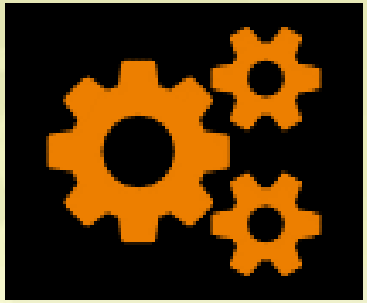
Introducing Dyalog version 18.0

But First: Switching Gears



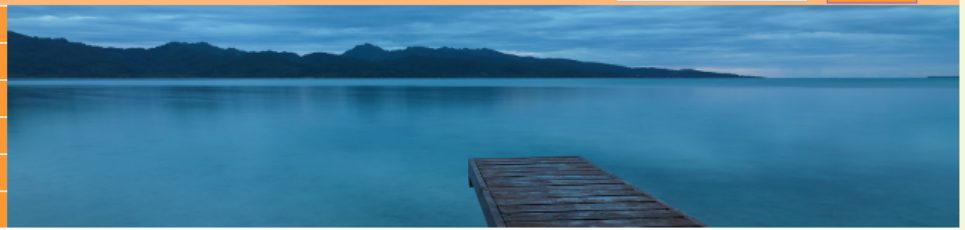
- Due to Corona virus lockdowns, many APL meetings have been cancelled or postponed
  - FinnAPL, SwedAPL, APL Germany, British APL Association, NYC Meetups...
- Dyalog will now run ”webinars” every two weeks
  - We will also start producing "Web casts" to explain how to get started with new (and some old) features.

DYALOG The tool of thought for software solutions

Contact Us

Home Business Learning Community Resources News About Us

- Products
- Dyalog Services
- Prices and Licences
- Support (DSS)
- Download Dyalog – Free
- Dyalog '20



Home >> News >> Event Calendar

Dates for Your Diary

by date | by category | past events

April 2020

- **30** Dyalog webinar: Morten presents **Introducing Dyalog version 18.0**

May 2020

- ~~**4-5** APL Germany spring conference in Berlin, Germany~~ POSTPONED
- **7** BAA webinar: Adám Brudzewsky (Dyalog Ltd) presents **Meet the New APL Wiki**
- **14** Dyalog webinar: Adám presents **Language Features of Dyalog version 18.0 in Depth**
- **21** BAA webinar: Phil Last presents **Acre Source-code Manager Update**
- **28** Dyalog webinar: RichardP presents **APL and Microsoft Excel**

...

October 2020

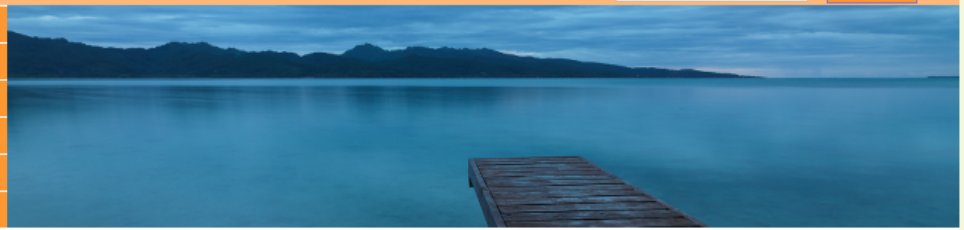
- **11-15** Dyalog '20 in Olhão, Portugal

DYALOG The tool of thought for software solutions

Contact Us

Home Business Learning Community Resources News About Us

- Products
- Dyalog Services
- Prices and Licences
- Support (DSS)
- Download Dyalog – Free
- Dyalog '20



Home >> News >> Event Calendar

Dates for Your Diary

by date | by category | past events

April 2020

- **30** Dyalog webinar: Morten presents **Introducing Dyalog version 18.0**

May 2020

- ~~**4-5** APL Germany spring conference in Berlin, Germany~~ POSTPONED
- **7** BAA webinar: Adám Brudzewsky (Dyalog Ltd) presents **Meet the New APL Wiki**
- **14** Dyalog webinar: Adám presents **Language Features of Dyalog version 18.0 in Depth**
- **21** BAA webinar: Phil Last presents **Acre Source-code Manager Update**
- **28** Dyalog webinar: RichardP presents **APL and Microsoft Excel**

...

October 2020

- **11-15** Dyalog '20 in Olhão, Portugal

Dyalog Version 18.0 - Themes

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

Performance

- "Performance QA" nearly 10%faster

Dyalog Version 18.0 - Themes

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

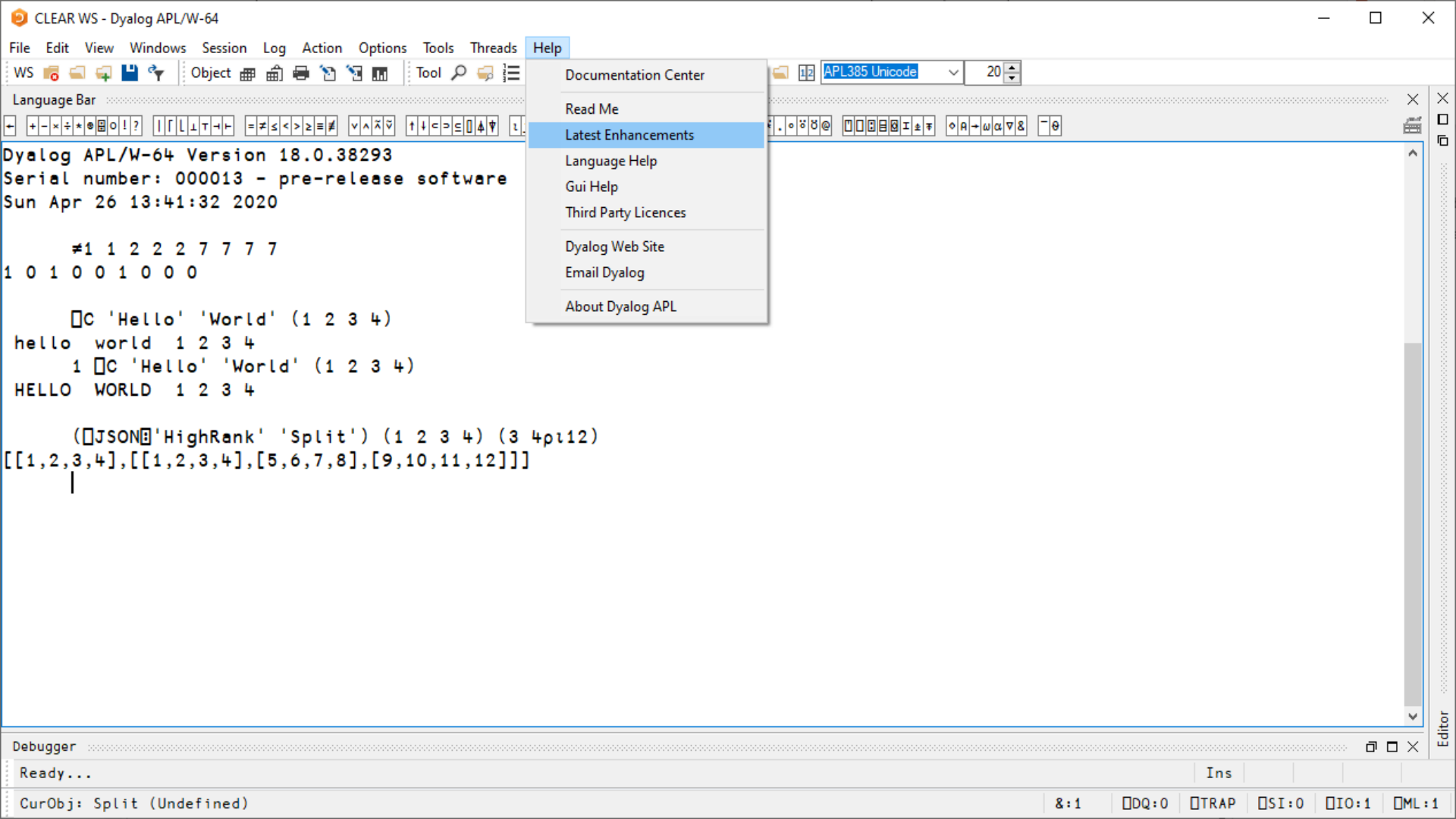
Performance

- "Performance QA" nearly 10%faster

Improving the Tool of Thought

- Operators ⍨ Constant, ⍤ Atop, ⍥ Over
- Function ≠ Unique Mask
- Case Folding with ⎕C
- Date Conversions with ⎕DT
- Multi-Line Input

NB: All Release Notes are on Line!


CLEAR WS - Dyalog APL/W-64
File Edit View Windows Session Log Action Options Tools Threads Help
WS Object Tool
APL385 Unicode
20
Language Bar
Documentation Center
Read Me
Latest Enhancements
Language Help
Gui Help
Third Party Licences
Dyalog Web Site
Email Dyalog
About Dyalog APL
Dyalog APL/W-64 Version 18.0.38293
Serial number: 000013 - pre-release software
Sun Apr 26 13:41:32 2020
≠1 1 2 2 2 7 7 7 7
1 0 1 0 0 1 0 0 0
⎕C 'Hello' 'World' (1 2 3 4)
hello world 1 2 3 4
1 ⎕C 'Hello' 'World' (1 2 3 4)
HELLO WORLD 1 2 3 4
(⎕JSON⍠'HighRank' 'Split') (1 2 3 4) (3 4⍴⍳12)
[[1,2,3,4],[[1,2,3,4],[5,6,7,8],[9,10,11,12]]]
Debugger
Editor
Ready...
Ins
CurObj: Split (Undefined)
&:1
⎕DQ:0
⎕TRAP
⎕SI:0
⎕IO:1
⎕ML:1

Key Features

Installation and Configuration

- Version 18.0 introduces platform-independent configuration files. See Introducing Configuration Files.

Running Script Files

- Version 18.0 provides the capability of launching the Dyalog program on script files. This is achieved by two new parameters which may be specified on the command-line. See Load and LX.

New Language Features

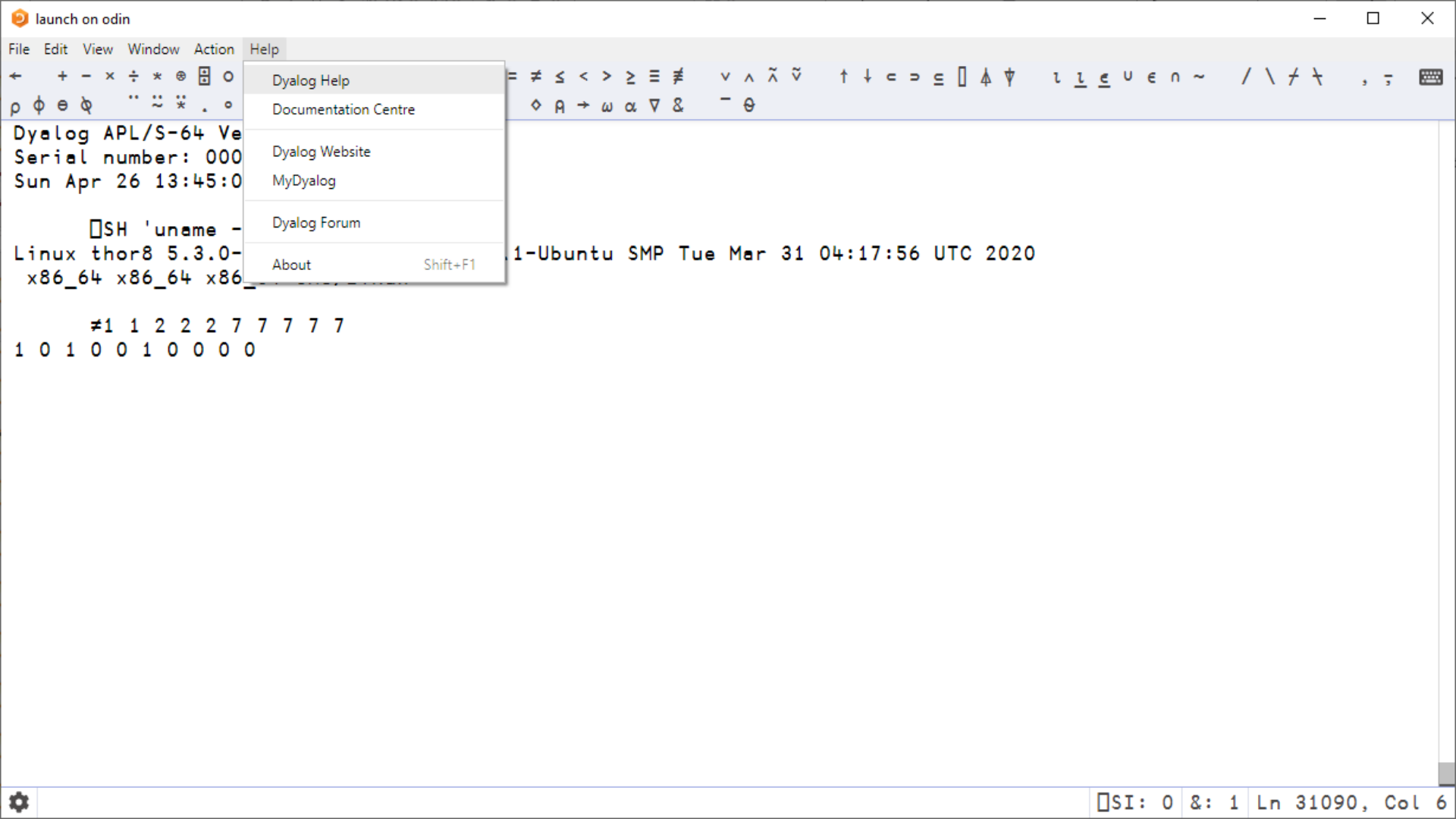
- New monadic ≠ function. See Unique Mask.
- New operators Atop and Constant.
- New System Function ⎕C. This new function is intended to replace 819⌶ which is deprecated. See Case Convert.
- New system function ⎕DT. See Date-time.
- New i-beam function. See Format Date-time.

Improved Language Features

- ⎕JSON provides a new **HighRank** option to handle arrays with rank >1 automatically, and a new **Dialect** option that permits JSON5 extensions.
- ⎕NPUT provides a new **NEOL** option that specifies how embedded line separators are treated.
- The Where function (monadic ⍸), which previously accepted only a Boolean argument, has been extended to allow non-negative integers. See Extension to Where.
- Partitioned enclose has been extended to permit empty result elements. See Partitioned Enclose.
- Mix has been improved so that certain APL2 extensions which were provided only when ⎕ML≥2 are now provided at all levels of ⎕ML. See Extensions to Mix.
- ⎕S and ⎕R now have a **Regex** Variant option that can be used to disable regular expression matching. See Regex Variant Option.
- The rules concerning the serialisation of arrays that contain external references have been re-defined. See Serialising Namespaces.

