

DYALOG

Eastbourne 2026

Programme for Dyalog '26



Monday 12 October – Friday 16 October 2026



Table of Contents

A Message from Our Managing Director	2
Team Dialog at Dialog '26	3
About Dialog '26	3
Evening Events.....	3
Schedule: Monday 12 October	4
Schedule: Tuesday 13 October	6
Schedule: Wednesday 14 October	8
Schedule: Thursday 15 October	10
Schedule: Friday 16 October	12
Abstracts: Workshops.....	14
Abstracts: Dialog Presentations.....	20
Abstracts: User Presentations	26
Practical Information	30

For practical information, see the back cover

If you have any questions not related to APL, please ask Karen



A Message from Our Managing Director

A very warm welcome to everyone joining us in Eastbourne for Dyalog '26.

These user meetings continue to be one of the highlights of the year for all of us at Dyalog Ltd. They provide a unique opportunity to share what we have been working on, discuss our plans for the future, and, most importantly, spend time face to face with the people who use Dyalog every day. While we communicate with many of you on a regular basis, nothing replaces the conversations, ideas, and connections that emerge when we gather together in person.

As usual, we look forward to telling you about the next phase of our development. Dyalog Ltd has grown significantly since the last user meeting on the technical, support, and administrative sides of the business, including a growing consulting team who can help you to modernise or build new applications in APL. Many members of the team who have been around for a while have new responsibilities. We will also share some of the work we are doing to reach new audiences through universities, competitions, conferences, and other community activities designed to help secure the future of APL and Dyalog.

We hope that you will use this week not only to learn about our plans but also to tell us about yours. Your feedback, ideas, successes, and challenges help shape our priorities and guide our future development. Please take every opportunity to speak with us, challenge us, and let us know how we can serve you better.

This year's user meeting will also provide a moment to remember Gitte Christensen. Gitte's contribution to Dyalog, to the APL community, and to the user meetings that she helped shape over so many years continues to be felt throughout our organisation and community.

I wish you a rewarding and enjoyable user meeting, and hope that you will leave with a renewed network, fresh inspiration, and perhaps that one idea, insight, or connection that will make a difference to you and your organisation in the years ahead.

Stine Kromberg, Managing Director (CEO)



Team Dyalog at Dyalog '26

This year, Dyalog Ltd is represented by 28 members of the current team. Meet Team Dyalog at <http://www.dyalog.com/meet-team-dyalog.htm>.

About Dyalog '26

Materials from the presentations and workshop will be available on the [Dyalog '26 webpage](#) after the user meeting.

We will be recording and publishing as many presentations as we can. We ask for your help in ensuring that question and answer sessions are also recorded; you can help by only asking questions from the microphone podium. If your question cannot wait until the Q&A session that concludes each presentation, or if the presenter specifically states that questions are welcome throughout, then please make your way to the microphone podium. Please do not ask questions from your seat as the recording cannot pick them up (and often people in the auditorium cannot hear them either).

Everyone from Dyalog Ltd will be happy to answer questions relating to their topics at any time during the user meeting.

Evening Events

Tuesday is quiz night! Join Jada and Karta for *Whyalog '26*, a fun, relaxed, brain-teasing quiz event. A Dyalog spin on the traditional pub quiz, with a blend of general knowledge, trivia, and puzzles. See if you can outsmart the Quiz Masters – we doubt it! Whyalog '26 is taking place in the Horizon Suite (mezzanine floor) after dinner.

On **Wednesday** there are "Birds of a Feather" discussion groups. These open, informal discussions are not presentations or problem-fixing sessions, but the opportunity to find and engage with others who share your interest in a subject area. Choose a topic you're interested in and look for the appropriate person in the Vista Bar after dinner.

NOTE: No laptop/stationery/equipment is needed for either evening.



