

Dyalog Version 18.2



Dyalog version 18.2 was released in March 2022 and is supported under Microsoft Windows, IBM AIX, Linux (including the Raspberry Pi) and Apple macOS. This document discusses highlights of the release.

For additional information regarding the features mentioned in this document, see:

- the release notes for Dyalog v18.2 ([PDF](#))
- the release notes for User Commands v2.5 ([PDF](#))
- the list of issues resolved in the RIDE v4.4 ([GitHub](#))
- the list of enhancements in Link v3 ([GitHub](#))
- the licensing terms and conditions ([HTML](#))

Overview

Dyalog v18.2 contains many new features aimed at improving developer productivity and the ease with which solutions based on Dyalog APL can be deployed and maintained.

A significant non-technical feature of Dyalog v18.2 is the new "Basic" licence, which allows free re-distribution of Dyalog APL for non-commercial purposes as well as limited commercial use below an annual revenue cap.

Basic Licence replaces the Non-Commercial Licence

From Dyalog v18.2, the Non-commercial licence is replaced by a [Basic licence](#).

A Basic Licence is a free licence that allows APL users to have a copy of the latest Dyalog technology for personal or non-commercial use and experimentation. Common categories of non-commercial use include educational use, personal use, experimental use, proof of concept use, participating in programming competitions for prizes and just for fun use.

Holders of a Basic Licence can distribute Dyalog with their work under the terms of the [Royalty-Based Run-Time Licence](#), which will apply as the default run-time licence.

The Royalty-Based Run-Time Licence charges a royalty fee calculated as 2% of all gross revenue generated by sales, consulting and support of applications and end products developed using Dyalog, where such annual revenue is equal to or exceeds £5,000.

Launching Dyalog from Text Files

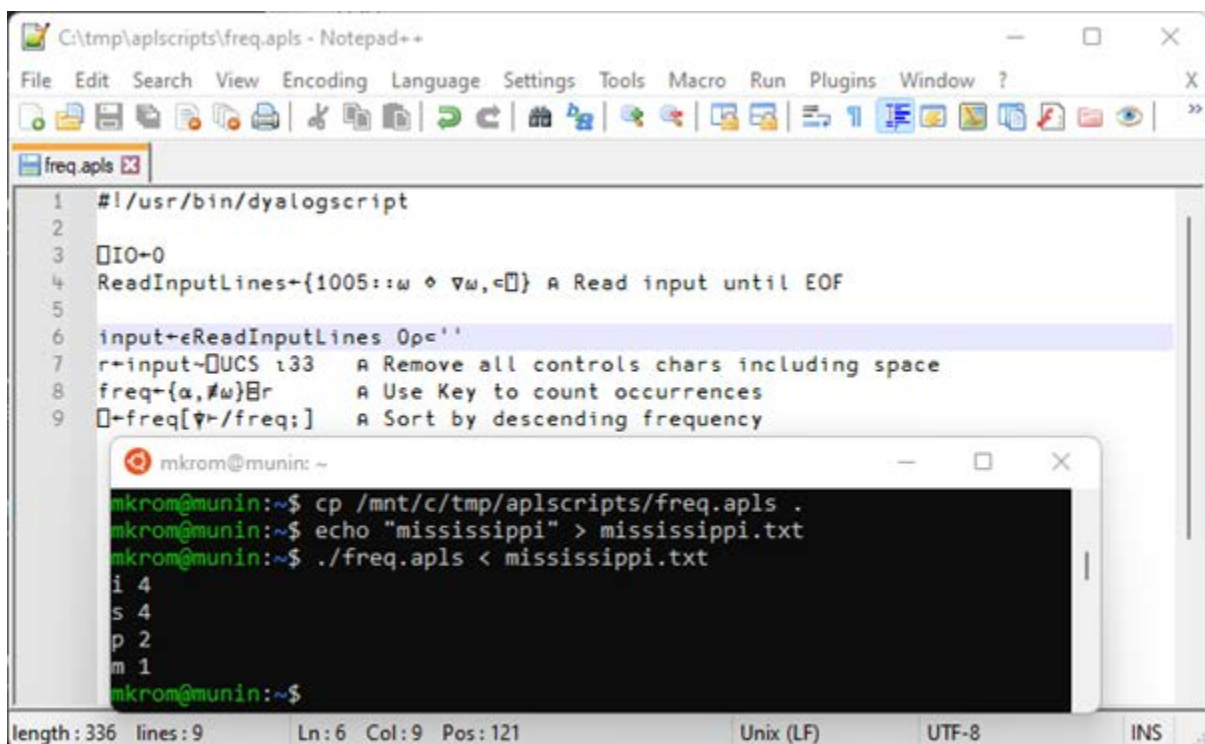
The most important new features of Dyalog v18.2 are related to the integration of Dyalog APL into modern development and deployment environments, frameworks, and tool chains. With Dyalog v18.2, Dyalog APL can be launched from text files:

- using a shell script (also called shebang or `#!` scripts) ([more...](#)).
- from a single text file containing the source for a function, namespace or class ([more...](#)).
- from a JSON configuration file, if the file contains a suitable expression in the LX parameter (Latent eXpression) ([more...](#)).

