Dyalog Version 19.0

Dyalog version 19.0 was released in March 2024 and is supported under Microsoft Windows, IBM AIX, Linux for x64 and ARM (Raspberry Pi), and Apple macOS. This document discusses highlights of the release.

For additional information regarding the features mentioned below, see:

- the release notes for Dyalog v19.0 (PDF)
- the release notes for Link v4.0 (GitHub)
- the list of enhancements/fixes in Ride v4.5 (GitHub)

Highlights of Dyalog Version 19.0

Dyalog v19.0 makes it easier to install and maintain your APL system, and to develop and deploy applications built upon it. It:

- runs on new hardware and software platforms.
- helps to make your APL production systems more robust through:
  - improved error handling.
  - better support for collaboration between independent multi-threaded applications.
  - an experimental Health Monitor, which allows the monitoring of large collections of Dyalog processes.
- provides better support for external source code management systems and continuous integration pipelines.
- contains many other features to increase developer productivity, including a package manager.

64-bit ARM Support (macOS and Raspberry Pi)

Dyalog v19.0 adds native support for 64-bit ARM-based chips, including:

- Apple's M1, M2, and M3 chips (running macOS)

In addition, the following (which should run on most ARM-based Linux distributions) will be available later in 2024:

- Amazon Graviton chips, available from Amazon Web Services
- 64-bit Raspberry Pi running Pi OS BookWorm.

.NET 8.0 Support

The.NET interface is getting closer in capability to the .NET Framework interface. Most importantly, Dyalog v19.0 allows you to create .NET assemblies and executables using .NET (v8.0 – LTS version) in addition to the .NET Framework (v4.8).

NOTE: We expect to continue to support both .NET and the .NET Framework for the foreseeable future, although it is likely that some future functionality may not be provided for the .NET Framework. Although some productivity tools make use of .NET functionality, our policy is that the APL interpreter itself will never become dependent on .NET or any other application framework.

Production Systems in APL

Dyalog v19.0 provides better support for scaling your applications and constructing applications from components developed completely independently:

- Improved WS FULL handling ensures that your event handling will be successful even in situations where the interpreter has (almost) completely run out of memory.
- A new system function, ⎕TALLOC, allows independent components to safely use tokens for thread synchronisation without knowledge of each other.
- The interpreter exposes an experimental Health Monitor protocol that can be used to monitor the status of large collections of APL processes.

Tatin Packages

Dyalog v19.0 comes with built-in support for Tatin, the APL community's package manager. Most tools for APL developers that Dyalog implements in APL are now available as open-source repositories on GitHub, and an increasing number are also appearing as Tatin packages.

Although these packages are released independently of new releases of Dyalog, it is worth mentioning the following new packages that we have developed or enhanced during the Dyalog v19.0 release cycle:

- HttpCommand – an HTTP client for APL
- Jarvis – our Web Service framework
- NuGet – a tool for consuming .NET NuGet packages from Dyalog

You can monitor our progress at https://tatin.dev/v1/group-homepage?name=dyalog.

Source Code, Package Management, and Continuous Integration

Continuous Integration (CI) is a process in which changes to source code trigger automatic builds, running of tests, and, in some cases, deployment of new code. Dyalog v19.0 includes Link version 4.0, which allows you to base APL development on text files rather than binary workspaces; this makes it easy to make use of state-of-the art toolchains for source code management and CI.

Developer Productivity

As usual, Dyalog v19.0 contains many new features intended to make APL developers more productive. For example:

- `⎕NCOPY` and `⎕NMOVE` now provide an option for an APL callback function to be invoked during execution. This allows the programmer to monitor and/or report progress when processing a lot of data.
- The *Recurse* variant option for `⎕NINFO` has been extended to limit the level of sub-directories to be searched.
- `⎕NINFO` has been extended to provide file times as UTC Dyalog Date Numbers.
- File Components >2GB in size can now be compressed using LZ4 compression.
- `⎕FHOLD` now accepts an optional left argument to specify a time-out.
- The HTMLRenderer provides a number of new Properties and Methods – `AllowContextMenu`, `ExecuteJavaScript`, `GetZoomLevel`, `IsLoading`, `LoadEnd`, and `SetZoomLevel`.
- Multi-line input, which was introduced in Dyalog v18.0, is now enabled by default. This allows the definition of multi-line dfns directly in the APL session.
- Session log files are now saved in JSON format.
- Lines output to the Session that are associated with errors are now syntax-coloured using the error colour for the selected Session colour scheme.
- Ride v4.5, which is released with Dyalog v19.0, has many usability improvements including a redesigned startup screen.

Visit the Documentation Centre for the documentation for Dyalog v19.0.