

# DYALOC

2020



## The Road Ahead

*Morten Kromberg*  
*CTO, Dyalog Ltd.*

# Destinations

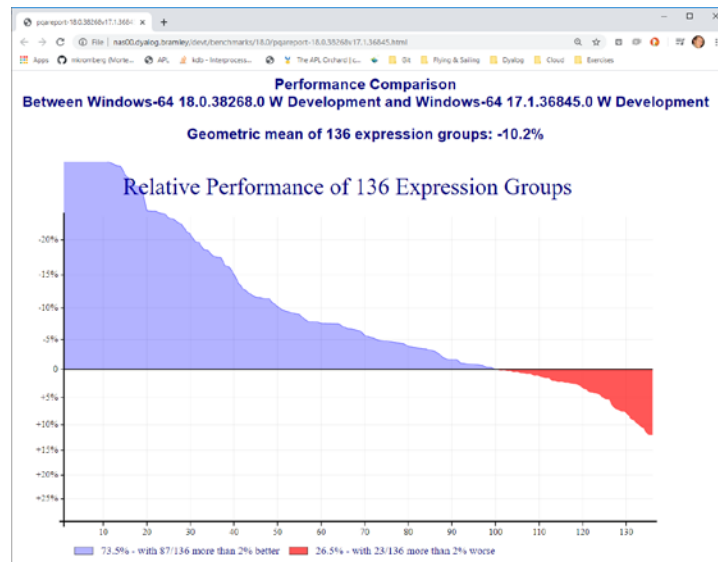
- Language Engine
- Interfaces
- Application Components
- Development Tools
- Deployment Mechanisms
- Documentation and Training Materials



# Destination Language Engine

# Version 18.0 Language Engine

- New Primitive Operators
  - ~ Constant
  - ÷ Atop
  - ö 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'  
 □R/□S '\f&' : 'Regex'  
 □NPUT: 'NEOL'  
zY  
 X<Y  
 ↑[k]Y



## New

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

## Improved

The Good:

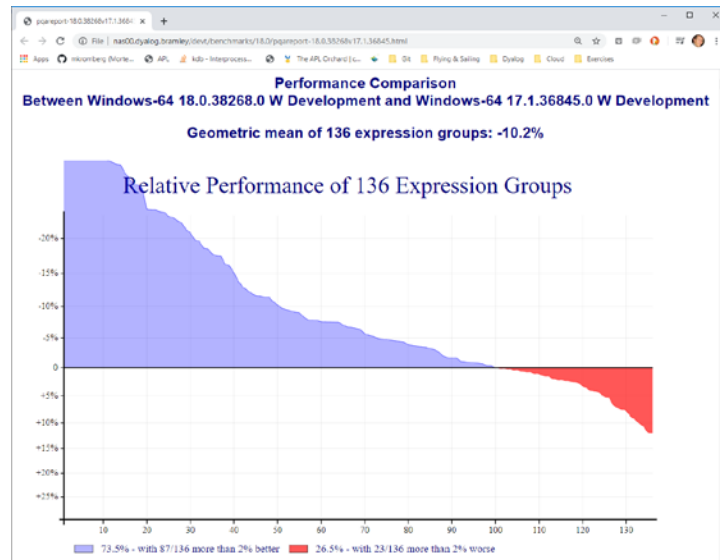
See Adam's five excellent webinars on version 18.0 Language features

`±Y`  
`X<Y`  
`↑[k]Y`



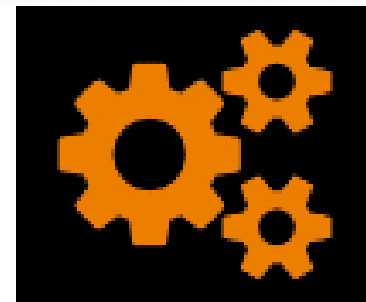
# V18: The Bad and The Ugly...

- Some of the performance work behind the "Blue Tsunami" was insufficiently reviewed & tested
- After the release of v18.0, bugs were found in special cases of widely used primitives
  - Special enough not to be caught by our tests
  - Common enough to be dangerous
- We issued a "caution" against the use of v18.0 in production environments, which we expect to withdraw in November.



# Version 19.0 Language Engine

- No new primitive features are planned
- We will continue with
  - Code review of recent optimisations
  - Completing code coverage of ALL primitive functions
- **Further optimisation work will await the completion of the above**
- With immediate effect, all code changes to existing primitives require code review and 100% code coverage
  - Previously, peer review was only required for patches to supported versions





# Compilers

- We are considering withdrawing the experimental optimiser that is managed with 4001 (aka "Jay's Compiler")
  - Contact us if you want to talk us out of that idea
- Aaron Hsu is now working full time alongside the Dyalog interpreter team
  - Broadening the applicability of the co-dfns compiler, turning his PhD research project into a product
  - Follow (contribute to) his task list and progress at <https://github.com/Co-dfns/Co-dfns/issues>

# Destination Interfaces

# Version 18.0 Interfaces

- Variant Extensions to JSON
  - HighRank**: As JavaScript does not have arrays of rank  $>1$ , when exporting from APL, split any such arrays thereby making vectors of vectors (of vectors...).
  - Dialect**: Support JSON5 extensions which include comments and more user-friendly formatting.
- .NET Core Bridge
  - Over the next few years, the Microsoft.NET Framework is being replaced by an open source, portable platform which has temporarily been named ".NET Core" - available for Windows, macOS and Linux.



