- The Tool Chains are still evolving



Worlds first electric car  
1888 Flocken Electrowagen  
(~Current state of Web UI)

## Revisiting the Mountain

- ◆ There are a lot of areas that require renewed attention
  - ◆ Cross platform development
  - ◆ Web / Portable User Interfaces
  - ◆ Deployment
  - ◆ Development Tools
  - ◆ APL Training





# Cross platform development

- ◆ .NET Core Bridge
- ◆ Configuration files

Download .NET (Linux, macOS, ...)

dotnet.microsoft.com/download

Microsoft | .NET About Learn Architecture Docs Downloads Community Get Started All Microsoft

Home > Download

.NET Preview Want to try out the latest preview? .NET 5.0.0-preview.3 is available. [Get .NET Preview >](#)

# Download .NET

Downloads for .NET Framework and .NET Core, including ASP.NET and ASP.NET Core

Not sure where to start? See the [Hello World in 10 minutes tutorial](#) to install .NET and build your first app.

**Windows** Linux macOS Docker

## .NET Core

### .NET Core 3.1

.NET Core is a cross-platform version of .NET for building websites, services, and console apps.

Run Apps [Download .NET Core Runtime](#)

Build Apps [Download .NET Core SDK](#)

Advanced [All .NET Core downloads...](#)

## .NET Framework

### .NET Framework 4.8

.NET Framework is a Windows-only version of .NET for building any type of app that runs on Windows.

Run Apps [Download .NET Framework Runtime](#)

Build Apps [Download .NET Framework Dev Pack](#)

Advanced [All .NET Framework downloads...](#)

.net core zip at DuckDuckGo

duckduckgo.com/?q=.net+core+zip&atb=v108-1&ia=web

Apps: mkromberg (Morte... APL kdb - Interprocess... The APL Orchard | c... Git Flying & Sailing Dyalog Cloud Exercises

.net core zip

Privacy, simplified.

All Images Videos News Maps Settings

Denmark Safe Search: Moderate Any Time

### ZipFile Class (System.IO.Compression) | Microsoft Docs

<https://docs.microsoft.com/en-us/dotnet/api/system.io.compression.zipfile>

Provides static methods for creating, extracting, and opening **zip** archives. This example shows how to create and extract a **zip** archive by using the ZipFile class. It compresses the contents of a folder into a **zip** archive, and then extracts that content to a new folder. To use the ZipFile class, you must reference the System.IO.Compression ...

### How to: Compress and extract files | Microsoft Docs

<https://docs.microsoft.com/en-us/dotnet/standard/io/how-to-compress-and-extract-files>

The following example shows how to create and extract a compressed **.zip** file by using the ZipFile class. The example compresses the contents of a folder into a new **.zip** file, and then extracts the **zip** to a new folder. To run the sample, create a start folder in your program folder and populate it with files to **zip**.

### Download .NET Core (Linux, macOS, and Windows)

<https://dotnet.microsoft.com/download/dotnet-core>

**Core** is a cross-platform version of **.NET**, for building apps that run on Linux, macOS, and Windows. This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use.

# Under Windows, macOS or Linux

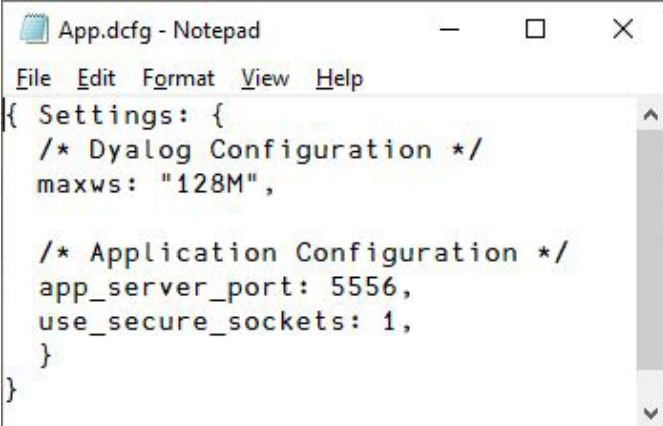
## File Compression

```
□ USING← 'System. IO. Compression, System. IO. Compression. ZipFile'  
folder← '/folder/to/be/zipped'  
zipfile← '/bla/bla/zipped.zip'
```

```
ZipFile.CreateFromDirectory folder zipfile
```

## Version 18.0: Configuration Files

- ◆ Identical across platforms
- ◆ Easily readable & editable
- ◆ Cascading configuration files provide flexible configuration of
  - ◆ each application
  - ◆ each version of APL
  - ◆ each user



```
App.dcfg - Notepad
File Edit Format View Help
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

  /* Application Configuration */
  app_server_port: 5556,
  use_secure_sockets: 1,
}
}
```