Support for shell scripts is new in Dyalog v18.2 (see below), and the support for other types of text has been extended. Binary workspaces will continue to be fully supported and recommended as a tool for deployment and debugging.

Shell Scripts

Shell scripts are widely used to run sequences of shell commands under Linux and similar operating systems. Shell scripts that begin with the symbols `#!` ("hash bang", or "shebang") followed by the name of an interpreter allow the rest of the script to be handled by the named interpreter rather than the default script processor. Dyalog v18.2 includes a "dyalogscript" interpreter, which means that you can now use APL as your script interpreter.



```
1 #!/usr/bin/dyalogscript
2
3 []IO-0
4 ReadInputLines-({1005::w ⍎ ∇w,c[]} A Read input until EOF
5
6 input←εReadInputLines 0p←''
7 r←input~[]UCS 133 A Remove all controls chars including space
8 freq-({α,#w)[]r A Use Key to count occurrences
9 []-freq[∇~/freq:] A Sort by descending frequency
```

```
mkrom@munin: ~
mkrom@munin:~$ cp /mnt/c/tmp/aplscripts/freq.apls .
mkrom@munin:~$ echo "mississippi" > mississippi.txt
mkrom@munin:~$ ./freq.apls < mississippi.txt
i 4
s 4
p 2
m 1
mkrom@munin:~$
```

length: 336 lines: 9 Ln: 6 Col: 9 Pos: 121 Unix (LF) UTF-8 INS

The screenshot shows a shell script containing APL statements that read input and display the frequency of occurrence of each letter that occurs in that input. The contents of the script file are being edited using the Notepad++ editor. In the foreground, a WSL (Windows Subsystem for Linux) session is visible; within the WSL session, a file called **mississippi.txt** is created, and on the next line the script is invoked with this file as input. The frequency counts are then returned.

Under Microsoft Windows, shell scripts run under PowerShell. On all platforms, the `dyalogscript` interpreter is only available with Unicode editions of Dyalog.

Text Source

Dyalog v18.2 includes [Link v3](#), which is a significant upgrade to Dyalog's tool for using text-based source. Link loads source into the active workspace from text files and synchronises changes made during the course of a development session in both directions. This means that changes made using the built-in editor are immediately reflected in the source files, and changes made using external editors or source code management systems are immediately replicated in the APL session.

In addition to Link, third-party tools based around text source files have started to appear. Examples include [Dado](#) and [Cider](#), which add project and workflow management features over the basic functionality provided by Link.