Documentation Changes

- The Composition operator, which was previously described in terms of 4 forms has been renamed Beside (forms I and IV) and Bind (forms II and III). Only the descriptions have changed, the features and funtionality have not changed.
- The documentation for Run Time applications has been updated to take advantage of the new configuration files. See Run-Time Applications and Components.


launch on odin
File Edit View Window Action Help
Dyalog Help
Documentation Centre
Dyalog Website
MyDyalog
Dyalog Forum
About Shift+F1
Dyalog APL/S-64 Ve
Serial number: 000
Sun Apr 26 13:45:0
SH 'uname -
Linux thor8 5.3.0-
1-Ubuntu SMP Tue Mar 31 04:17:56 UTC 2020
x86_64 x86_64 x86_
≠1 1 2 2 2 7 7 7 7 7
1 0 1 0 0 1 0 0 0 0
SI: 0 &: 1 Ln 31090, Col 6

Key Features

Installation and Configuration

- Version 18.0 introduces platform-independent configuration files. See Introducing Configuration Files.
- During installation, `setup.exe` associates a number of new file extensions with the Dyalog APL Editor `dyaedit.exe`. See File Associations.

Running Script Files

- Version 18.0 provides the capability of launching the Dyalog program on script files. This is achieved by two new parameters which may be specified on the command-line. See Load and LX.

New Language Features

- New monadic ≠ function. See Unique Mask.
- New operators Atop and Constant.
- New System Function ⎕C. This new function is intended to replace `819⌶` which is deprecated. See Case Convert.
- New system function ⎕DT. See Date-time.
- New i-beam function. See Format Date-time.

Improved Language Features

- ⎕JSON provides a new **HighRank** option to handle arrays with rank >1 automatically, and a new **Dialect** option that permits JSON5 extensions.
- ⎕NPUT provides a new **NEOL** option that specifies how embedded line separators are treated.
- The Where function (monadic `⍸`), which previously accepted only a Boolean argument, has been extended to allow non-negative integers. See Extension to Where.
- Partitioned enclose has been extended to permit empty result elements. See Partitioned Enclose.
- Mix has been improved so that certain APL2 extensions which were provided only when ⎕ML≥2 are now provided at all levels of ⎕ML. See Extensions to Mix.
- ⎕S and ⎕R now have a **Regex** Variant option that can be used to disable regular expression matching. See Regex Variant Option.
- The rules concerning the serialisation of arrays that contain external references have been re-defined. See Serialising Namespaces.

Documentation Changes

Launching APL on Source Files

Launching APL on Source Files

- The interpreter can be launched on
  - a binary workspace (.dws)
  - a "dyalog application file" (.dyapp)

Launching APL on Source Files

- The interpreter 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)

Launching APL on Source Files

Launching APL

HelloWorld.aplf - Notepad

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
∇
```

Launching APL

HelloWorld.aplf - Notepad

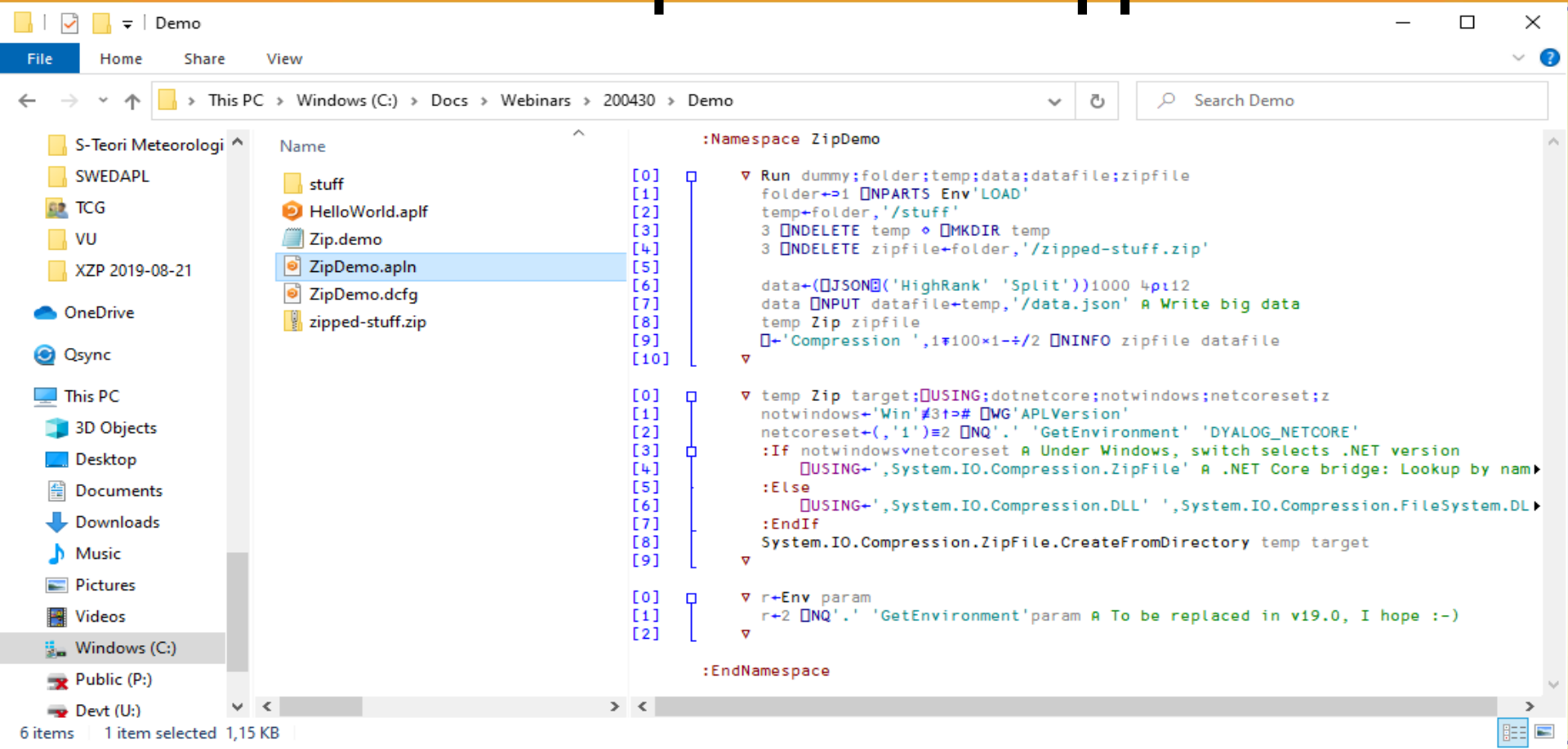
File Edit Format View Help

```
∇ 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
∇
```

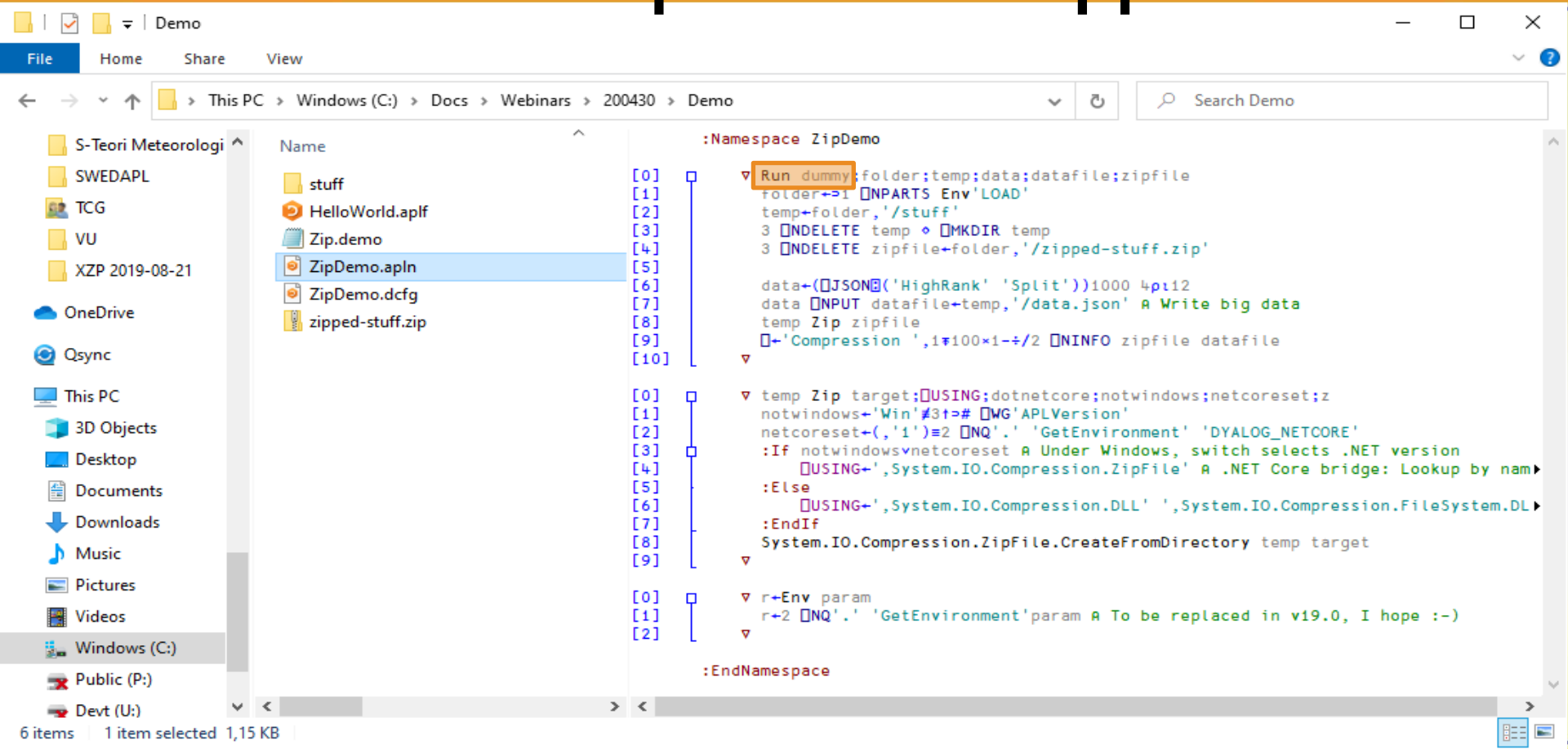
Command Prompt

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

C:\Docs\Webinars\200430\Demo>
```

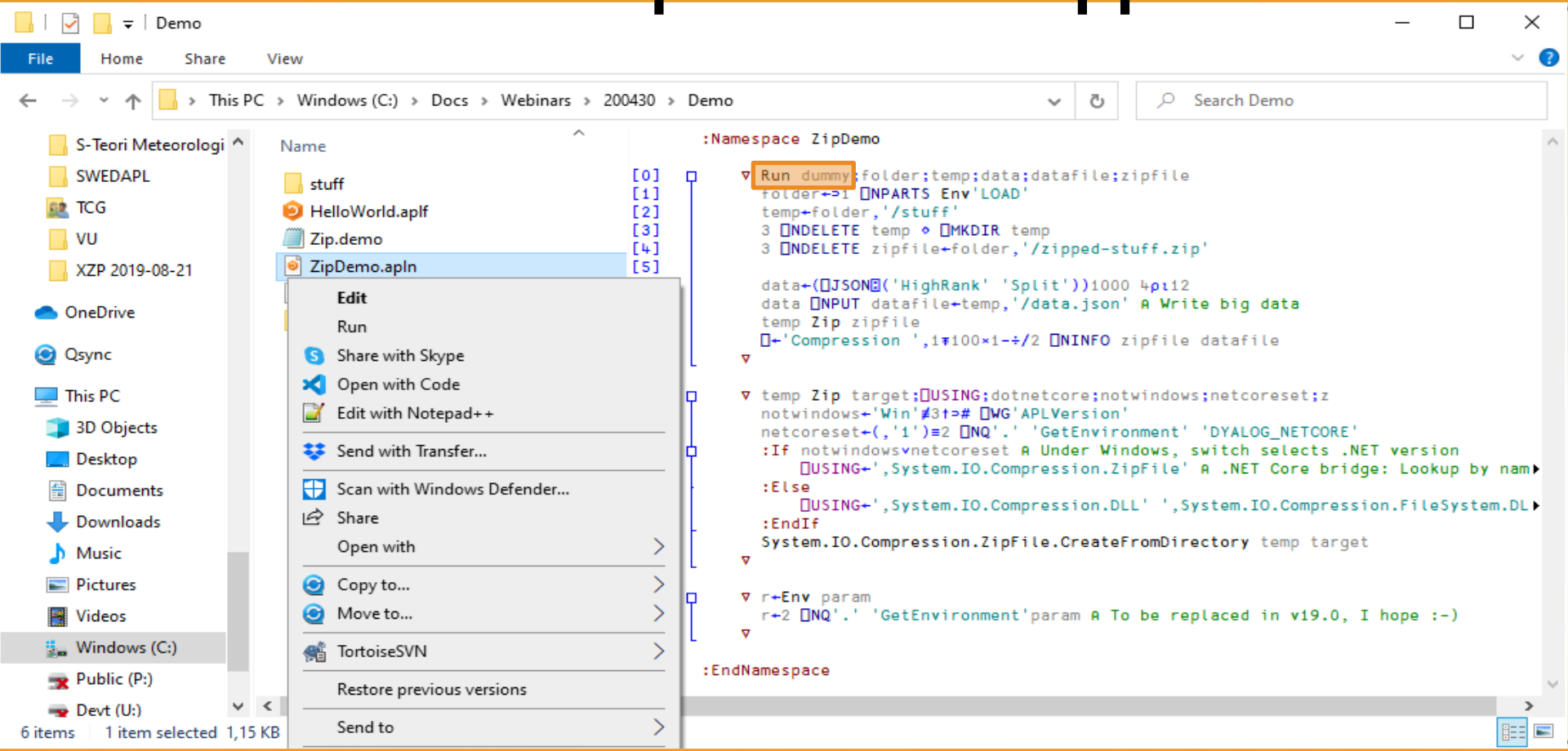
1 Namespace File = Application



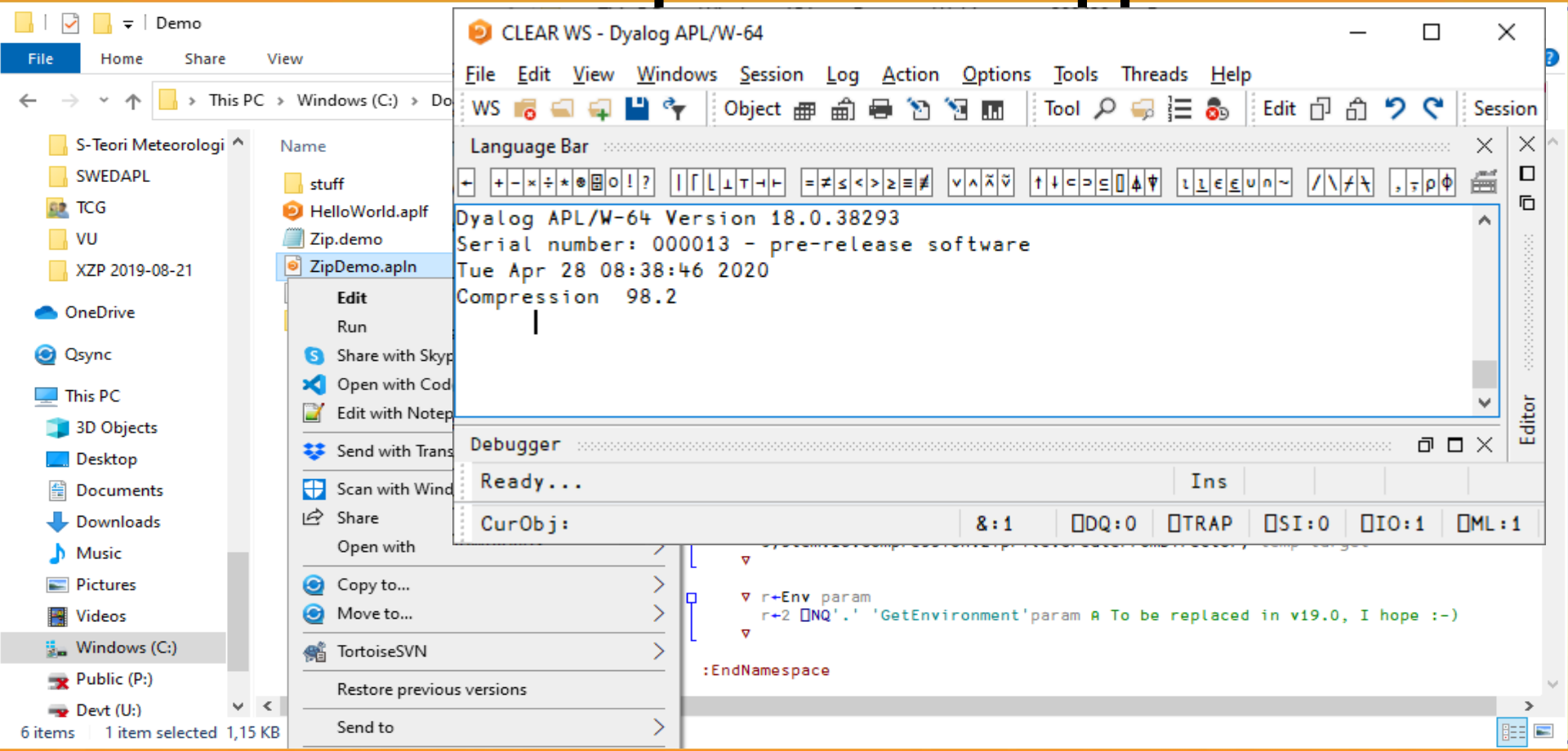
1 Namespace File = Application



1 Namespace File = Application



1 Namespace File = Application



Who cares? I *LIKE* my Workspace!