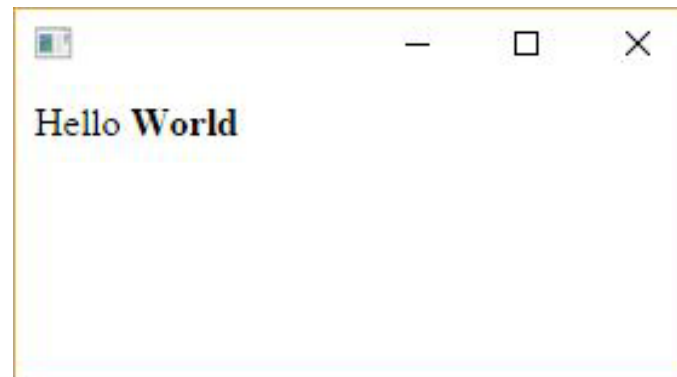
# User Interfaces



## What is the HTMLRenderer again?

Allows presentation of HTML/JS UI/Documents on Windows, macOS and Linux:

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



## What about [Web] User Interfaces?

- ◆ Web frameworks are in a constant state of churn
  - ◆ ASP.NET, Ext JS, React, Angular, Vue, Ember, Svelte ...
  - ◆ Online discussions reveal little developer satisfaction with the current state of the art
- ◆ Our own MiServer has not been widely adopted outside our own team
- ◆ We have not cracked this nut



# Deployment

- ◆ Source Files
- ◆ Jarvis
- ◆ Conga

## Version 18.0 - Deployment

- ◆ Launch on APL Source Files
- ◆ Configuration Files
- ◆ Improved support for Docker Containers
- ◆ Experimental Distribution of NuGet packages

## Launching APL on Source Files

- ◆ Pre-18.0 interpreters can be launched on
  - ◆ a binary workspace (.dws)
  - ◆ a "dyalog application file" (.dyapp)
- ◆ Version 18.0 allows ANY APL source file - with special treatment of
  - ◆ Functions (.aplf)
  - ◆ Namespaces (.apln)
  - ◆ Classes (.aplc)

# Jarvis

Jarvis makes your APL code available as web services

- ◆ Deploy APL code as RESTful or HTTP/JSON services
  - ◆ Combines and replaces JSONServer and RESTServer
- ◆ Can also replace MiServer for serving static HTTP
- ◆ Working on integrated WebSocket publish/subscribe
  - ◆ Think: socket.io

*\*Every\** new project we have seen recently uses Jarvis.

- ◆ Public repository at <https://github.com/Dyalog/Jarvis>

Json  
And  
Rest  
servIS

## Conga (our TCP/UDP framework)

We are preparing a set of Open-Source template projects to be delivered with Dyalog v19.0:

- ◆ Running APL applications as services / daemons under Windows and Linux
- ◆ Deploying and operating secure and load-balanced solutions in Dyalog APL
- ◆ Continuous integration, generating containers from APL code in GitHub, deploying to the cloud

As part of this effort, we intend to publish performance benchmarks for the core components



## Conga Performance

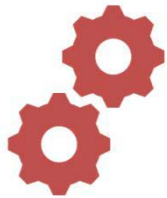
- ◆ Conga clients or servers are core component for nearly all new APL applications
- ◆ Benchmarking Conga has led to the discovery of several opportunities for speed-ups in Conga and the  $\square$ NA interface.
- ◆ In some cases, Conga is significantly faster
  - ◆ >Now 6000 transactions/sec on a single thread