Download .NET (Linux, macOS, a...

dotnet.microsoft.com/download

Appsmkromberg (Morte...APLkdb - Interprocess...The APL Orchard | c...GitFlying & SailingDyalogCloudExercises

Microsoft.NET

AboutLearnArchitectureDocsDownloadsCommunity

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.

WindowsLinuxmacOSDocker

## .NET Core

### .NET Core 3.1

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

Run Apps ⓘ

Build Apps ⓘ

Advanced ⓘ

Download .NET Core Runtime

Download .NET Core SDK

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 ⓘ

Build Apps ⓘ

Advanced ⓘ

Download .NET Framework Runtime

Download .NET Framework Dev Pack

All .NET Framework downloads...



.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.

# .NET Core Usage Example 1

## File Compression - identical under Windows, macOS and Linux

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

```
System.IO.Compression.ZipFile.CreateFromDirectory folder zipfile
```





# Version 19.0 Interfaces

Talks@Dyalog'20  
Mon 15:45 John Daintree  
The .NET Core Bridge

- v18.0 supports *consumption* of .NET Core libraries
- v18.0 does NOT allow APL to *provide* .NET Core components
- The most important v19.0 Goals are
  - Exporting APL code as classes
  - Generating executables (a "first" for macOS and Linux)
- Stretch targets, which may slip to the next release
  - Better support for threading and "Async" patterns
  - Generic classes and methods
  - Ability to use both Framework + Core from the same process





# Destination Application Components

# Version 18.0 Application Components

- Case folding with `⊞C` (replaces 819⊞)

```

1 ⊞C 'Hello' (⊞NS'') (1 2 3)
HELLO #.[Namespace] 1 2 3

```

- Date / Time manipulation with `⊞DT`

```

1 ⊞DT c⊞TS
43949.56179
'__da__Dddd, DDoo mmmm YYYY; hh:mm:ss' (1200⊞) dt
Onsdag, 13. februar 2019; 10:16:56

```



# What is the HTMLRenderer again?

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

# What is the HTMLRenderer again?

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

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



# Recent HTMLRenderer Enhancements

## In Version 17.1 & 18.0

- Support for **WebSockets** between the APL session and the CEF client window.
- A **DoPopup** event that is triggered when the CEF client issues a request for a new window.
- A **SelectCertificate** event that is triggered on request for a resource that requires a certificate.
- A **ShowDevTools** method that will toggle the visibility of the Chromium Developer Tools.
- Support for several **WC** properties
  - Caption, SysMenu, MinButton, MaxButton, Sizeable and Moveable.
- Ability to **print to PDF**.
- An **ExecuteJavaScript** method.
- Much improved **documentation**.



# v19.0 HTMLRenderer Enhancements

Brian's wish list includes

- Allow `file://` in href and src attributes so resources can be used directly without simulated HTTP
- Support `<input type="file">` for file upload
- macOS: Give newly created windows the focus

Get in touch if you want to nominate additional features!



# Version 19.0 Application Components

- Non-linear Random Distributions
  - Some random distributions are difficult to generate accurately AND efficiently using APL code

**Talks@Dyalog'20**

**Mon 16:30 Ron Murray**

Time Travel Debugging and  
Statistical Distributions

# Version 19.0 Application Components

- Non-linear Random Distributions
- Jarvis – for RESTful or simple HTTP/JSON services



# Jarvis

If you have one or more APL functions, Jarvis can make them available as a web service

- Deploy APL code as a pragmatic HTTP/JSON service, or as a "RESTful" service if you master that paradigm
- Jarvis 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 I have looked at recently uses Jarvis.

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

J son  
And  
Rest  
serVIS

# Version 19.0 Application Components

- Non-linear Random Distributions
- Jarvis – for RESTful or simple HTTP/JSON services
- Conga Servers: Very significant speed-ups

# 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 will publish performance benchmarks for core components
  - ◆ We started with Conga ...



# Conga Performance

- Conga clients or servers are core components for nearly all new APL applications
- Benchmarking Conga 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



# v19.0 Application Components

Psst! Help me talk Richard into this one:

```
]disp emp
```

↓Name	Age	Zipcode
Bobby	21	30102
Sally	32	43001

```
([JSON @ 'Tables' 'RowsAsObjects') emp  
[{"Name":"Bobby","Age":21,"Zipcode":"30102"},  
 {"Name":"Sally","Age":32,"Zipcode":"43001"}]
```



# 3rd Party Application Components

## ◆ APL2XL

- ◆ Generate Excel spreadsheets w/out Excel or ANY external libraries (except DotNetCore or alternative compression software)
- ◆ Public repository: <https://github.com/the-carlisle-group/APL2XL>
- ◆ Complements <https://github.com/the-carlisle-group/XL2APL>

## ◆ qWC – For moving **qWC** applications to the web, to macOS and Linux

- ◆ (or moving **qWC** developers to build new applications on ... ☺)
- ◆ Provides **qWC** emulation (with extensions), projected into web browsers (or the HTMLRenderer) via web sockets.
- ◆ Proprietary solution developed by MJHSoftware

**Talks@Dyalog'20**

**Tue 16:30**

**Chris & Michael Hughes**

Building applications using qWC  
on the web.