- So keep using workspaces ...
- New users and modern organisations want to use:
  - Editors they already know
    - VS Code, Emacs, VIM
  - Source code management tools
    - Git[Hub], SVN, ...
  - Continuous Integration Pipelines
- Transparency: Text can be viewed, transmitted and edited by anyone
- No ”Build” process (unless you want one, in which case it is now really easy to create one)

The LOAD= parameter

The LOAD= parameter

- Specifies the "artefact" to be used to start the APL session

The LOAD= parameter

- Specifies the "artefact" to be used to start the APL session
- Workspace files will be ⎕LOAD'ed

The LOAD= parameter

- Specifies the "artefact" to be used to start the APL session
- Workspace files will be ⎕LOAD'ed
- All other files will be [attempted] ⎕FIXed into #

The LOAD= parameter

- Specifies the "artefact" to be used to start the APL session
- Workspace files will be ⎕LOAD'ed
- All other files will be [attempted] ⎕FIXed into #
- For the following file extensions, a function is called w/ a right argument:
  - `name.aplf` (function): The function will be called
  - `name.apln` (namespace): Call `name.Run`
  - `name.aplc` (class): Call `name.Run` (*Shared Public*)

The LOAD= parameter

- Specifies the "artefact" to be used to start the APL session
- Workspace files will be ⎕LOAD'ed
- All other files will be [attempted] ⎕FIXed into #
- For the following file extensions, a function is called w/ a right argument:
  - `name.aplf` (function): The function will be called
  - `name.apln` (namespace): Call `name.Run`
  - `name.aplc` (class): Call `name.Run` (*Shared Public*)
- For any other file extension, nothing will run
  unless the LX= parameter is also set

The LOAD= parameter

- Specifies the "artefact" to be used to start the APL session
- Workspace files will be ⎕LOAD'ed
- All other files will be [attempted] ⎕FIXed into #
- For the following file extensions, a function is called w/ a right argument:
  - `name.aplf` (function): The function will be called
  - `name.apln` (namespace): Call `name.Run`
  - `name.aplc` (class): Call `name.Run` (*Shared Public*)
- For any other file extension, nothing will run
  unless the LX= parameter is also set
- DO NOT MAKE ANY ASSUMPTIONS
  about the contents form of the right argument!

Configuration Files

Configuration Files

- Imagine ...

Configuration Files

- Imagine ...
- ... if the application configuration could *ALSO be just another text file...

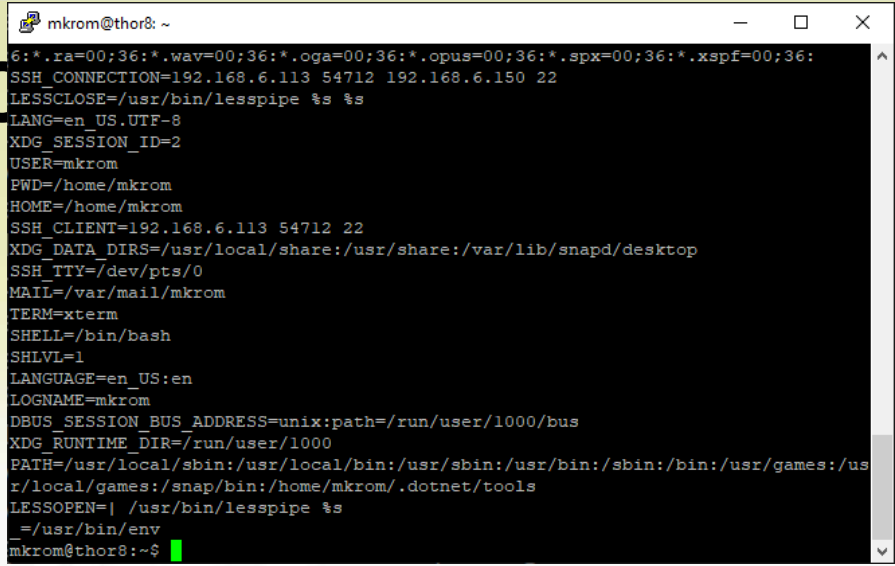
Configuration - History

Configuration - History

- DOS & UNIX:
  Environment Variables

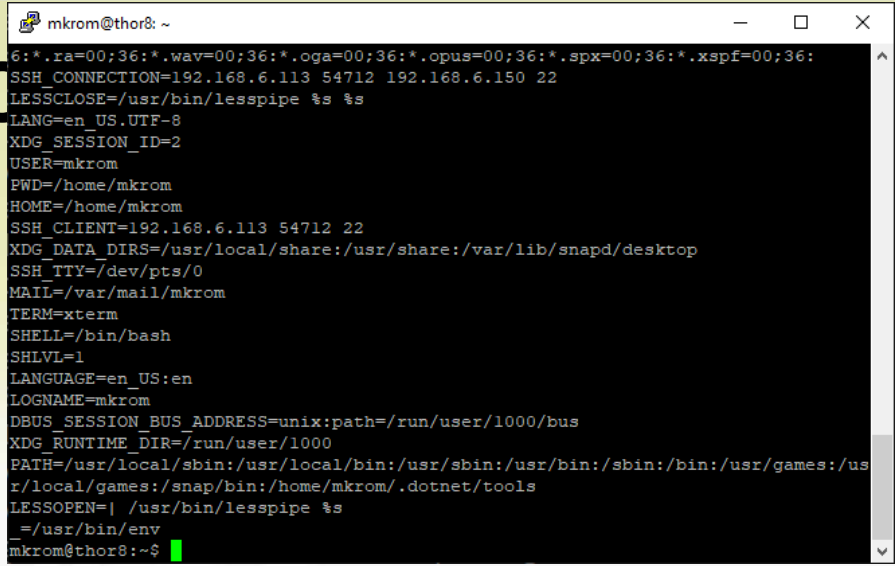
Configuration - Histo

- DOS & UNIX:
  Environment Variables



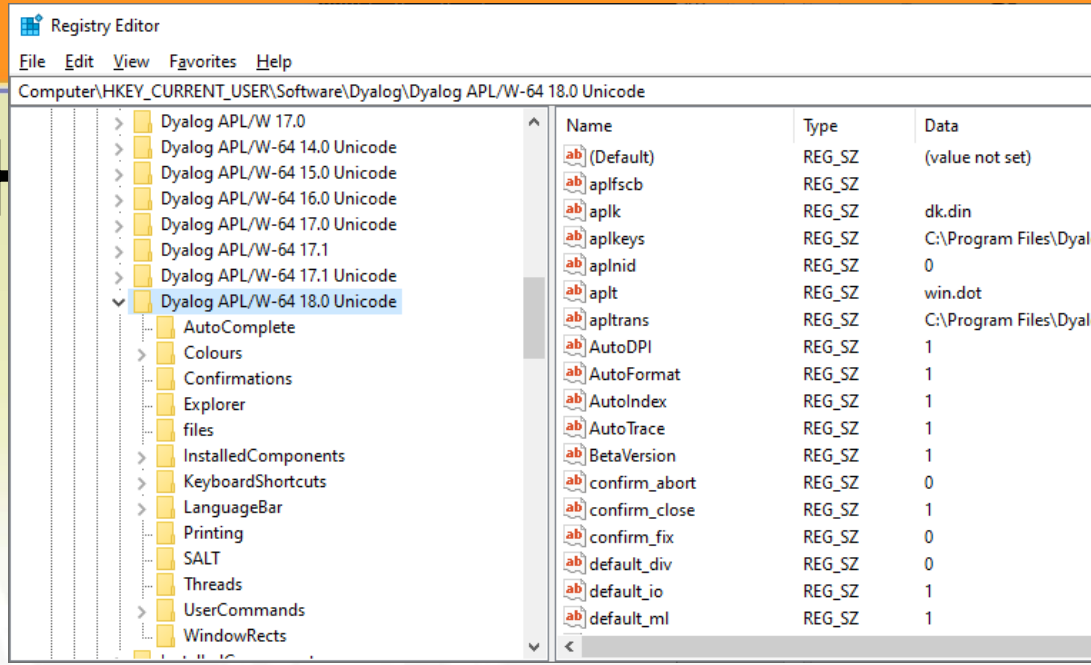
Configuration - Histo

- DOS & UNIX:
  Environment Variables
- Windows:
  - "INI files"
  - Windows Registry

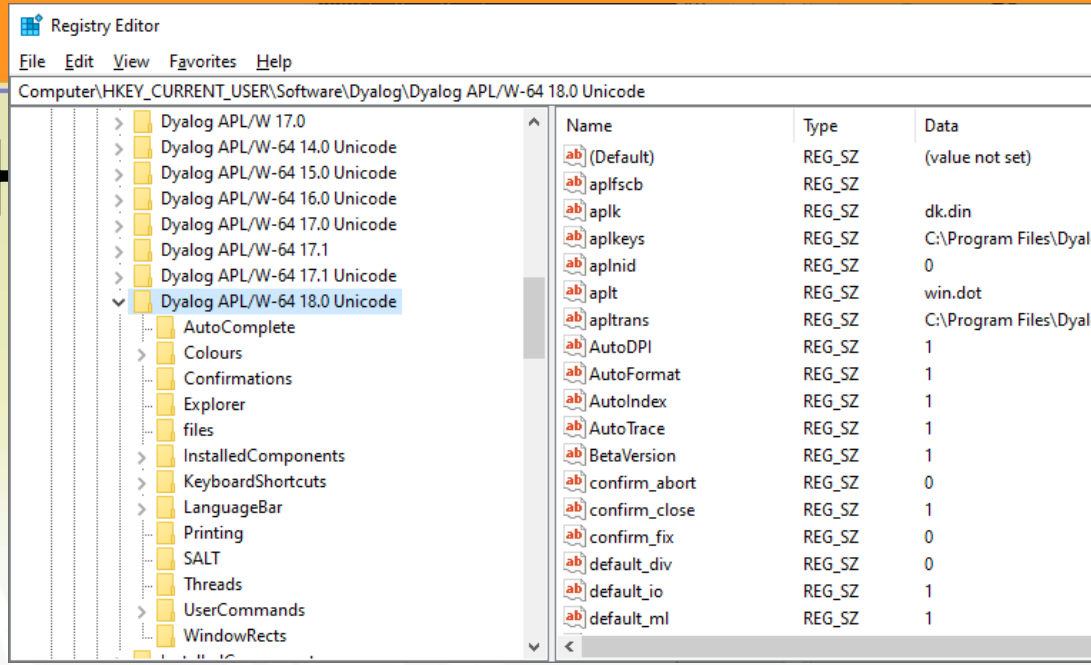


Configuration - H

- DOS & UNIX: Environment Variables
- Windows:
  - "INI files"
  - Windows Registry



Configuration - H



- DOS & UNIX:
  Environment Variables
- Windows:
  - "INI files"
  - Windows Registry
- Command line switches:

Windows: `dyalog app.dws MAXWS=1G`

'Nix: `MAXWS=1G dyalog app.dws`

Version 18.0: Configuration Files

Version 18.0: Configuration Files

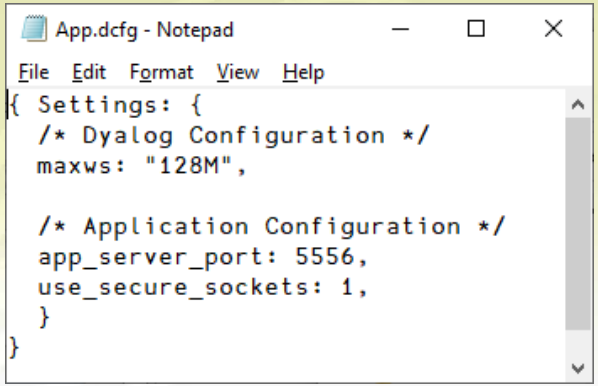
- Identical across platforms

Version 18.0: Configuration Files

- Identical across platforms
- Easily readable & editable text files (sez Morten)

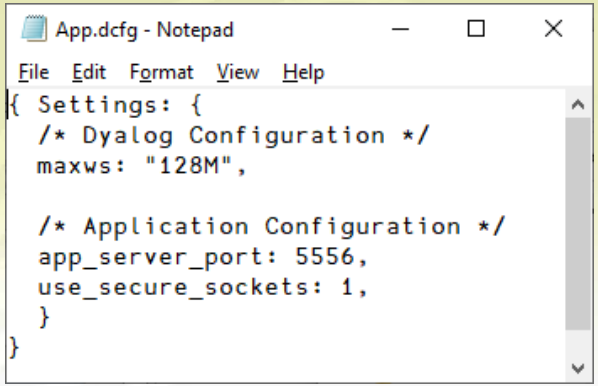
Version 18.0: Configuration Files

- Identical across platforms
- Easily readable & editable text files (sez Morten)

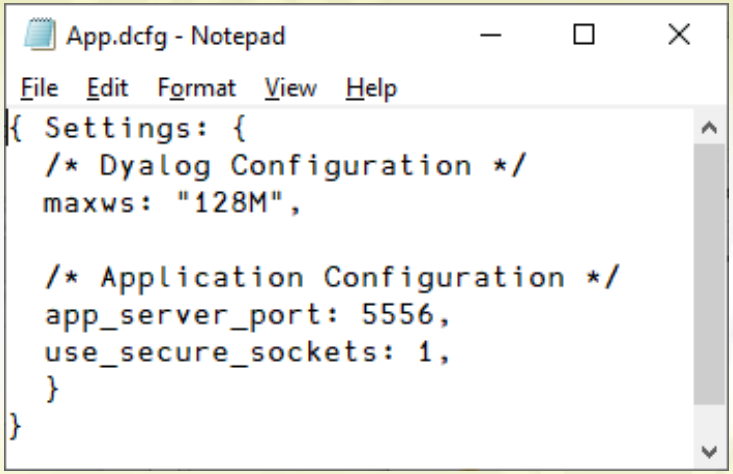


Version 18.0: Configuration Files

- Identical across platforms
- Easily readable & editable text files (sez Morten)
- Provide the ability to configure
  - each application
  - each version of APL
  - each user