# Development Tools



## Version 18.0 Development Tools



Experimental  
Multi-Line Input



Link 2.0

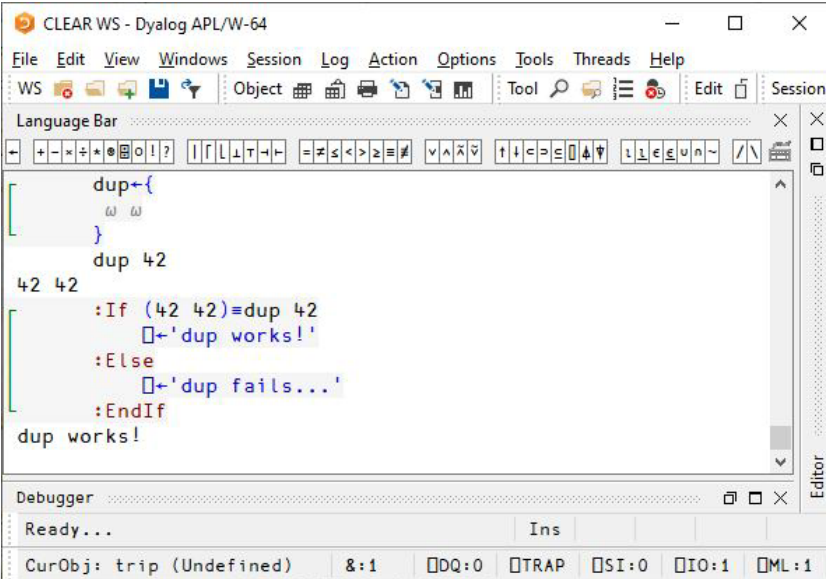


RIDE 4.3



# Multi-Line Input

- Experimental in v18.0, with `DIALOG_LINEEDITOR_MODE=1`



The screenshot shows the Dyalog APL/W-64 editor window. The title bar reads "CLEAR WS - Dyalog APL/W-64". The menu bar includes File, Edit, View, Windows, Session, Log, Action, Options, Tools, Threads, and Help. The toolbar contains various icons for file operations, editing, and debugging. The Language Bar shows standard APL symbols. The main editor area contains the following code:

```
dup+{
  ω ω
}
dup 42
42 42
:If (42 42)=dup 42
  ⎕←'dup works!'
:Else
  ⎕←'dup fails...'
:EndIf
dup works!
```

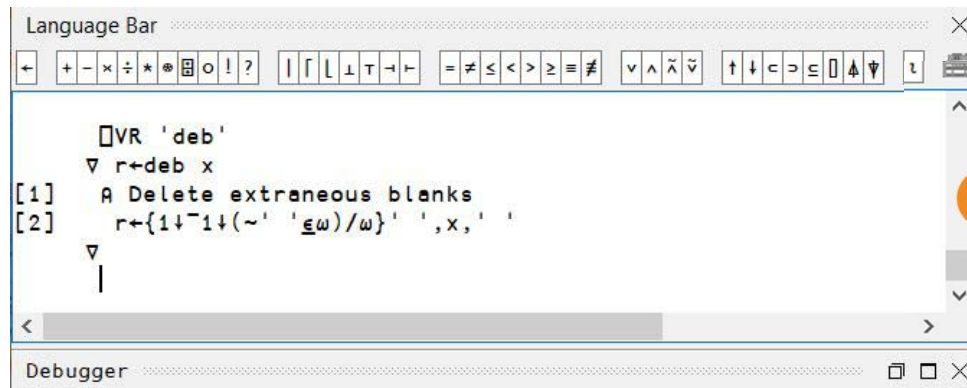
The status bar at the bottom shows "Ready...", "Ins", and "CurObj: trip (Undefined) &:1 □DQ:0 □TRAP □SI:0 □IO:1 □ML:1".

Supports:

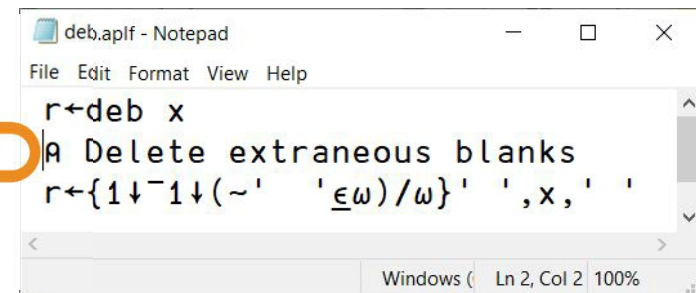
- Multi-line dfns
- Tradfns (replaces DEL editor)
- Control Structures



## What is Link?



```
Language Bar
[VR 'deb'
  r←deb x
  [1] A Delete extraneous blanks
  [2] r←{1↓~1↓(~' '⊆ω)/ω}' ',x,' '
  ]
]
Debugger
```



```
deb.aplf - Notepad
File Edit Format View Help
r←deb x
A Delete extraneous blanks
r←{1↓~1↓(~' '⊆ω)/ω}' ',x,' '
Windows ( Ln 2, Col 2 100%
```

- ◆ Each code item in the active workspace is **linked** to a file
- ◆ If the object is edited, the file is updated
- ◆ If the file is changed, the workspace is updated



# Array Notation

Could be mentioned in almost all categories

- ◆ Will become a core language feature
- ◆ Will make it easier to
  - ◆ Write (and read) code
  - ◆ Define and edit data in the APL session
    - ◆ Or in external editors
  - ◆ Effectively Use source code management systems
  - ◆ Configure and deploy systems
- ◆ Proposals refined for 5 years
  - ◆ Planning a final round of community review