# What about [Web] User Interfaces?

# What about [Web] User Interfaces?

- Web frameworks are in a constant state of churn
  - ASP.NET, Ext JS, React, Angular, Vue, Ember, Svelte ...
  - Nobody seems *\*really\** happy with the current state of the art
- MiServer has not been widely adopted outside Dyalog
  - We have not cracked this nut



# Solutions for Web User Interfaces?

- qWC may offer a solution for certain types of apps
- We must ensure that APL works well as a service
  - Jarvis allows the construction of APL-based engines that work well with web frontend tools
  - WebSocket support will complete the Jarvis package

# Destination Development Tools

# Version 18.0 Development Tools



Experimental  
Multi-Line Input



Link 2.0



RIDE 4.3

# Multi-Line Input

- Experimental in v18.0, enabled with an optional switch
- John will tell you all about it after the break



# Link



# Link


The screenshot shows the RStudio interface. The top toolbar includes icons for file operations, editing, and running code. The R console window displays the following output:

```

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

```

The message "A Delete extraneous blanks" indicates that the R console has automatically removed trailing spaces from the input. The second line shows the assignment of a string to variable 'r', which contains a complex expression involving a loop and a function call.



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

File

Edit

Selection

View

Go

Debug

Terminal

Help

GIT LENS

REPOSITORIES

Link master 251 01 • ~4 -1 • Last fetched 9:29pm...

master 251 01 ⇄ origin/master

Over a month ago

Handle listing links with no linked members ...

Don't init globals +0 ~1 -0 • abrudz, 2 months...

Exception trap +0 ~1 -0 • abrudz, 2 months ago

Merge branch 'master' of https://github.com...

Add general casing util +0 ~1 -0 • abrudz, 2 ...

Actually apply the case code, not just remov...

Fix #74 +0 ~1 -0 • abrudz, 2 months ago

Fixes #75: Src files incorrectly updated after )...

Fixes #77: FileSystemWatcher threading issu...

FileSystemWatcher.apln StartupS...

Fixed #76: Add "ViewMetaData" +1 ~0 -0 • Y...

... Show More Commits

25 commits behind

5 files changed

Compare master (working) with <branch, tag, ...

FILE HISTORY

LINE HISTORY

COMPARE

SEARCH COMMITS

FileSystemWatcher.apln (0f00180) ↔ FileSystemWatcher.apln (2e934c5) - Link - Visual Studio Code

FileSystemWatcher.apln (0f00180) ↔ FileSystemWatcher.apln (2e934c5)

Dev

Link

StartupSession

Link

FileSystemWatcher.apln

27

r+JNEW Disposable watcher A wrap a de

28

r.ODF 1φ'][,##.U.WinSlash>args

29

▼

30

:Class Disposable

31

:Field Public Object

32

▼ make ref

33

:Access Public

34

:Implements Constructor

35

Object+ref

36

▼

37

▼ do\_dispose ref

38

:Trap 0

39

ref.Dispose

40

:EndTrap

41

▼

42 -

▼ dispose;tid

43

:Implements Destructor

44

Object.EnableRaisingEvents+0

45 -

tid+do\_dispose&Object

46

▼

47

:EndClass

48

:EndNamespace

49

27

r+JNEW Disposable watcher A wrap

28

r.ODF 1φ'][,##.U.WinSlash>args

29

▼

30

:Class Disposable

31

:Field Public Object

32

▼ make ref

33

:Access Public

34

:Implements Constructor

35

Object+ref

36

▼

37

▼ do\_dispose ref

38

:Trap 0

39

ref.Dispose

40

:EndTrap

41

▼

42 +

▼ dispose You, 2 months a

43

:Implements Destructor

44

Object.EnableRaisingEvents+0

45 +

do\_dispose Object

46

▼

47

:EndClass

48

:EndNamespace

49

master\* 251 01 0 0 0

You, 2 months ago Ln 42, Col 1 Spaces: 2 UTF-8 with BOM APL



# Link Version 2.1 (Targeting Dyalog v19.0)


- Thorough testing
  - Several bug fixes
  - Improved argument validation
  - Many improved error messages
- Speedups & cleanups
  - More consistent API
  - Removing differences in behaviour when using built-in and external editors
- Supports "case coding"
  - Allow names which differ only in case (FOO vs foo) by using file names that encode the case information
- Improved support for array notation