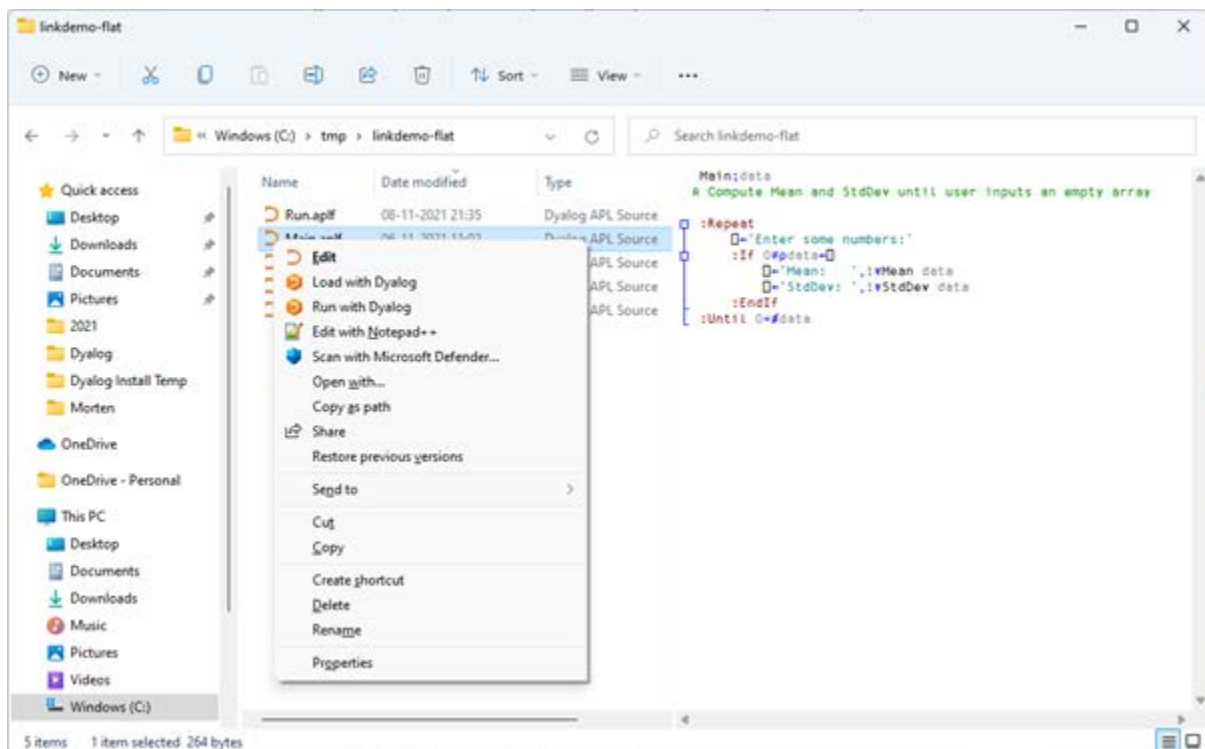
LOAD Parameter

Support for the [LOAD parameter](#) has been enhanced. In addition to a workspace or a text file defining a function, namespace, or class, the target can now be a directory. If the LOAD parameter identifies a directory, then Link loads the contents of the named directory into the root of the active workspace as the interpreter starts. If a function called **Run** is defined at the end of that process, it will be called.

This means that, if APL is started with **LOAD=/tmp/helloworld.aplf**, and that file contains the source for a function called **helloworld**, then the interpreter will define the function in the workspace and run it.

Improved Microsoft Windows Explorer Integration

The Dyalog v18.2 installer adds additional file associations to support loading and running APL source code directly from the Microsoft Windows Explorer. The screenshot below shows the menu that appears if you right-click or press Shift+F10 on a file containing the source of an APL function (screenshot taken using Microsoft Windows 11).



Within this menu, selecting **Edit** will open the stand-alone editor on the source file, **Load with Dyalog** will start a Dyalog session and load the source file (but not run it), and **Run with Dyalog** will invoke the code after loading it.

These file associations can be managed using the `]FileAssociations` user command, which has been significantly enhanced in Dyalog v18.2 and now also supports setting associations for users without administrative rights.

Improved Docker Support

Dyalog v18.2 can run without an attached terminal. This means that containers such as <https://hub.docker.com/r/dyalog/dyalog> no longer need to have a pseudo-terminal attached or require the use of the `-it` switch to keep stdin open.

By default, the public **dyalog/dyalog** container supports launching APL from a single source file and/or configuration file; this requires the default application folder (containing a single source file or configuration file) to be mapped into the container when it is started.

An important consequence of the new Basic Licence is that you can now share docker containers that are built upon our public containers (or embed Dyalog APL in other ways) without violating the terms of the licence – as long as the revenue cap is not exceeded and distributed containers include the LICENSE file in a prominent location.

The screenshot below shows how, when docker is available, a text file containing the source for an APL function can be run without installing Dyalog.

```
mkrom@mumin:~$ cat /mnt/c/tmp/helloworld/helloworld.aplf
helloworld
E←'Hello World!'

mkrom@mumin:~$ docker run -v /mnt/c/tmp/helloworld:/app dyalog/dyalog

DIALOG

https://www.dyalog.com

found aplf file Launching with LOAD=/app/helloworld.aplf
Dyalog APL/S-64 Version 18.2.45298
Serial number: UNREGISTERED - not for commercial use
+-----+
| Dyalog is free for non-commercial use but is not free software.
| A basic licence can be used for experiments and proof of
| concept until the point in time that it is of value.
| For further information visit
| https://www.dyalog.com/prices-and-licences.htm
+-----+
Fri Mar 4 14:29:33 2022
Loaded: #.helloworld from "/app/helloworld.aplf"
Hello World!
```

The RIDE v4.4

RIDE releases are independent of Dyalog releases, but the RIDE v4.4 is being released at roughly the same time as Dyalog v18.2 and is bundled with Dyalog on macOS. The RIDE v4.4 is another major step in the direction of closing the gap between the RIDE and the Microsoft Windows IDE. Important enhancements include:

- full support for debugging multi-threaded applications.
- saving responses to prompts that previously required user interaction each time.
- awareness of which version of Dyalog is being used when displaying the language bar and displaying language help.

RIDE issues are maintained on GitHub; a complete list of issues resolved in the new release is available [here](#).

Developer Productivity

Dyalog v18.2 includes many smaller enhancements that are aimed at making it easier to write applications in Dyalog APL. For example, Dyalog v18.2 introduces:

- support for formatting APL matrices with named columns as JSON arrays of objects ([more...](#)).
- a new system function, `⊞ATX`, which provides a one-stop-shop for metadata ([more...](#)).
- an experimental mechanism for generating non-linear random number distributions ([more...](#)).
- many new user commands, for example, `⌈APLCar t` and `⌈Get t`.

Visit the [Documentation Centre](#) for the documentation for Dyalog v18.2.