**14:30 Adám Brudzewsky**  
Array Notation RC1

```
colours←[
  'red'   (255  0  0)
  'orange' (255 165 0)
  'purple' (128  0 128)
  'green'  (  0 255  0)
  'blue'   (  0  0 255)
  'gray'   (128 128 128)
]
```

# APL Training



# Teaching and Documentation

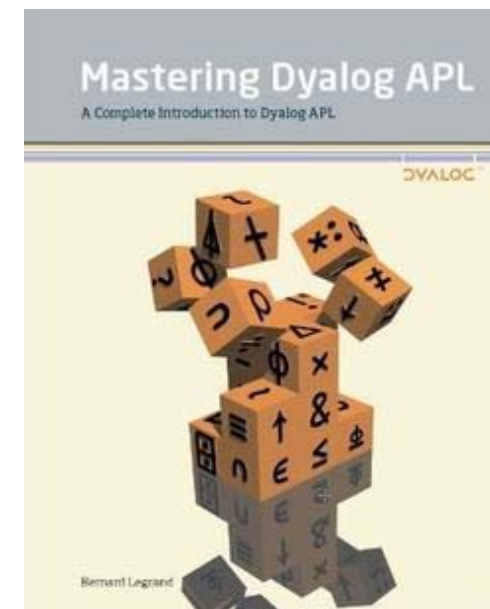
- ◆ Before:
  - APL was taught at work
  - Mentor – Apprentice
  - Physical courses
- ◆ Today:
  - Everybody looks to the internet for inspiration, webcasts, documentation and help

# Teaching and Documentation

- ◆ We are working hard at:
  - ◆ Producing on-line material and making it widely available
  - ◆ Making our website more inviting and informative for newcomers
  - ◆ Enhancing the chances of coming across APL as a choice for complex solutions
  - ◆ Making our documentation better suited for on-line searching and digestion

# Documentation and Training Materials

- Reworked Mastering Dyalog APL
- Example Projects
  - Running APL applications as services / daemons under Windows and Linux
  - Deploying and operating secure and load-balanced solutions in Dyalog APL
  - Continuous integration, generating containers from APL code in GitHub, deploying to the cloud
- Maintain high frequency of webinars
  - And webcasts
- New and improved TryAPL





TryAPL Version 3.0.1 (Dyalog version 18.0)  
 Tue Nov 24 2020 05:30:39  
 Copyright (c) Dyalog Limited 1982-2020

```

i10
1 2 3 4 5 6 7 8 9 10

```

---

**Got a minute? — Try APL!**

APL is an array-oriented programming language that will change the way you think about problems and data. With a powerful, concise syntax, it lets you develop shorter programs that enable you to think more about the problem you're trying to solve than how to express it to a computer.

TryAPL runs on Dyalog, which you can [download for free](#), or try it now by entering an expression (use the language bar above to type the special APL symbols), or clicking one of these expressions, followed by **Enter**, to see it in action:

<code>2 + 2</code>	No points for guessing this
<code>4 2 3 + 8 5 7</code>	Functions apply to arrays

---

<code>i10</code>	Generate the first ten integers
<code>+/i100000</code>	Sum the first 100 000 integers
<code>×/i10</code>	A long, slow way to write <code>!10</code>

---

<code>Avg+{(+/ω)÷#ω}</code>	Average is the sum divided by the count
<code>Avg 1 6 3 4</code>	... and apply it

---

<code>throws+?10000p6</code>	Store 10 000 dice throws
<code>+/1=throws</code>	Of 10 000 throws, how many 1s?
<code>+/i6°. =throws</code>	Frequency of all 6 possibilities

---

<code>'Hello, World!'</code>	Not just about maths!
<code>{α,#ω}⊖'Mississippi'</code>	See?

**What can APL do for you?**

Are you a Problem Solver (a domain or subject matter expert with problems to solve) or a Programmer (someone who translates those solutions into a computer-executable format)? Problem Solvers benefit from APL's ability to concisely express advanced concepts without getting bogged down with a lot of computerese syntax. Programmers benefit from APL's productivity and brevity. Shorter programs means quicker development time and less code to maintain and debug.

In either case, concepts that take several lines of code in other languages can often be reduced to a few characters in APL. You'll quickly find that APL's symbols are organized in a logical,

## Summary

- ◆ So yes – we are still busy
- ◆ We always look forward to telling you about it
- ◆ We would have liked to share a good meal and a beer with you – but sadly not this year