# What is RIDE?

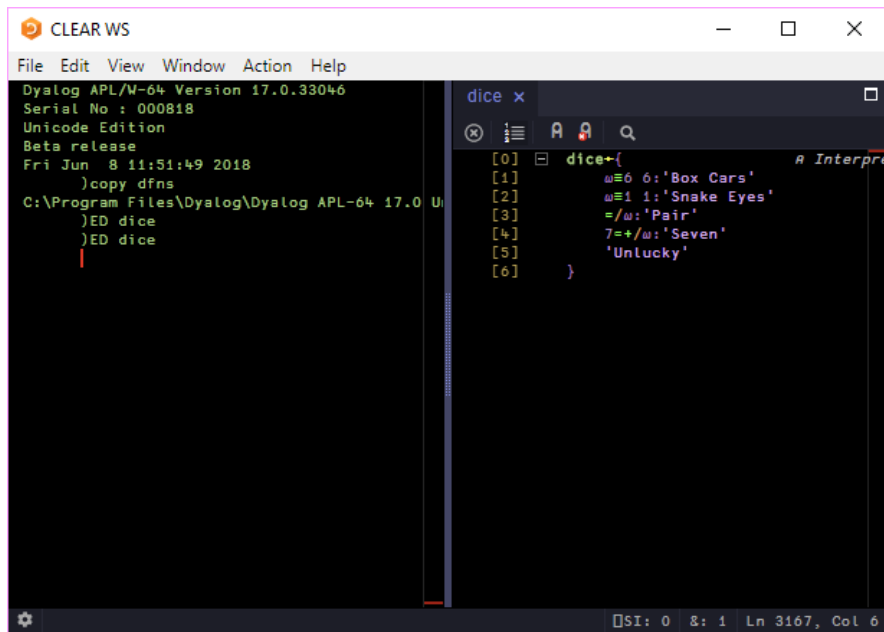
Call	CheckOpenEditors	Confirm	Connect	EditClub
EditUser	Error	FinishTour	GetSession	Init
JoinTour	ListSelector	Login	MakeClient	MakeEdit
MakePrompt	NewGuid	OpenTab	PlayGame	Progress
RegisterForm	Run	Status	SwitchToForm	TSFmtNice
UpdateUserPanel	UserActions	ViewClub	ViewTour	
WSMonitor	doBoardClick	doClubClick	doClubSave	
doEditFormCallback		doInvitation	doListSelector	doLogin
doPrompt	doRegister	doTogglePwds	doTourClick	
doUTree	doUpload	doUserActions	doUserSave	dtb
qNQ	qWC	qWD	qWG	qWN
			qWS	

72 minutes ago



```
[0] r←larg doLogin event;S;change;data;values;res;hos
[1] a Perform a login
[2]
[3] r←0
[4] Progress'Processing login...'
[5]
[6] (data values)←larg
[7]
[8] change←'Change'=6↑(event)qWG'Caption'
[9]
[10] S←GetSession''
[11]
[12] :If ~^(2↑values)≠''
[13]     Error'You must enter a userid and password'
[14] :EndIf
[15]
[16] :If change^≠/2↑values
[17]     Error'Passwords do not match'
[18]     →0
[19] :ElseIf Connect S((2+change)↑values)
[20]     'login'qWS'Visible' 0
[21]     'sbomain'qWS'Visible' 1
[22] :EndIf
```

## Example of a recent RIDE enhancement: Dark and light themes



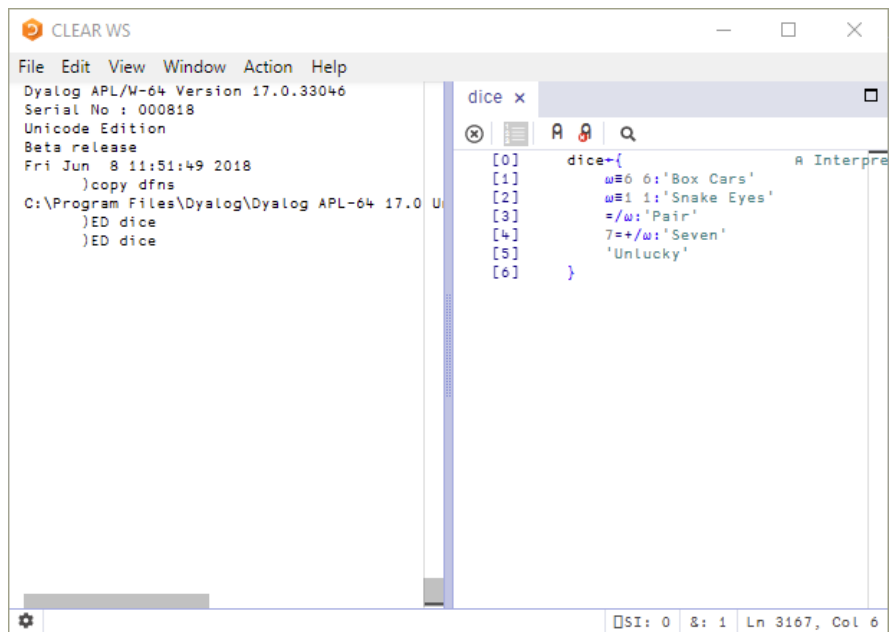
A screenshot of the Dyalog APL IDE running in dark theme. The window title is 'CLEAR WS'. The menu bar includes 'File', 'Edit', 'View', 'Window', 'Action', and 'Help'. The left pane shows the command history with the following text:

```
Dyalog APL/W-64 Version 17.0.33046
Serial No : 000818
Unicode Edition
Beta release
Fri Jun 8 11:51:49 2018
)copy dfns
C:\Program Files\Dyalog\Dyalog APL-64 17.0 U
)ED dice
)ED dice
```

The right pane shows a code editor with a file named 'dice x'. The code is as follows:

```
[0] dice←{
[1]   ω≡6 6:'Box Cars'
[2]   ω≡1 1:'Snake Eyes'
[3]   =/ω:'Pair'
[4]   7=+/ω:'Seven'
[5]   'Unlucky'
[6] }
```

The status bar at the bottom indicates 'SI: 0 &: 1 Ln 3167, Col 6'.