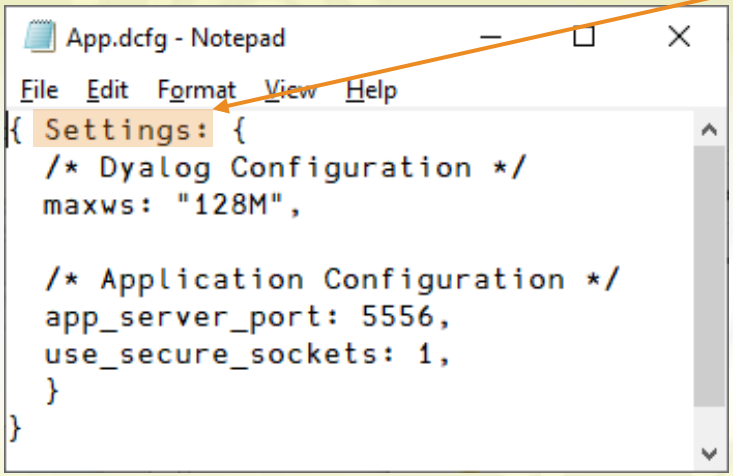


Configuration Files: JSON5



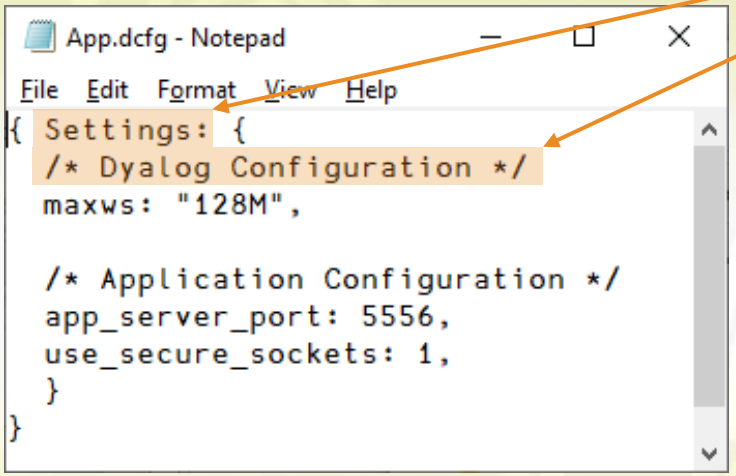
JSON5 allows
- Unquoted names
- Comments
- A trailing comma

Configuration Files: JSON5



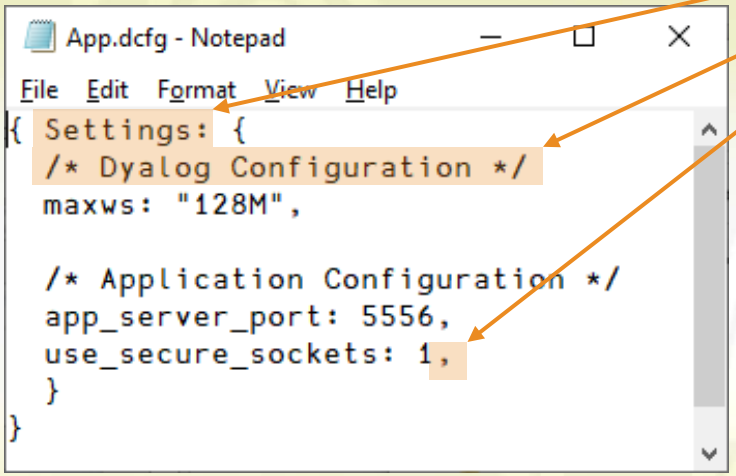
JSON5 allows
- Unquoted names
- Comments
- A trailing comma

Configuration Files: JSON5



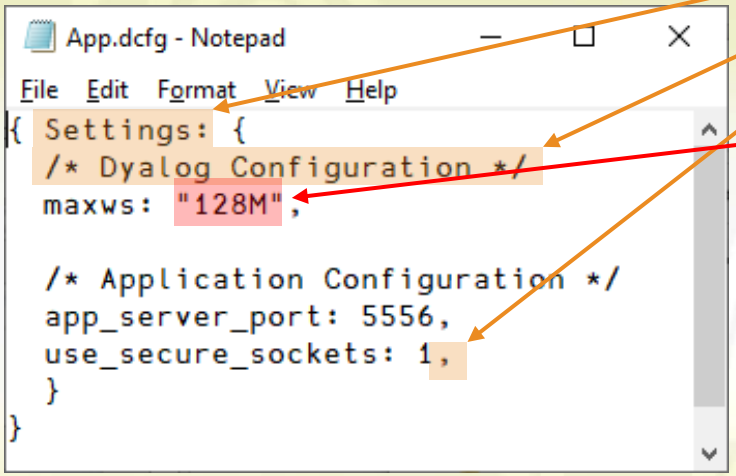
JSON5 allows
- Unquoted names
- Comments
- A trailing comma

Configuration Files: JSON5



JSON5 allows
- Unquoted names
- Comments
- A trailing comma

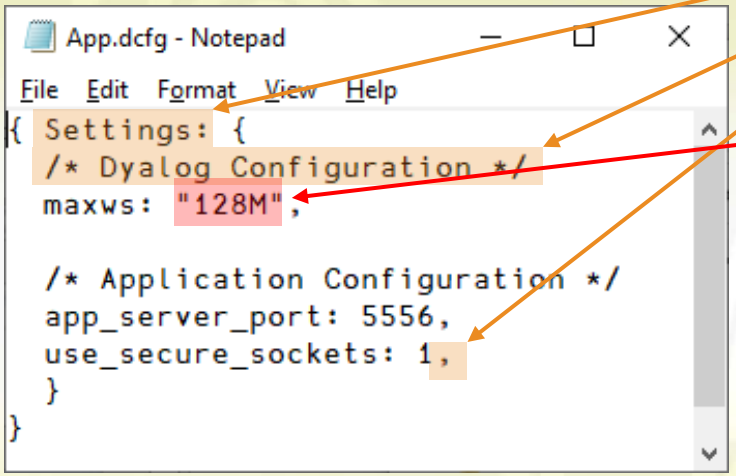
Configuration Files: JSON5



JSON5 allows
- Unquoted names
- Comments
- A trailing comma

NB you DO need quotes here

Configuration Files: JSON5

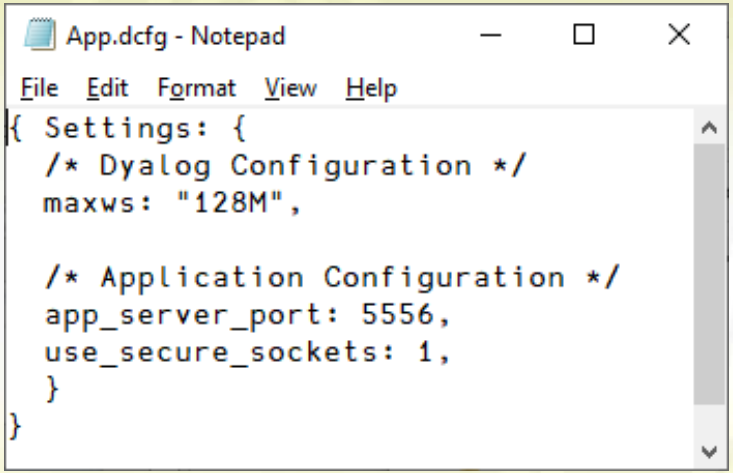


JSON5 allows
- Unquoted names
- Comments
- A trailing comma

NB you DO need quotes here

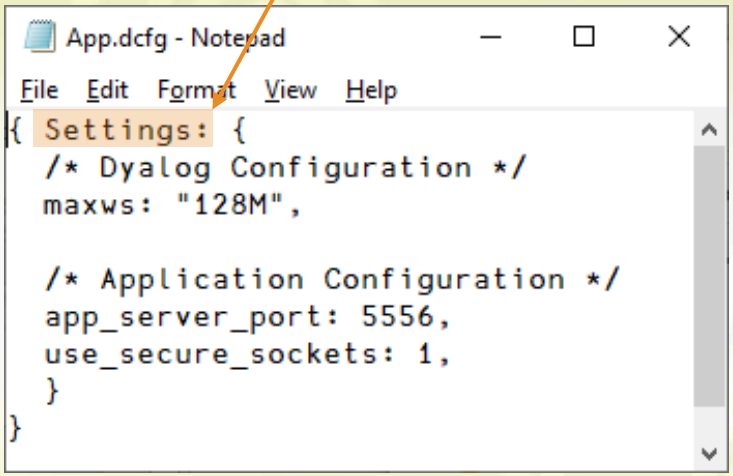
JSON can read JSON5 files, but cannot "round trip" files containing comments.

Configuration Files: JSON5



Configuration Files: JSON5

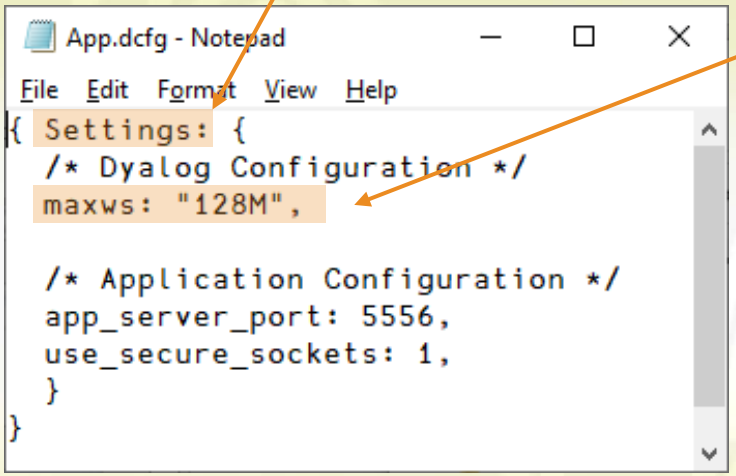
All parameters are within "Settings" (all names are case insensitive)



Configuration Files: JSON5

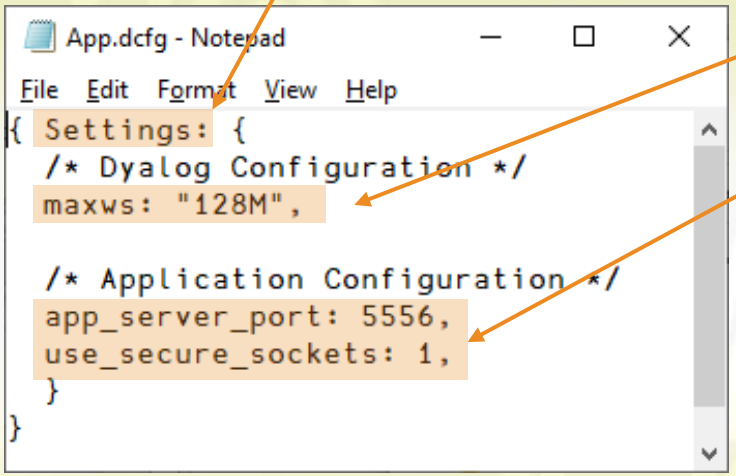
All parameters are within "Settings" (all names are case insensitive)

You can set all Dyalog parameters



Configuration Files: JSON5

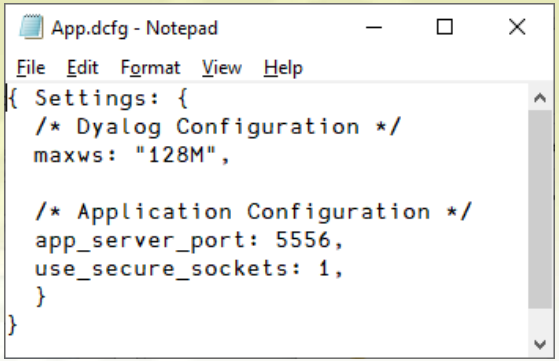
All parameters are within "Settings" (all names are case insensitive)



You can set all Dyalog parameters

And your own application-specific parameters

Configuration Files can *Cascade*



Configuration Files can *Cascade*

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration */
   app_server_address: "localhost",
   }
}
```

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

Configuration Files can *Cascade*

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration */
   app_server_address: "localhost",
   }
}
```

maxws
app_server_port
use_secure_sockets
are decided by
App.dcfg

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

Configuration Files can *Cascade*

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration */
   app_server_address: "localhost",
   }
}
```

maxws
app_server_port
use_secure_sockets
are decided by
App.dcfg

app_server_address
by Client.dcfg

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

Configuration Files can *Cascade*

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration */
   app_server_address: "localhost",
   }
}
```

Server.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Dyalog Configuration */
   maxws: "1G",

   /* Application Server Configuration */
   server_debug: 1,
   }
}
```

maxws
app_server_port
use_secure_sockets
are decided by
App.dcfg

app_server_address
by Client.dcfg

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

Configuration Files can *Cascade*

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration */
   app_server_address: "localhost",
   }
}
```

maxws
app_server_port
use_secure_sockets
are decided by
App.dcfg

app_server_address
by Client.dcfg

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

Server.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Dyalog Configuration */
   maxws: "1G",

   /* Application Server Configuration */
   server_debug: 1,
   }
}
```

Server.dcfg
overrides
the maxws setting

Application and User Configuration

Application and User Configuration

- Application Configuration provides settings needed by the application
  - Specified using CONFIGFILE=
  - **Can be inferred:** if running `myapp.dws` or `myapp.aplf`, Dyalog looks for `myapp.dcfg` in the same folder (if CONFIGFILE is not set)

Application and User Configuration

- Application Configuration provides settings needed by the application
  - Specified using CONFIGFILE=
  - **Can be inferred:** if running `myapp.dws` or `myapp.aplf`, Dyalog looks for `myapp.dcfg` in the same folder (if CONFIGFILE is not set)
- User Configuration usually manages the development environment
  - Specified using USERCONFIGFILE=
  - Defaults to `~/.dyalog/dyalog.180U64.dcfg`
  - ... except under Windows, where the default is to continue to use the Windows Registry

Application and User Configuration

- Application Configuration provides settings needed by the application
  - Specified using CONFIGFILE=
  - **Can be inferred:** if running `myapp.dws` or `myapp.aplf`, Dyalog looks for `myapp.dcfg` in the same folder (if CONFIGFILE is not set)
- User Configuration usually manages the development environment
  - Specified using USERCONFIGFILE=
  - Defaults to `~/.dyalog/dyalog.180U64.dcfg`
  - ... except under Windows, where the default is to continue to use the Windows Registry
- Version-independent User Configuration
  - All Dyalog-supplied User Config files *extend* `~/.dyalog/dyalog.dcfg`

The Big Picture

The Big Picture

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration */
   app_server_address: "localhost",
   }
}
```