A screenshot of the Dyalog APL IDE running in light theme. The window title is 'CLEAR WS'. The menu bar includes 'File', 'Edit', 'View', 'Window', 'Action', and 'Help'. The left pane shows the command history with the following text:

```
Dyalog APL/W-64 Version 17.0.33046
Serial No : 000818
Unicode Edition
Beta release
Fri Jun 8 11:51:49 2018
)copy dfns
C:\Program Files\Dyalog\Dyalog APL-64 17.0 U
)ED dice
)ED dice
```

The right pane shows a code editor with a file named 'dice x'. The code is as follows:

```
[0] dice←{
[1]   ω≡6 6:'Box Cars'
[2]   ω≡1 1:'Snake Eyes'
[3]   =/ω:'Pair'
[4]   7=+/ω:'Seven'
[5]   'Unlucky'
[6] }
```

The status bar at the bottom indicates 'SI: 0 &: 1 Ln 3167, Col 6'.

# RIDE goals for Dyalog v19.0

- Tested and enhanced for debugging multi-threaded server applications
- More sensitive to the active version of APL
  - Language Bar, Online Help, Syntax Colouring of new features, ...
- Remember responses to confirmation prompts
- Support multi-line input
- Documentation / examples of secure RIDE configuration
  - E.g. how to require client-side certificates



# Version 19.0 Development Tools

- ✧ In addition to updates to RIDE and Link
- ✧ Finish multi-line input
  - ✧ (Requires RIDE support)
- ✧ Revise handling of I/O to facilitate
  - ✧ "#!" scripting  
([https://en.wikipedia.org/wiki/Shebang\\_\(Unix\)](https://en.wikipedia.org/wiki/Shebang_(Unix)))
  - ✧ Redirected I/O
  - ✧ Interpreter support for RIDE in headless environments

Talks@Dyalog'20

Mon 15:15 John Daintree  
Multi-line input and Scripting

# Emerging Tools built outside Dyalog



## Acre Desktop: Project management

<https://github.com/the-carlisle-group/Acre-Desktop>

- Created by Paul Mansour / The Carlisle Group

README.md



## Acre Desktop

### Introduction

Acre Desktop is an enhancement to the Dyalog APL IDE, delivered as a small set of user commands, that introduces the concept of a **project**, and stores code outside the workspace in plain text files.

It makes working with 3rd party source code management (SCM) systems like Git easy.

However, while one of Acre Desktop's main purposes is to facilitate SCM using text files, it has tremendous value as a simple addition to the IDE, eliminating many of the downsides of workspaces, while keeping all of the benefits.



#dy



# Emerging Tools built outside Dyalog

Finally! (We've been talking about this for years...)

## ⊗ Tatin: APL Package Manager

<https://tatin.dev>

- ◆ Tatin is built on Acre
- ◆ Developed by Kai Jaeger / APLTeam  
Co-funded by Dyalog
- ◆ Still a prototype with rudimentary documentation







# Tatin



# Tatin

IMPORTANT



# Tatin Registry

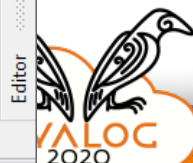
## Main page

This is the Tatin Registry. It holds packages designed to be used under Dyalog APL.

1. [List of all groups](#)
2. [List of all packages](#)
3. [List of all tags](#)
4. [Documentation Center](#)

# Tatin Demo

```
]tatin.ListPackages [tatin]
Group      Package Name      Major
-----
aplteam    OS                      3
aplteam    ShowChmHelp              3
aplteam    APLTreeUtils2            1
aplteam    CompareSimple            4
aplteam    WinSys                   5
aplteam    Logger                   5
aplteam    IniFiles                 5
aplteam    EventCodes               3
aplteam    Execute                  3
aplteam    DotNetZip                1
aplteam    FilesAndDirs             4
aplteam    HandleError              4
aplteam    CodeCoverage             0
aplteam    WinRegSimple             3
aplteam    WinReg                   5
aplteam    ServiceState             3
aplteam    WindowsEventLog         3
aplteam    Tester2                  3
```





The screenshot shows the APL IDE interface. The main window is the code editor, which contains the following APL code:

```

)ns myns
#.myns
  myns.⌊FX 'r+goo x' ':If x=1' 'r+'x is one'' ':Else' 'r+'x is not one'' ':EndIf'
  ⌊VR 'myns.goo'
  ▽ r+goo x
[1] :If x=1
[2]   r+'x is one'
[3] :Else
[4]   r+'x is not one'
[5] :EndIf
  ▽


]tatin.loadpackage [tatin]/aplteam-CodeCoverage #.MyPkgs
#.MyPkgs.CodeCoverage
C←NEW #.MyPkgs.CodeCoverage(,←#.myns')
C.filename←'/tmp/myns-coverage'
C.Start 0

  myns.goo 1
x is one

  C.Stop 0
  C.Finalize 0
/tmp/myns-coverage.dcf
  ⌊+1 #.MyPkgs.CodeCoverage.ProcessDataAndCreateReport C.filename
/tmp/myns-coverage.html

```

The bottom of the IDE shows a debugger window with the status "Ready..." and a toolbar with icons for various debugging actions. The status bar at the very bottom displays the current object as "CurObj: aplteam (Undefined)" and various system metrics.