Application Configuration inferred or set with CONFIGFILE=

The Big Picture

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration *
   app_server_address: "localhost",
   }
}
```

Application Configuration inferred or set with CONFIGFILE=

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

The Big Picture

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration *
   app_server_address: "localhost",
   }
}
```

Application Configuration inferred or set with CONFIGFILE=

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

*dyalog.180U64.dcfg - Notepad

File Edit Format View Help

User Configuration (usually for devt environment)

```
{ extend: "dyalog.dcfg", settings: {

        /* Morten's Settings */
        MAXWS: "256M",

        /* Workspace search paths */
        WSPATH: [
                ".",
                "[DYALOG]/ws",
                "[DYALOG]/xflib",
                "[DYALOG]/Samples/OO4APL",
        ],

        /* Session file location */
        SESSION_FILE: "[DYALOG]/default.dse",

        /* Session log file location */
        LOG_FILE: "[DYALOG_CONFIGDIR]/session_log_[DYALOG_SHORTVERSION].dlf",

        /* SALT settings file */
```

The Big Picture

Client.dcfg - Notepad

File Edit Format View Help

```
{  extend: "App.dcfg",

   settings: {
   /* Application Client Configuration *
   app_server_address: "localhost",
   }
}
```

Application Configuration inferred or set with CONFIGFILE=

App.dcfg - Notepad

File Edit Format View Help

```
{ Settings: {
  /* Dyalog Configuration */
  maxws: "128M",

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

*dyalog.180U64.dcfg - Notepad

File Edit Format View Help

User Configuration (usually for devt environment)

```
{ extend: "dyalog.dcfg", settings: {

        /* Morten's Settings */
        MAXWS: "256M",

        /* Workspace search paths */
        WSPATH: [
                ".",
                "[DYALOG]/ws",
                "[DYALOG]/xflib",
                "[DYALOG]/Samples/OO4APL",
        ],

        /* Session file location */
        SESSION_FILE: "[DYALOG]/default.dse",

        /* Session log file location */
        LOG_FILE: "[DYALOG_CONFIGDIR]/session_log_[DYALOG_SHORTVERSION].dlf",

        /* SALT settings file */
```

*dyalog.dcfg - Notepad

File Edit Format View Help

```
{ settings: {

        /* Whether to start tracing on errors */
        TRACE_ON_ERROR: 1,

        /* Maximum workspace size */
        MAXWS: "512M",

}}
```

The Even Bigger Picture

All existing mechanisms for configuration continue to be supported.

Configuration files have been injected into the existing precedence rules:

- Command-line settings override
- ***Application configuration file* settings, which override**
- Environment variable settings, which override
- ***User configuration file* settings, which override**
- Settings in the registry (Windows only), which override
- Built-in defaults

Dyalog hopes that config files will replace other mechanisms, especially environment variables and the Windows registry.

Configuration: Future Plans

- Consider user feedback
- Limit the use of the Windows Registry to settings related to the Development Environment
  - Move all "interpreter" settings to a USERCONFIGFILE
  - (as RIDE has it's own configuration)
- Provide a better tool to query configuration than

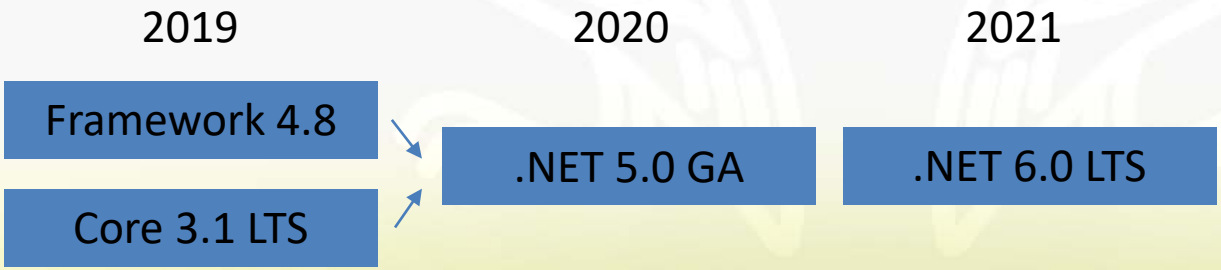
```
2 ⎕NQ '.' 'GetEnvironment' [name]
```

- Ensure that APL will always work without any configuration on all platforms

- (Eventually, a tool to update config files)

Microsoft.NET Core Bridge

- Dyalog APL has had a bridge to the .NET Framework for ~20 years
  - .NET provides a vast library of useful application APIs
- The .NET "Core" is a new, open source, cross platform (Windows, Linux, macOS) implementation of .NET
- Version 18.0 adds a bridge to the .NET Core



.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...

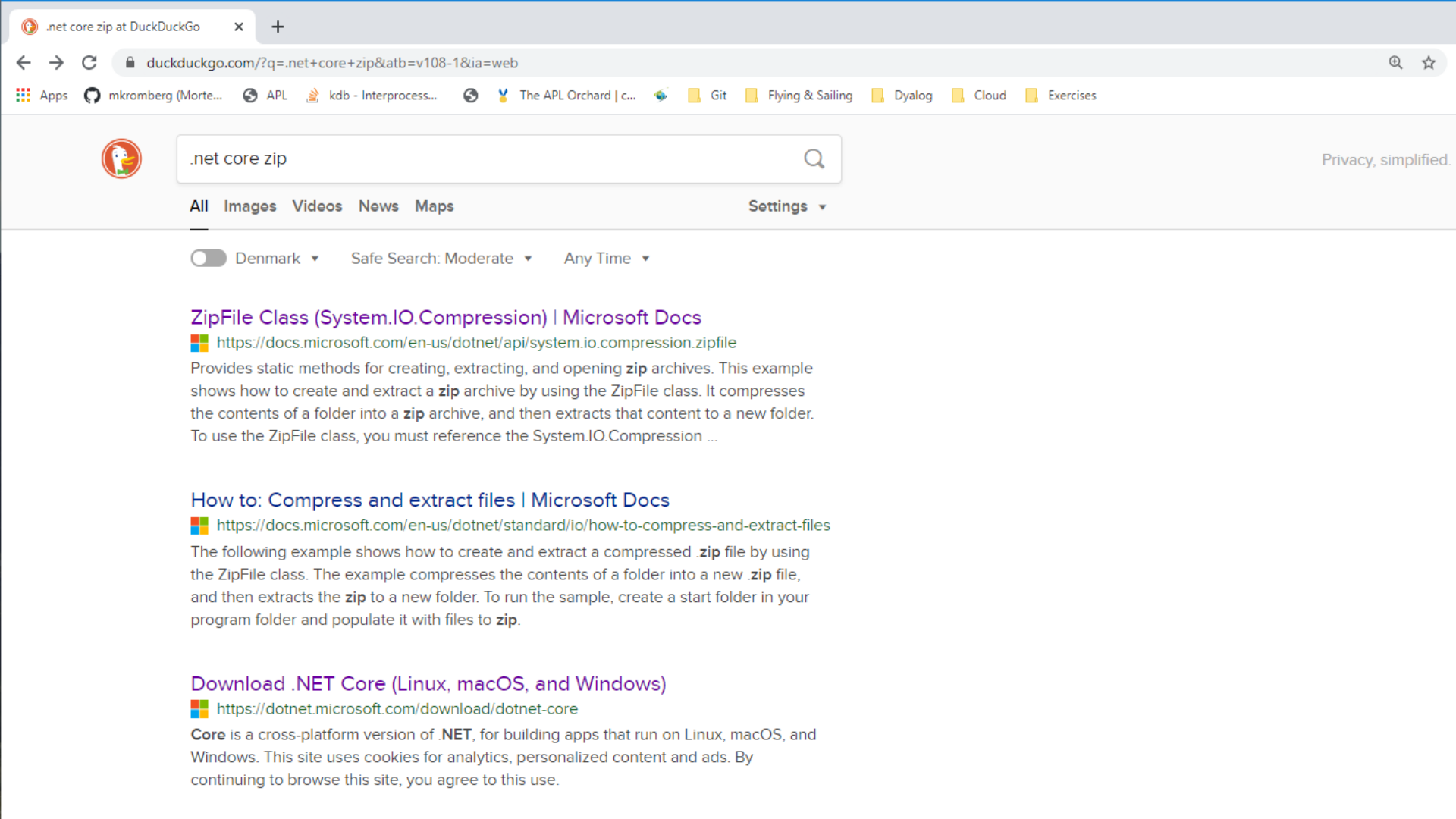
.NET Framework

.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...

https://docs.microsoft.com/en-gb/dotnet/api/?view=netcore-3.1


.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.

ZipFile Class

Namespace: System.IO.Compression
Assembly: System.IO.Compression.ZipFile.dll

Provides static methods for creating, extracting, and opening zip archives.

C# Copy

```
public static class ZipFile
```

Inheritance Object → ZipFile

Examples

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.

Tip

To use the **ZipFile** class, you must reference the `System.IO.Compression.FileSystem` assembly in your project.

C# Copy

```
using System;
using System.IO.Compression;

class Program
```

Version
.NET Core 3.1
Search
System.IO.Compression
BrotliDecoder
BrotliEncoder
BrotliStream
CompressionLevel
CompressionMode
DeflateStream
GZipStream
ZipArchive
ZipArchiveEntry
ZipArchiveMode
ZipFile
ZipFile
Methods
CreateFromDirectory
ExtractToDirectory
Open
OpenRead

ZipFile.CreateFromDirectory Method

Namespace: System.IO.Compression

Assembly: System.IO.Compression.ZipFile.dll

Creates a zip archive that contains the files and directories from the specified directory.

Overloads

CreateFromDirectory(String, String)	Creates a zip archive that contains the files and directories from the specified directory.
CreateFromDirectory(String, String, CompressionLevel, Boolean)	Creates a zip archive that contains the files and directories from the specified directory, uses the specified compression level, and optionally includes the base directory.
CreateFromDirectory(String, String, CompressionLevel, Boolean, Encoding)	Creates a zip archive that contains the files and directories from the specified directory, uses the specified compression level and character encoding for entry names, and optionally includes the base directory.

CreateFromDirectory(String, String)

Creates a zip archive that contains the files and directories from the specified directory.

C# Copy

Is this page helpful?
Yes No
In this article
Definition
Overloads
CreateFromDirectory(String, String)
CreateFromDirectory(String, String, CompressionLevel, Boolean)
CreateFromDirectory(String, String, CompressionLevel, Boolean, Encoding)
Applies to

Demo ZipFile Namespace

Dyalog .NET Bridge Road Map

- v18.0 supports basic use of .NET Core libraries
- v19.0 Goals include
  - Exporting APL code as classes
  - Ability to use both Framework + Core

  New / Improved Functionality over Framework Bridge, e.g.
  - Better support for threading and "Async" patterns
  - Generic classes and methods

Dyalog Version 18.0 - Themes

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

Performance

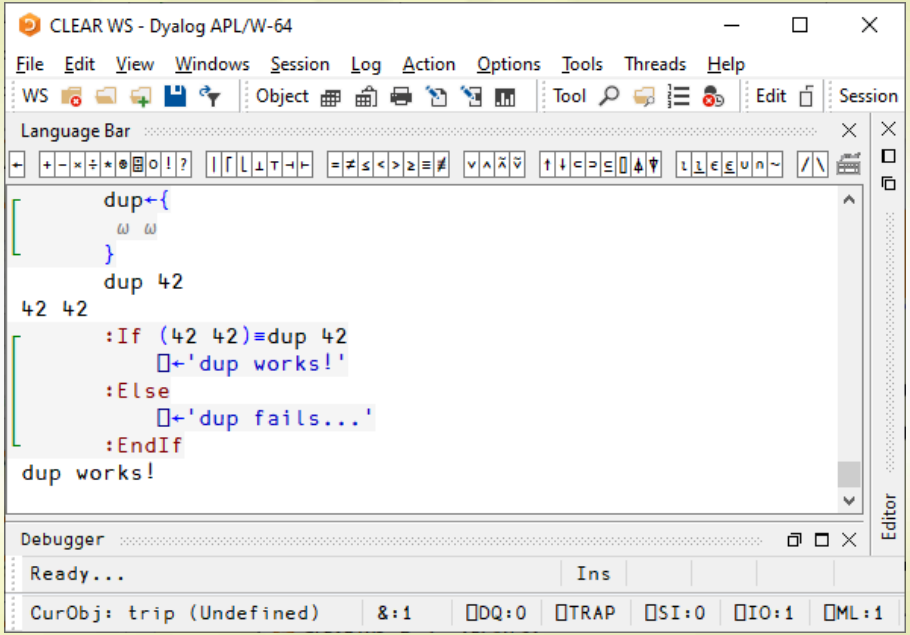
- "Performance QA" nearly 10%faster

Improving the Tool of Thought

- Operators ⍨ Constant, ⍤ Atop ⍥ Over
- Function ≠ Unique Mask
- Case Folding with ⎕C
- Date Conversions with ⎕DT
- Multi-Line Input

Multi-Line Input

- With DYALOG_LINEEDITOR_MODE=1



Supports:

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

Demo Multi-Line Input Mode

- Demo...
- NB: Experimental in v18.0
  - Not supported by RIDE
- A step on the path to making the running of APL scripts more powerful in version 19.0

Dyalog Version 18.0 - Themes

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

Performance

- "Performance QA" nearly 10%faster

Improving the Tool of Thought

- Operators ⍨ Constant, ⍤ Atop ⍥ Over
- Function ≠ Unique Mask
- Case Folding with ⎕C
- Date Conversions with ⎕DT
- Multi-Line Input

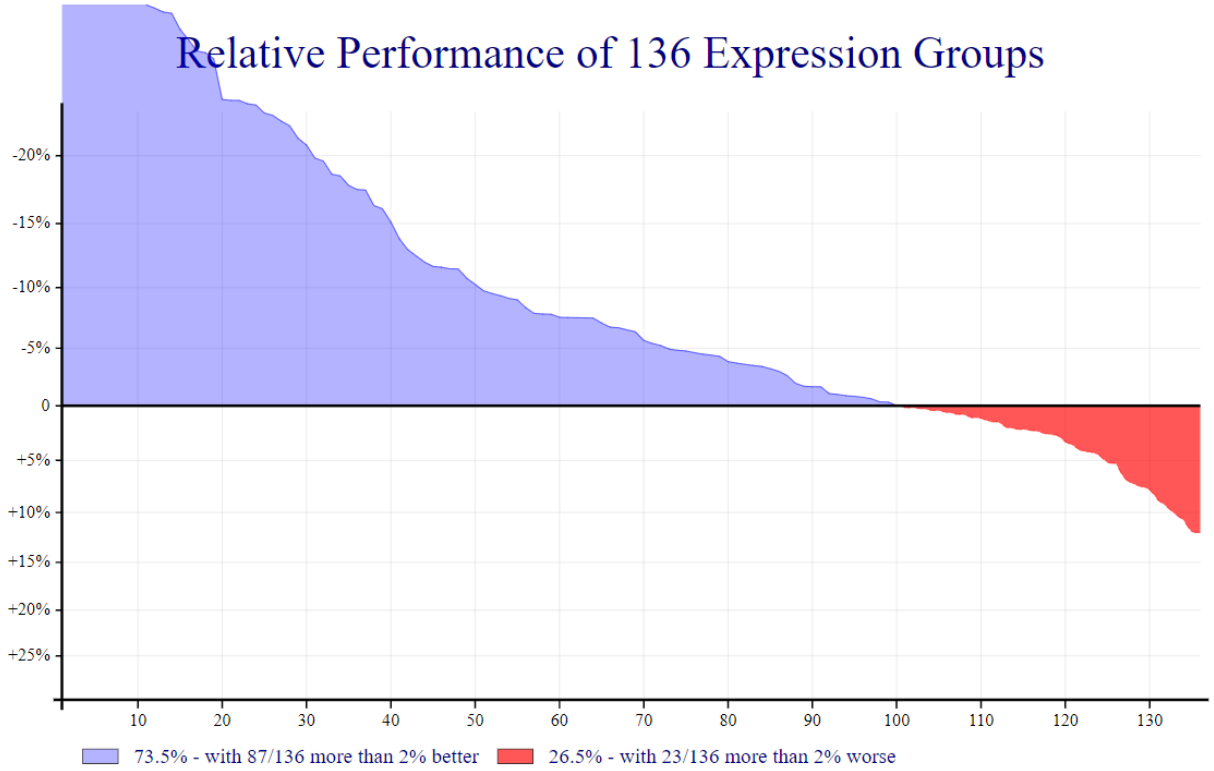
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



Focus Areas

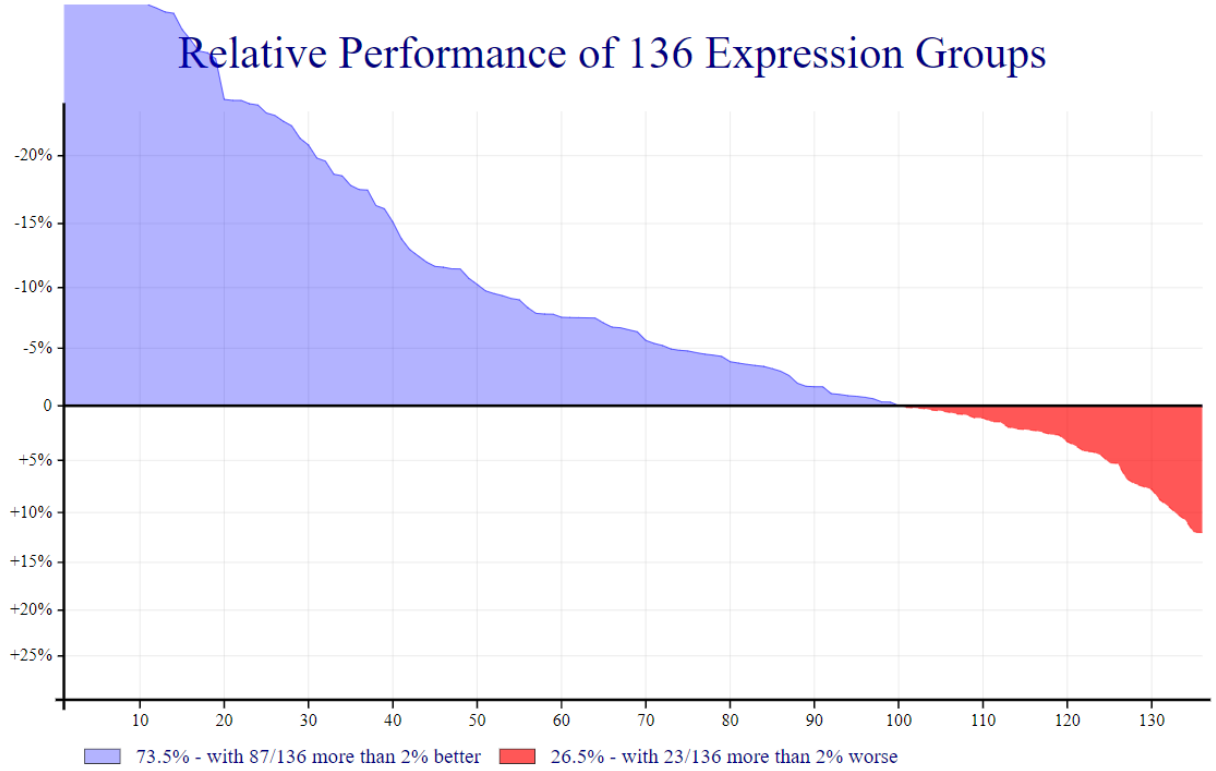
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



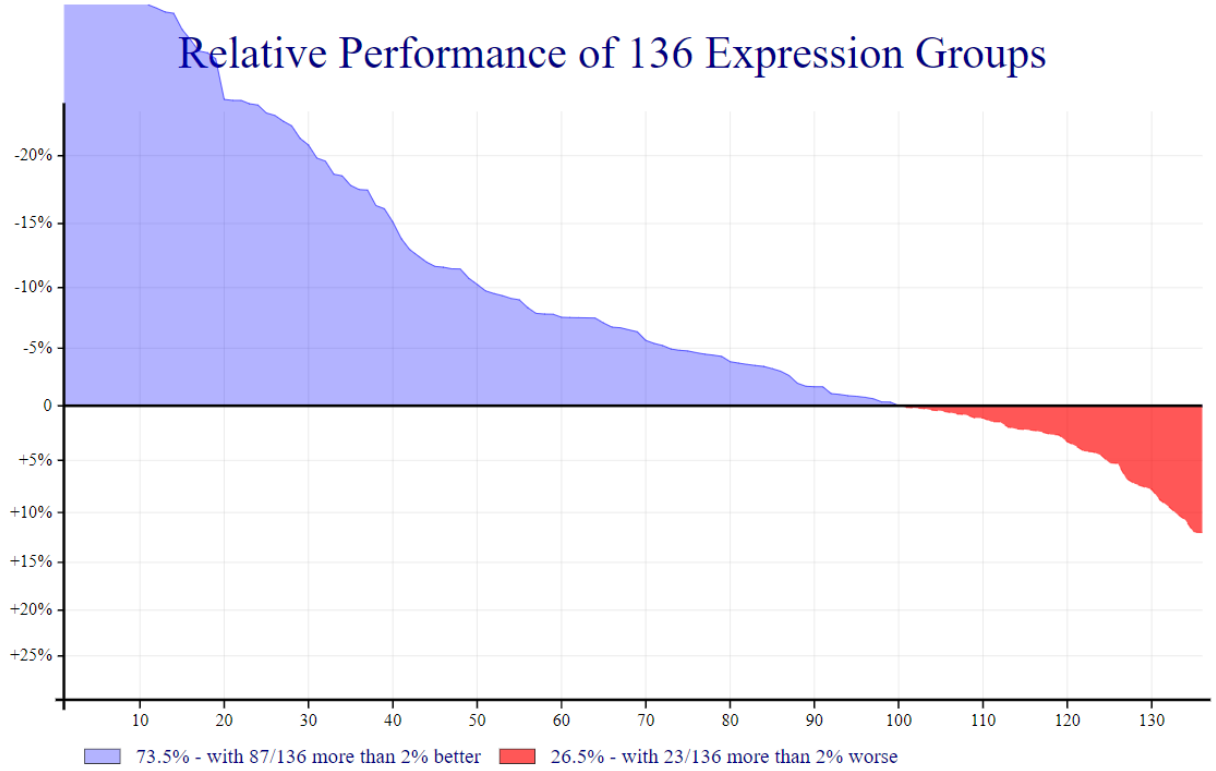
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



Focus Areas

- Cache Usage

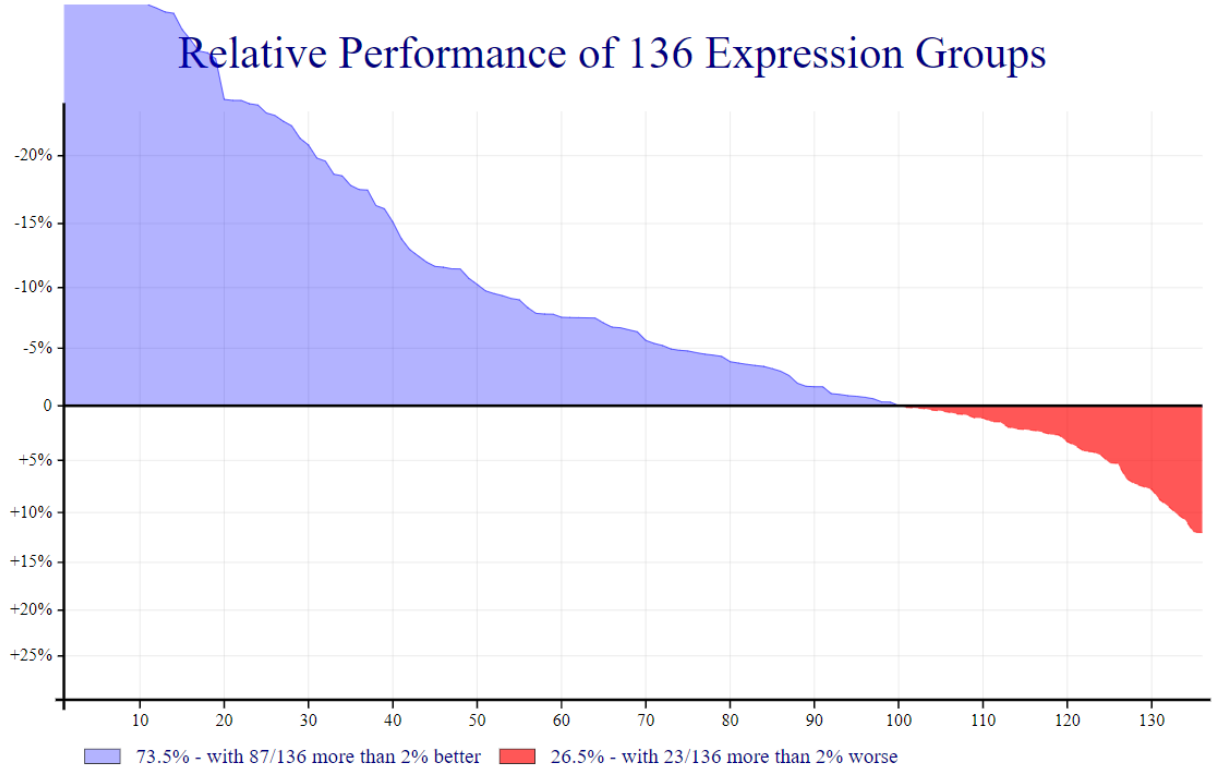
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



Focus Areas

- Cache Usage
- Less Overhead

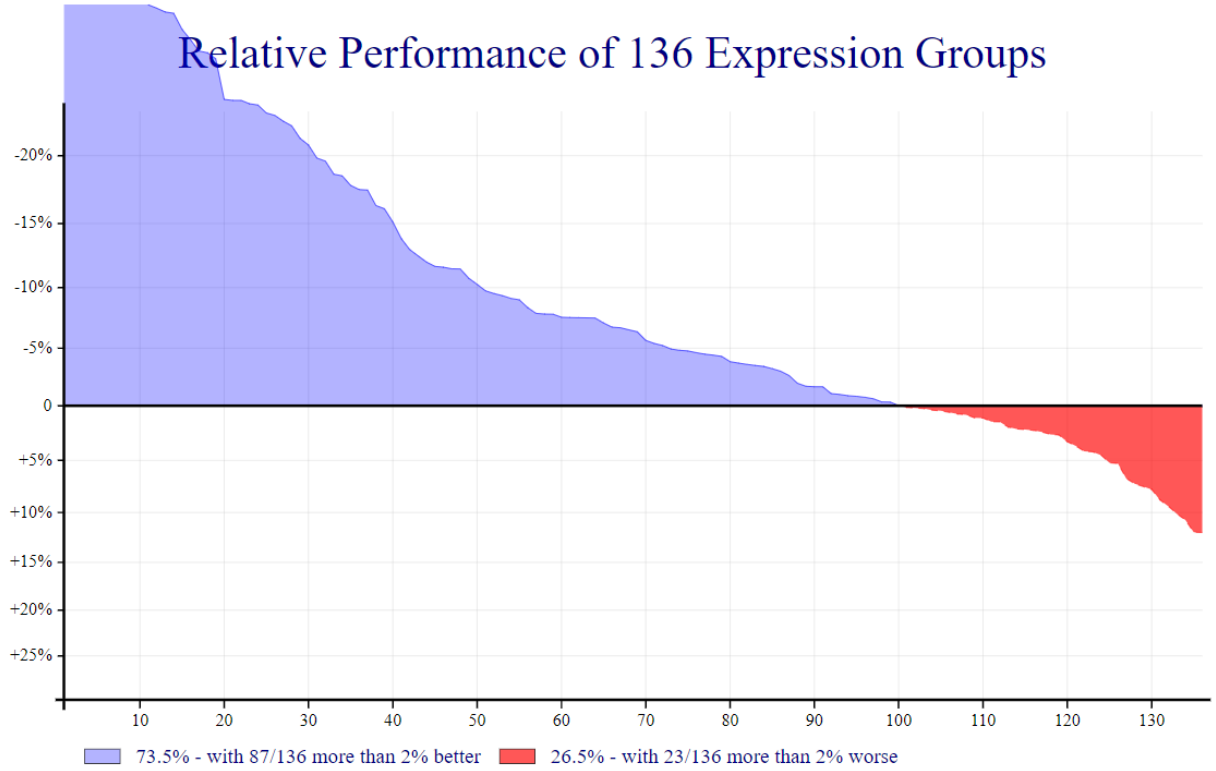
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



Focus Areas

- Cache Usage
- Less Overhead
- Sets `⍳⍷∩∪~`

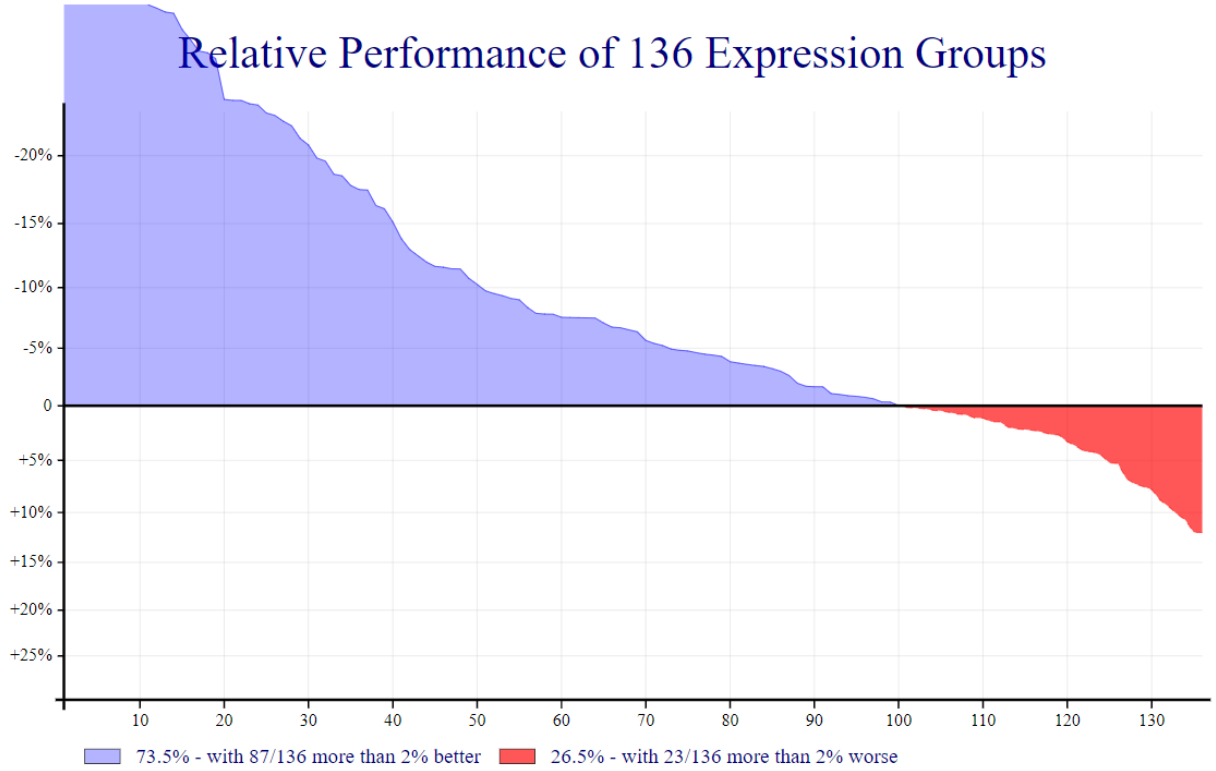
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



Focus Areas

- Cache Usage
- Less Overhead
- Sets `⍳⍷∩∪~`
- Searches

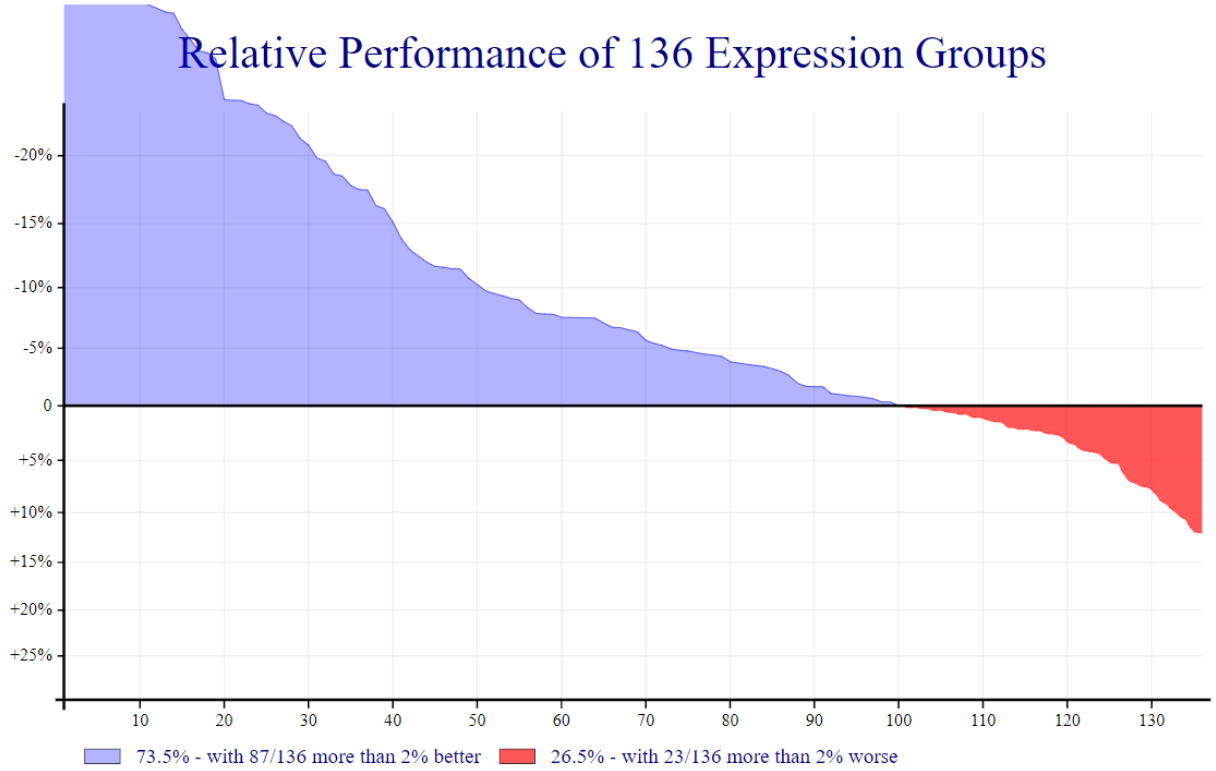
pqareport-18.0.38268v17.1.3684

File | nas00.dyalog.bramley/devt/benchmarks/18.0/pqareport-18.0.38268v17.1.36845.html

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

Performance Comparison
Between Windows-64 18.0.38268.0 W Development and Windows-64 17.1.36845.0 W Development

Geometric mean of 136 expression groups: -10.2%



Focus Areas

- Cache Usage
- Less Overhead
- Sets `⍳⍷∩∪~`
- Searches
- Sorts

Dyalog Version 18.0 - Themes

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

Performance

- "Performance QA" nearly 10%faster

Improving the Tool of Thought

- Operators ⍨ Constant, ⍤ Atop ⍥ Over
- Function ≠ Unique Mask
- Case Folding with ⎕C
- Date Conversions with ⎕DT
- Multi-Line Input

Dyalog Version 18.0 - Themes

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

Performance

- "Performance QA" nearly 10%faster

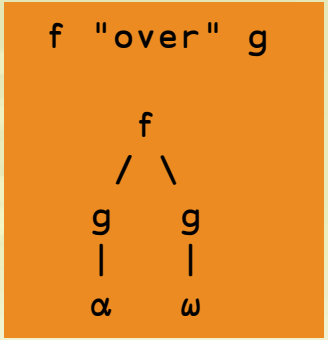
Improving the Tool of Thought

- Operators ⍨ Constant, ⍤ Atop ⍥ Over
- Function ≠ Unique Mask
- Case Folding with ⎕C
- Date Conversions with ⎕DT
- Multi-Line Input

v18 Operators: Over f⍤⍥g

Definition:

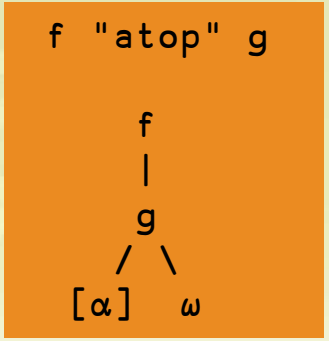
```
      α (f⍤⍥g) ω ↔ (g α) f g ω ↔ "f over g"
```