The screenshot shows the APL385 IDE interface. The top menu bar includes File, Edit, View, Windows, Session, Log, Action, Options, Tools, and Help. Below the menu is a toolbar with various icons for file operations, editing, and debugging. The main window is titled 'Language Bar' and contains a code editor with the following APL code:

```

)ns myns
#.myns
myns.[FX 'r+goo x' ':If x=1' 'r+'x is one'' ':Else' 'r+'x is not one'' ':EndIf'
[VR 'myns.goo'
  ▽ r+goo x
[1] :If x=1
[2]   r+'x is one'
[3]   :Else
[4]   r+'x is not one'
[5]   :EndIf
  ▽

]tatin.loadpackage [tatin]/aplteam-CodeCoverage #.MyPkgs
#.MyPkgs.CodeCoverage
C+NEW #.MyPkgs.CodeCoverage(, <'#.myns')
C.filename+ '/tmp/myns-coverage'
C.Start 0

myns.goo 1
x is one

C.Stop 0
C.Finalize 0
/tmp/myns-coverage.dcf
[+1 #.MyPkgs.CodeCoverage.ProcessDataAndCreateReport C.filename
/tmp/myns-coverage.html

```

A red rectangle highlights the following code block:

```

  ▽ r+goo x
[1] :If x=1
[2]   r+'x is one'
[3]   :Else
[4]   r+'x is not one'
[5]   :EndIf
  ▽

```

Below the code editor is a 'Debugger' window with the text 'Ready...' and 'Ins'. At the bottom of the IDE, there is a status bar showing 'CurObj: aplteam (Undefined)' and various system metrics like '&:1', 'DQ:0', 'TRAP', 'SI:0', 'IO:1', and 'ML:1'.

File Edit View Windows Session Log Action Options Tools Threads Help

WS Object Tool Edit Session APL385 Unicode 16

Language Bar

```
)ns myns
#.myns
  myns.FX 'r+goo x' ':If x=1' 'r+'x is one'' ':Else' 'r+'x is not one'' ':EndIf'
  VR 'myns.goo'
  r+goo x
[1]   :If x=1
[2]   r+'x is one'
[3]   :Else
[4]   r+'x is not one'
[5]   :EndIf
  v
```

```
]tatin.loadpackage [tatin]/aplteam-CodeCoverage #.MyPkgs
#.MyPkgs.CodeCoverage
C+NEW #.MyPkgs.CodeCoverage(,c'#.myns')
C.filename+ '/tmp/myns-coverage'
C.Start 0

myns.goo 1
x is one

C.Stop 0
C.Finalize 0
/tmp/myns-coverage.dcf
+1 #.MyPkgs.CodeCoverage.ProcessDataAndCreateReport C.filename
/tmp/myns-coverage.html
```

NB: If CodeCoverage had any dependencies, they would also have been loaded

Debugger

Ready...

Ins

CurObj: aplteam (Undefined)

&amp;:1

DQ:0

TRAP

SI:0

IO:1

ML:1



#C







File Edit View Windows Session Log Action Options Tools Threads Help

WS Object Tool Edit Session APL385 Unicode 16

Language Bar

```
)ns myns
#.myns
  myns.FX 'r+goo x' ':If x=1' 'r+'x is one'' ':Else' 'r+'x is not one'' ':EndIf'
  VR 'myns.goo'
  r+goo x
[1]   :If x=1
[2]   r+'x is one'
[3]   :Else
[4]   r+'x is not one'
[5]   :EndIf
  r+
]tatin.loadpackage [tatin]/aplteam-CodeCoverage #.MyPkgs
#.MyPkgs.CodeCoverage
C+NEW #.MyPkgs.CodeCoverage(,c'#.myns')
C.filename+ '/tmp/myns-coverage'
C.Start 0

  myns.goo 1
x is one

C.Stop 0
C.Finalize 0
/tmp/myns-coverage.dcf
+1 #.MyPkgs.CodeCoverage.ProcessDataAndCreateReport C.filename
/tmp/myns-coverage.html
```

Debugger Ready... Ins

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





The screenshot shows the APL385 Unicode editor with the following code:

```

)ns myns
#.myns
  myns.⌈FX 'r+goo x' ':If x=1' 'r+'x is one'' ':Else' 'r+'x is not one'' ':EndIf'
  ⌈VR 'myns.goo'
  ▽ r+goo x
[1] :If x=1
[2]   r+'x is one'
[3] :Else
[4]   r+'x is not one'
[5] :EndIf
  ▽

]tatin.loadpackage [tatin]/aplteam-CodeCoverage #.MyPkgs
#.MyPkgs.CodeCoverage
C+⌈NEW #.MyPkgs.CodeCoverage(, <'#.myns')
C.filename+ '/tmp/myns-coverage'
C.Start 0

  myns.goo 1
x is one

  C.Stop 0
  C.Finalize 0
/tmp/myns-coverage.dcf
  ⌈+1 #.MyPkgs.CodeCoverage.ProcessDataAndCreateReport C.filename
/tmp/myns-coverage.html