```
      1 2 3 =⍤⍥| 1 ¯2 3    ⍝ Compare magnitudes
1 1 1

      'Hello' 'World' (,⍤⍥⊆) 2 3⍴⍳12 ⍝ Catenate enclosures
```

Hello	World	1 2 3
		4 5 6

v18 Operators: Atop f⍤g



v18 Operators: Atop f⍤g

Definition:

```
      (f⍤g) ⍵ ↔ f   g ⍵
⍺     (f⍤g) ⍵ ↔ f ⍺ g ⍵
```

```
f "atop" g

    f
    |
    g
   / \
 [⍺]  ⍵
```

v18 Operators: Atop f⍤g

Definition:

```
      (f⍤g) ⍵ ↔ f   g ⍵
α     (f⍤g) ⍵ ↔ f α g ⍵
```

Benefit: Simpler trains

```
f "atop" g

    f
    |
    g
   / \
 [α]  ⍵
```

v18 Operators: Atop f⍤g

```
f "atop" g

    f
    |
    g
   / \
 [α]  ω
```

Definition:

```
      (f⍤g) ω ↔ f   g ω
α     (f⍤g) ω ↔ f α g ω
```

Benefit: Simpler trains

```
((⌊/⊣),(⌈/⊢))      ⍝ Can now be written …
 (⌊/⍤⊣,⌈/⍤⊢)
```

v18 Operators: Atop f⍤g

```
f "atop" g

    f
    |
    g
   / \
 [α]  ω
```

Definition:

```
      (f⍤g) ω ↔ f   g ω
α     (f⍤g) ω ↔ f α g ω
```

Benefit: Simpler trains

```
((⌊/⊣),(⌈/⊢))      ⍝ Can now be written …
 (⌊/⍤⊣,⌈/⍤⊢)

 {(⌊/α),⌈/ω}       ⍝ Morten is a bit old fashioned
```

v18 Operator: Constant ⍺⍨

v18 Operator: Constant ⍺⍨

```
mat←2 3⍴⍳12      ⍝ A small matrix
```

v18 Operator: Constant α⍨

```
mat←2 3⍴⍳12      ⍝ A small matrix

0 ⍨ mat          ⍝ Always 0: Not VERY useful
```

v18 Operator: Constant α⍠⍠

```
      mat←2 3⍴⍳12      ⍝ A small matrix

      0 ⍠⍠ mat         ⍝ Always 0: Not VERY useful
0
```

v18 Operator: Constant α⍨⍨

```
      mat←2 3⍴⍳12      ⍝ A small matrix

      0 ⍨⍨ mat         ⍝ Always 0: Not VERY useful
0

      0 ⍨⍨⍨ mat        ⍝ 0 for each item of mat: (⍴mat)⍴0
```

v18 Operator: Constant α⍨⍨

```
      mat←2 3⍴⍳12      ⍝ A small matrix

      0 ⍨⍨ mat         ⍝ Always 0: Not VERY useful
0

      0 ⍨⍨⍨ mat        ⍝ 0 for each item of mat: (⍴mat)⍴0
0 0 0
0 0 0
```

v18 Operator: Constant α⍨⍨

```
      mat←2 3⍴⍳12      ⍝ A small matrix

      0 ⍨⍨ mat         ⍝ Always 0: Not VERY useful
0

      0 ⍨⍨⍨ mat        ⍝ 0 for each item of mat: (⍴mat)⍴0
0 0 0
0 0 0

      1 ¯1⍨⍨∘1⊢mat     ⍝ 2 columns with same number of rows as mat
```

v18 Operator: Constant α⍤⍬

```
      mat←2 3⍴⍳12      ⍝ A small matrix

      0 ⍤⍬ mat         ⍝ Always 0: Not VERY useful
0

      0 ⍤⍬⍤⍬ mat       ⍝ 0 for each item of mat: (⍴mat)⍴0
0 0 0
0 0 0

      1 ¯1⍤⍬⍤⍬1⊢mat    ⍝ 2 columns with same number of rows as mat
1 ¯1
1 ¯1
```

v18 Functions: Unique Mask ≠⍵

```
      ≠ 1 1 2 2 2 3  ⍝ Mask which would select unique items
1 0 1 0 0 1

      (≠,⊢) ↑2 3⍴'IBM' 'APL' ⍝ Works on Major Cells
1 IBM
0 IBM
1 APL
0 APL
0 APL
```

v18 Functions: Case Conversion ⎕C

```
      ⎕←A←'Hello' 'World' (,⍤⊆) ⎕NULL
```

Hello	World	[Null]

```
      ⎕C A   ⍝ ⎕C replaces 819⌶.
```

hello	world	[Null]

```
      ⍝ Note that non-character elements are preserved

      'hello' ≡⍤⎕C 'Hello' ⍝ "over ⎕C" means case insensitive
1
```

v18 Functions: Case Conversion ⎕C

v18 Functions: Case Conversion ⎕C

```
]box off
```

v18 Functions: Case Conversion ⎕C

```
]box off

1 ⎕C 'hello' 'world' ⍝ ⍺=1: Uppercase, 0: Lowercase
```

v18 Functions: Case Conversion ⎕C

```
      ]box off

      1 ⎕C 'hello' 'world' ⍝ ⍺=1: Uppercase, 0: Lowercase
HELLO  WORLD
```

v18 Functions: Case Conversion ⎕C

```
      ]box off

      1 ⎕C 'hello' 'world' ⍝ ⍺=1: Uppercase, 0: Lowercase
HELLO  WORLD

      1 0 ⎕C 'hello' 'world' ⍝ Not a scalar function
```

v18 Functions: Case Conversion ⎕C

```
      ]box off

      1 ⎕C 'hello' 'world' ⍝ ⍺=1: Uppercase, 0: Lowercase
 HELLO  WORLD

      1 0 ⎕C 'hello' 'world' ⍝ Not a scalar function
DOMAIN ERROR: Invalid left argument
```

v18 Functions: Case Conversion ⎕C

```
      ]box off

      1 ⎕C 'hello' 'world' ⍝ ⍺=1: Uppercase, 0: Lowercase
 HELLO  WORLD

      1 0 ⎕C 'hello' 'world' ⍝ Not a scalar function
DOMAIN ERROR: Invalid left argument

      1 (⎕C@(1 1)(2 1)) 'hello' 'world'
```

v18 Functions: Case Conversion ⎕C

```
      ]box off

      1 ⎕C 'hello' 'world' ⍝ ⍺=1: Uppercase, 0: Lowercase
 HELLO  WORLD

      1 0 ⎕C 'hello' 'world' ⍝ Not a scalar function
DOMAIN ERROR: Invalid left argument

      1 (⎕C@(1 1)(2 1)) 'hello' 'world'
 Hello  World
```

v18 Extensions: Where ⍸ and Partition ⊂

v18 Extensions: Where ⍸ and Partition ⊂

```
⍸1 0 4          ⍝ Where on positive integers
```

v18 Extensions: Where ⍸ and Partition ⊂

```
      ⍸1 0 4                ⍝ Where on positive integers
1 3 3 3 3
```

v18 Extensions: Where ⍸ and Partition ⊂

```
      ⍸1 0 4                ⍝ Where on positive integers
1 3 3 3 3
      (⍸⍤¯1) 1 3 3 3 3      ⍝ Inverse is "frequency count" (for ⍵ ascending)
```

v18 Extensions: Where ⍸ and Partition ⊂

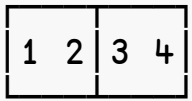
```
      ⍸1 0 4                ⍝ Where on positive integers
1 3 3 3 3
      (⍸⍤¯1) 1 3 3 3 3      ⍝ Inverse is "frequency count" (for ⍵ ascending)
1 0 4
```

v18 Extensions: Where ⍸ and Partition ⊂

```
      ⍸1 0 4              ⍝ Where on positive integers
1 3 3 3 3
      (⍸⍤¯1) 1 3 3 3 3    ⍝ Inverse is "frequency count" (for ⍵ ascending)
1 0 4

      1 0 1 0 ⊂ 1 2 3 4   ⍝ 1 starts a new partition: This has "always worked"
```

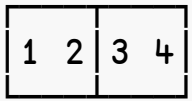
v18 Extensions: Where ⍸ and Partition ⊂

```
      ⍸1 0 4              ⍝ Where on positive integers
1 3 3 3 3
      (⍸⍤¯1) 1 3 3 3 3    ⍝ Inverse is "frequency count" (for ⍵ ascending)
1 0 4

      1 0 1 0 ⊂ 1 2 3 4   ⍝ 1 starts a new partition: This has "always worked"
```



v18 Extensions: Where ⍸ and Partition ⊂

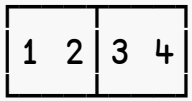
```
      ⍸1 0 4                ⍝ Where on positive integers
1 3 3 3 3
      (⍸⍤¯1) 1 3 3 3 3      ⍝ Inverse is "frequency count" (for ⍵ ascending)
1 0 4

      1 0 1 0 ⊂ 1 2 3 4     ⍝ 1 starts a new partition: This has "always worked"
```



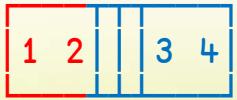

```
      1 0 3 0 ⊂ 1 2 3 4     ⍝ ⍺[i] decides how many elements are produced
```

v18 Extensions: Where ⍸ and Partition ⊂

```
      ⍸1 0 4              ⍝ Where on positive integers
1 3 3 3 3
      (⍸⍤¯1) 1 3 3 3 3    ⍝ Inverse is "frequency count" (for ⍵ ascending)
1 0 4

      1 0 1 0 ⊂ 1 2 3 4   ⍝ 1 starts a new partition: This has "always worked"
```




```
      1 0 3 0 ⊂ 1 2 3 4   ⍝ ⍺[i] decides how many elements are produced
```



v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

```
plens←2 0 0 3 0      ⍝ Partition Lengths
```

v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

```
plens←2 0 0 3 0      ⍝ Partition Lengths

+\¯1↓⎕IO,plens       ⍝ 1+Where empty cells need to go
```

v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

```
      plens←2 0 0 3 0      ⍝ Partition Lengths

      +\¯1↓⎕IO,plens       ⍝ 1+Where empty cells need to go
1 3 3 3 6
      (⍸⍤¯1) +\¯1↓⎕IO,plens
```

v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

```
      plens←2 0 0 3 0      ⍝ Partition Lengths

      +\¯1↓⎕IO,plens       ⍝ 1+Where empty cells need to go
1 3 3 3 6
      (⍸⍤¯1) +\¯1↓⎕IO,plens
1 0 3 0 0 1
```

v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

```
      plens←2 0 0 3 0      ⍝ Partition Lengths

      +\¯1↓⎕IO,plens       ⍝ 1+Where empty cells need to go
1 3 3 3 6
      (⍸⍤¯1) +\¯1↓⎕IO,plens
1 0 3 0 0 1

      1 0 3 0 0 1 ⊂ 'abcde'
```

v18 Extensions: Where ⍸ and Partition ⊂

Partitioning based on a partition length vector which contains zeros:

```
      plens←2 0 0 3 0      ⍝ Partition Lengths

      +\¯1↓⎕IO,plens       ⍝ 1+Where empty cells need to go
1 3 3 3 6
      (⍸⍤¯1) +\¯1↓⎕IO,plens
1 0 3 0 0 1

      1 0 3 0 0 1 ⊂ 'abcde'
┌──┬─┬─┬───┬─┐
│ab│ │ │cde│ │
└──┴─┴─┴───┴─┘
```

Date/Time Conversion: ⎕DT

Date/Time Conversion: ⎕DT

```
+2 ⎕NQ'.' 'DateToIDN' ⎕TS
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179

      ¯1 ⎕DT 43949.56179+0 1 ⍝ ¯1 = ⎕TS vector format
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179

     ¯1 ⎕DT 43949.56179+0 1 ⍝ ¯1 = ⎕TS vector format
```

2020 4 29 13 28 58 656	2020 4 30 13 28 58 656

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179

     ¯1 ⎕DT 43949.56179+0 1 ⍝ ¯1 = ⎕TS vector format
```

2020 4 29 13 28 58 656	2020 4 30 13 28 58 656

```
      3↑⎕FRDCI 1 1
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179

     ¯1 ⎕DT 43949.56179+0 1 ⍝ ¯1 = ⎕TS vector format
```

2020 4 29 13 28 58 656	2020 4 30 13 28 58 656

```
      3⊃⎕FRDCI 1 1
9.529003275E10
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179

     ¯1 ⎕DT 43949.56179+0 1 ⍝ ¯1 = ⎕TS vector format
```

2020 4 29 13 28 58 656	2020 4 30 13 28 58 656

```
      3⊃⎕FRDCI 1 1
9.529003275E10

      2 ¯1 ⎕DT 3⊃⎕FRDCI 1 1 ⍝ From 2 = FRDCI to ¯1 = ⎕TS
```

Date/Time Conversion: ⎕DT

```
      +2 ⎕NQ'.' 'DateToIDN' ⎕TS
43949

      1 ⎕DT ⊂⎕TS ⍝ Left argument is "target type": 1 = day number
43949.56179

     ¯1 ⎕DT 43949.56179+0 1 ⍝ ¯1 = ⎕TS vector format
```

2020 4 29 13 28 58 656	2020 4 30 13 28 58 656

```
      3⊃⎕FRDCI 1 1
9.529003275E10

      2 ¯1 ⎕DT 3⊃⎕FRDCI 1 1 ⍝ From 2 = FRDCI to ¯1 = ⎕TS
```

2020 4 29 13 33 32 500

Date/Time Conversion: ⎕DT

Date/Time Conversion: ⎕DT

```
⍉⍣¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
```

Date/Time Conversion: ⎕DT

```
      ⍉⍪1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911
```

Date/Time Conversion: ⎕DT

```
      ⍕⍣¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

      ↑⍣¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
```

Date/Time Conversion: ⎕DT

```
      ⍕⍣¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

      ↑⍣¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
2020 4 29 11 36 46 713
2020 4 29 13 36 46 713
```

Date/Time Conversion: ⎕DT

```
      ⍕¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

      ↑¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
2020 4 29 11 36 46 713
2020 4 29 13 36 46 713

      24×-/1 ⎕DT 'JZ' ⍝ Difference in hours between J = Local and Zulu
```

Date/Time Conversion: ⎕DT

```
      ⍕¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

      ↑¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
2020 4 29 11 36 46 713
2020 4 29 13 36 46 713

      24×-/1 ⎕DT 'JZ' ⍝ Difference in hours between J = Local and Zulu
2
```

Date/Time Conversion: ⎕DT

```
      ⍕¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

      ↑¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
2020 4 29 11 36 46 713
2020 4 29 13 36 46 713

      24×-/1 ⎕DT 'JZ' ⍝ Difference in hours between J = Local and Zulu
2

      ⍝ Date/Time formatting (experimental)
```

Date/Time Conversion: ⎕DT

```
      ⍕¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

      ↑¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
2020 4 29 11 36 46 713
2020 4 29 13 36 46 713

      24×-/1 ⎕DT 'JZ' ⍝ Difference in hours between J = Local and Zulu
2

      ⍝ Date/Time formatting (experimental)
      '__da__Dddd, DDoo mmmm YYYY; hh:mm:ss' (1200⌶) tn
```

Date/Time Conversion: ⎕DT

```
     ⍕¯1 ⎕DT 'Z' ⍝ Character Scalars represent Time Zones (Z = UTC)
2020 4 29 11 35 49 911

     ↑¯1 ⎕DT 'ZB' ⍝ Zulu (UTC) and Bravo (CET-Summer) time at ⎕TS
2020 4 29 11 36 46 713
2020 4 29 13 36 46 713

     24×-/1 ⎕DT 'JZ' ⍝ Difference in hours between J = Local and Zulu
2

     ⍝ Date/Time formatting (experimental)
     '__da__Dddd, DDoo mmmm YYYY; hh:mm:ss' (1200⌶) tn
  Onsdag, 13. februar 2019; 10:16:56
```

Dyalog version 18.0 - Datetime

help.dyalog.com/18.0/#Language/System%20Functions/dt.htm

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

DYALOG

Search for...

Contents

Time Numbers

If a value in X is positive it indicates that a time number type is expected in Y or generated in R, as follows. Note that the last column indicated whether (Yes) or not (No) negative numbers are allowed.

Group	Code	Description	Category	Date and time[1] represented by 0 (Epoch)	Negative values allowed?[8]
Dyalog APL	1	Dyalog Date Number	Day count with fractional part	1899-12-31 00:00	Yes
	2	Dyalog component file time	Tick count 1÷60s ticks[2]	1970-01-01 00:00	Yes
Other languages	10	J (J nanosecond time)	Tick count[3] 1ns ticks[2]	2000-01-01 00:00	Yes
	11	K7	Tick count 1ms ticks[2]	2024-01-01 00:00	Yes
	12	Javascript / D / Q	Tick count 1ms ticks[2]	1970-01-01 00:00	Yes
	13	R (R chron format)	Day count with fractional part	1970-01-01 00:00	Yes
UNIX	20	Unix time	Tick count 1s ticks[2]	1970-01-01 00:00	Yes
Microsoft	30	Microsoft DOS date/time	Encoded broken-down time 2s resolution	N/A	No
	31	Microsoft Win32 FILETIME	Tick count[3] 100ns ticks	1601-01-01 00:00	No

Dyalog version 18.0 - Datetime

help.dyalog.com/18.0/#Language/System%20Functions/dt.htm

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

DYALOG

Search for...

Contents

- Version 18.0 Key Feat
- Introducing Configura
- Multi-line Session Inp
- Extension to Where
- Extensions to Mix
- Regex Variant Option
- Serialising Namespac
- Load
- LX
- Bug Fixes
- Announcements
- Configuration
- Language Reference Cha
- Language Enhanceme
- Atop Operator
- Beside Operator
- Bind Operator
- Constant Operator
- Over Operator
- Unique Mask Functio
- Partitioned Enclose
- Case Convert System
- Datetime
- Format Date-time
- Write Text File

Application	42	Stata statistics package	1ms ticks[2]	1960-01-01 00:00	Yes
	43	SPSS statistics package	Tick count 1s ticks[2]	1582-10-14 00:00	No
	44	SAS	Tick count 1s ticks[2]	1960-01-01 00:00	Yes
Julian Date and variants	50	Julian Date	Day count with fractional part	−4717-11-24 12:00	No
	51	J (J dayno)	Day count with fractional part	1800-01-01 00:00	No
	52	Reduced Julian Date	Day count with fractional part	1858-11-16 12:00	Yes
	53	Modified Julian Date	Day count with fractional part	1858-11-17 00:00	Yes
	54	Dublin Julian Date	Day count with fractional part	1899-12-31 12:00	Yes
	55	CNES Julian Date	Day count with fractional part	1950-01-01 00:00	Yes
	56	CCSDS Julian Date	Day count with fractional part	1958-01-01 00:00	Yes
Decimal encoded[9]	60	Floating-point decimal encoded format Digits take the form yyyymmdd.hhmmss	Encoded broken-down time 1s resolution	N/A	No
	61	Integer decimal encoded format Digits take the form yyyymmddhhmmss(J digit time)	Encoded broken-down time 1s resolution	N/A	No

Version 18.0 – a MAJOR release!

Platform Equivalence

- .NET Core Bridge
- Launch Source Files
- Configuration Files
- HTMLRenderer + RIDE enhancements

Easier to Build, Test, Deploy

- Launch source files
- Configuration Files
- Multi-Line Input

Performance

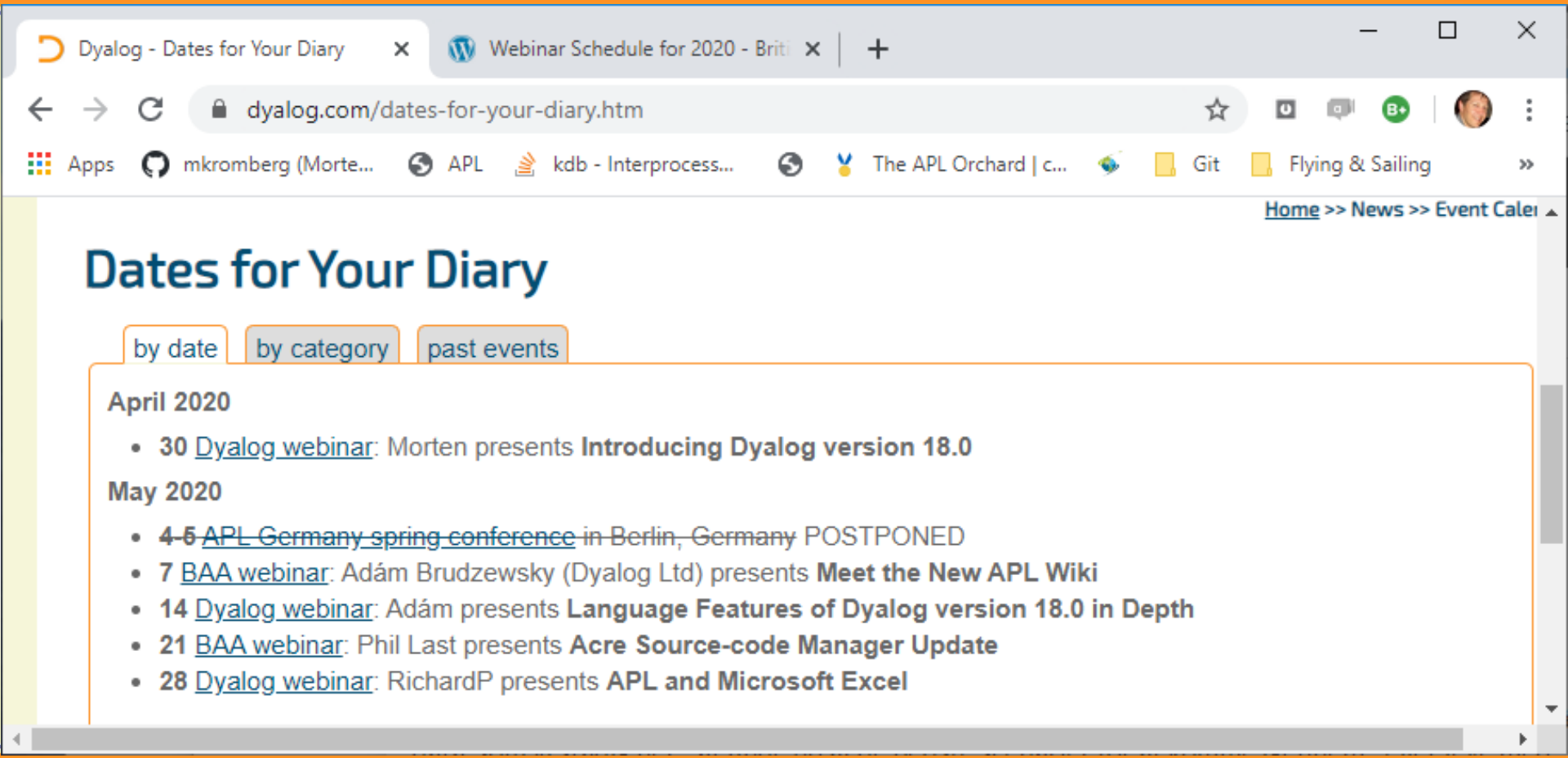
- "Performance QA" nearly 10% faster

Improving the Tool of Thought

- Operators ⍨ Constant, ⍤ Atop, ⍥ Over
- Function ≠ Unique Mask
- Case Folding with ⎕C
- Date Conversions with ⎕DT
- Multi-Line Input

Version 18.0 Availability

- In Beta Testing
- Release ”In May”


Dyalog - Dates for Your Diary
Webinar Schedule for 2020 - Brit
dyalog.com/dates-for-your-diary.htm
Apps
mkromberg (Morte...
APL
kdb - Interprocess...
The APL Orchard | c...
Git
Flying & Sailing
Home >> News >> Event Cale
Dates for Your Diary
by date
by category
past events
April 2020
30 Dyalog webinar: Morten presents Introducing Dyalog version 18.0
May 2020
4-5 APL Germany spring conference in Berlin, Germany POSTPONED
7 BAA webinar: Adám Brudzewsky (Dyalog Ltd) presents Meet the New APL Wiki
14 Dyalog webinar: Adám presents Language Features of Dyalog version 18.0 in Depth
21 BAA webinar: Phil Last presents Acre Source-code Manager Update
28 Dyalog webinar: RichardP presents APL and Microsoft Excel