```

The code is written in APL and uses Unicode characters for comments and formatting. The code defines a function 'myns' that takes an argument 'x' and returns a string 'x is one' or 'x is not one' based on the value of 'x'. The code also includes a section for generating coverage reports, which is highlighted with a red box. The interface includes a menu bar, a toolbar, and a status bar.



```

)ns myns
#.myns
  myns.⌈FX 'r+goo x' ':If x=1' 'r+'x is one'' ':Else' 'r+'x is not one'' ':EndIf'
  ⌈VR 'myns.goo'
  ▽ r+goo x
[1]   :If x=1
[2]     r+'x is one'
[3]   :Else
[4]     r+'x is not one'
[5]   :EndIf
  ▽

]tatin.loadpackage [tatin]/aplteam-CodeCoverage #.MyPkgs
#.MyPkgs.CodeCoverage
C+⌈NEW #.MyPkgs.CodeCoverage(,c'#.myns')
C.filename+ '/tmp/myns-coverage'
C.Start 0

  myns.goo 1
x is one

  C.Stop 0
  C.Finalize 0
/tmp/myns-coverage.dcf
  ⌈+1 #.MyPkgs.CodeCoverage.ProcessDataAndCreateReport C.filename
/tmp/myns-coverage.html

```

Debugger Ready... Ins

CurObj: aplteam (Undefined) &:1 DDQ:0 DTRAP DSI:0 DIO:1 DML:1

# Coverage Report

Watched: 1 fns/opr within #.goo

Overall 67% of the testable code is covered.

(Comment lines, empty lines, all :End\* lines etc. are all ignored)

0 of the fns/opr are 100% covered.

1 are partly covered:

Function/Operator	Lines not executed	Coverage	≠
<a href="#">#.goo.goo</a>	4	67%	3

## Listings

#.goo.goo

```
r←goo x
[1] :If x=1
[2]   r←'x is one'
[3] :Else
→[4]   r←'x is not one'
[5] :EndIf
```

# Please have a taste!

- Comment
- Contribute packages!
- Dyalog will ponder whether language extensions (for example to `□USING`) would make it easier to consume packages



# Destination Deployment

# Version 18.0 - Deployment

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



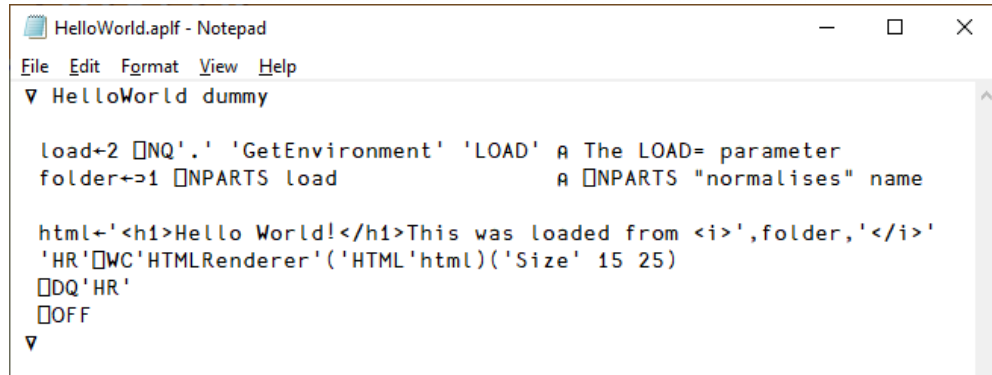


# Launching APL on Source Files



- 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.  
It will automatically run:
  - Functions (.aplf)
  - Namespaces (.apln)
  - Classes (.aplc)(for other types of source files, you must also specify LX=)

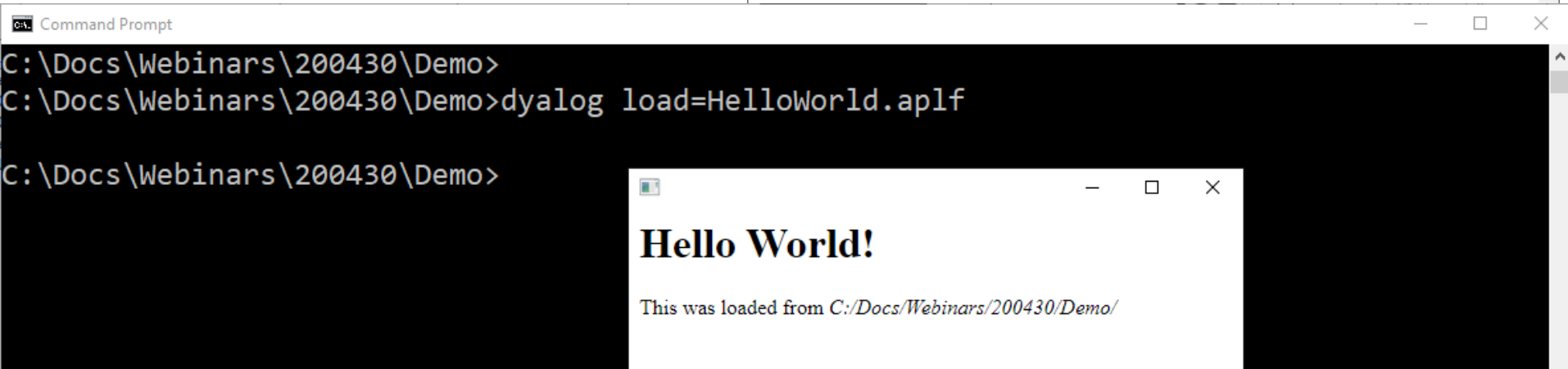
# Launching APL on Source Files



```
File Edit Format View Help
▼ HelloWorld dummy

load+2 □NQ'.' 'GetEnvironment' 'LOAD' ⌈ The LOAD= parameter
folder↔1 □NPARTS load ⌈ NPARTS "normalises" name

html+ '<h1>Hello World!</h1>This was loaded from <i>',folder,'</i>'
'HR'□WC'HTMLRenderer'('HTML'html)('Size' 15 25)
□DQ'HR'
□OFF
▼
```



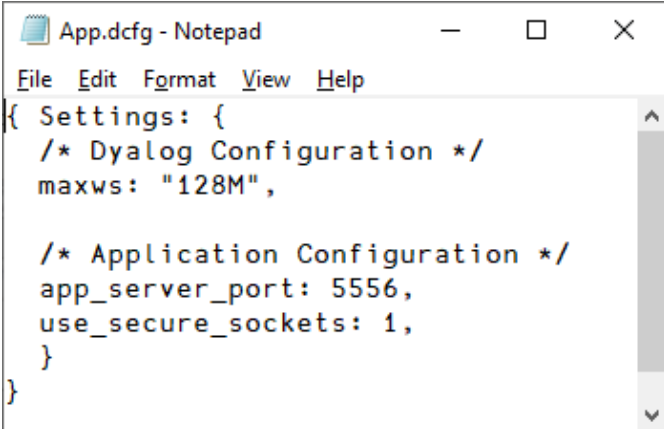
```
C:\Docs\Webinars\200430\Demo>
C:\Docs\Webinars\200430\Demo>dyalog load=HelloWorld.aplf
C:\Docs\Webinars\200430\Demo>
```

**Hello World!**

This was loaded from *C:/Docs/Webinars/200430/Demo/*

## 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,
}
}
```

# Version 19.0 - Deployment

Talks@Dyalog'20  
Tue 14:00 Morten Kromberg  
Dyalog's Docker Containers

- New Public Docker containers
  - dyalog/jarvis – Dyalog APL with Jarvis
  - dyalog/dyalog:dotnetcore – APL on Linux with DotNet Core installed

# Version 19.0 - Deployment

Talks@Dyalog'20  
Tue 14:00 Morten Kromberg  
Dyalog's Docker Containers

- New Public Docker containers
  - dyalog/jarvis – Dyalog APL with Jarvis
  - dyalog/dyalog:dotnetcore – APL on Linux with DotNet Core installed
- Reworked session I/O will remove need to allocate a pseudo-TTY (docker run -it ...)
  - ... but still be able to debug it using RIDE

# Version 19.0 - Deployment

Talks@Dyalog'20  
Tue 14:00 Morten Kromberg  
Dyalog's Docker Containers

- New Public Docker containers
  - dyalog/jarvis – Dyalog APL with Jarvis
  - dyalog/dyalog:dotnetcore – APL on Linux with DotNet Core installed
- Reworked session I/O will remove need to allocate a pseudo-TTY (docker run -it ...)
  - ... but still be able to debug it using RIDE
- NuGet packages still experimental
  - Contact us if you are interested in Dyalog updates via NuGet



# 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
  - ✿ Many thanks to Phil Last!
  - ✿ Planning a final round of community review

Talks@Dyalog'20

Tue 14:55 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)
]
```

# Increased Testing

- We have been steadily increasing our testing with each release
- The version 18.0 experience has shown the importance of this, and we will continue building out our tests
  - We will have automated "CI", building and testing several APL tools this winter
  - We are developing automated testing of our IDE UI
  - We are working towards full code coverage of primitive functions and operators
- The version 18.0 experience has also shown that increasing the volume of testing is not sufficient in itself



# Building Security In Maturity Model

- Client organisations in sensitive sectors face demands to audit core technology providers like Dialog
- We have decided to undertake a BSIMM audit of our development and operational practices
- In addition to allowing clients to check this box, we think this is a valuable step on the road to consistent quality and security

## BSIMM



# Destination Documentation Training Materials

# Documentation and Training Materials

- Reworked Mastering Dyalog APL
- Maintain high frequency of webinars
  - And webcasts
- New and improved TryAPL
- Open Source Project Templates
  - 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

**Talks@Dyalog'20**

**Mon 16:55 Rodrigo Girão Serrão**

Reworking Mastering Dyalog APL

**Tue 17:45 Richard Park**

APL Online!

# Destinations

- Language Engine
- Interfaces
- Application Components
- Development Tools
- Deployment Mechanisms
  
- Documentation and Training Materials