Schedule: Monday 12 October

08:30 – 09:30	Registration Dyalog desk, <i>Breakout Area (first floor)</i>
09:30 – 13:00	Workshops (MA1) Web Services in APL: An Introduction to REST and OpenAPI <i>Pier Suite 1 (first floor)</i> (MA2) Juggling Arrays and Namespaces <i>Martello Suite (first floor)</i> (MA3) Concurrent Programming in Dyalog <i>Pier Suite 2 (first floor)</i>
13:00 – 14:00	LUNCH <i>Artisan Restaurant (mezzanine floor)</i>
14:00 – 17:30	Workshops (MP1) Performance from the Bottom Up <i>Martello Suite (first floor)</i> (MP2) Asynchronous Services: What, Why, and How <i>Pier Suite 2 (first floor)</i> (MP3) Juggling Arrays and Namespaces <i>Pier Suite 1 (first floor)</i>
17:30 – 19:00	Free time
19:00 –	DINNER <i>Artisan Restaurant (mezzanine floor)</i>

For workshop abstracts, see pages 14-19.



NOTES:



Schedule: Tuesday 13 October

All presentations take place in the *Central Hall (first floor)*

08:30 – 09:30	Registration, <i>Dyalog desk, Breakout Area (first floor)</i>
09:30 – 10:00	(D01) Welcome to Dyalog '26 Stine Kromberg, Managing Director (CEO)
10:00 – 10:45	(D02) The Road Ahead Morten Kromberg, Technical Director (CTO)
10:45 – 11:15	TEA/COFFEE
11:15 – 11:45	(D03) Slowly Getting Faster Asher Harvey-Smith
11:45 – 12:15	(D04) Improving APL Threads Peter Mikkelsen
12:15 – 12:45	(U01) Think in Arrays: Design, Iterate, and Build in the Age of AI Dave Thomas (Bedarra Corporation)
12:45 – 13:45	LUNCH <i>Artisan Restaurant (mezzanine floor)</i>
13:45 – 14:15	(D05) Dyalog and AI Stefan Kruger
14:15 – 14:45	(U02) Dyalog in the Global Investment Management Architecture Oliver Lanz (SimCorp)
14:45 – 15:15	(D06) Inside Dyalog Labs – From Prototype to Project Martina Crippa
15:15 – 15:45	TEA/COFFEE
15:45 – 16:15	(U03) Coming Up for Air: What I'd Like to See from Dyalog Gilgamesh Athoraya (Tiamatica AB)



16:15 – 16:50	(D07) “Cooler Than Keanu” John Daintree
16:50 – 18:30	Free time
18:30 – 20:00	DINNER <i>Artisan Restaurant (mezzanine floor)</i>
20:00 –	Whyalog '26 <i>Horizon Suite (mezzanine floor)</i>

For Dyalog presentation abstracts, see pages 20-25.
For user presentation abstracts, see pages 26-29.

NOTES:



Schedule: Wednesday 14 October

All presentations take place in the *Central Hall (first floor)*

09:30 – 10:00	(D08) Web Services Overview Brian Becker
10:00 – 10:30	(D09) Making Web Services More Accessible Holden Hoover
10:30 – 11:10	(D10) Time's Up Richard Smith
11:10 – 11:40	TEA/COFFEE
11:40 – 12:10	(D11) Dyalog Static Analysis Tool – In Practice and the Future Brandon Wilson
12:10 – 12:40	(U04) Why CI is (Also) Important in the Age of D-AI-alog Lars Stampe Villadsen (SimCorp A/S)
12:40 – 13:40	LUNCH Artisan Restaurant (mezzanine floor)
13:40 – 13:50	(D12) The APL Forge – Introduction and Prize Ceremony Stine Kromberg
13:50 – 14:20	(U05) The APL Forge (Winner) – APLearn, Machine Learning in APL Borna Ahmadzadeh (York University)
14:20 – 14:50	(U06) The APL Forge (Winner) – Forging Voxels with Dyalog APL Kyle Croarkin (independent)
14:50 – 15:20	TEA/COFFEE
15:20 – 15:50	(U07) The APL Trust – An Update Diane Hymas (independent)



15:50 – 16:20	(U08) The Next APL Font Family Madeline Vergani (independent)
16:20 – 16:50	(U09) Trying to Make a Point: APL Glyphs in Action Katya Solstad (independent)
16:50 – 19:00	Free time
19:00 – 20:30	DINNER <i>Artisan Restaurant (mezzanine floor)</i>
after dinner	"Birds of a Feather" discussion groups <i>Vista Bar (ground floor)</i>

For Dyalog presentation abstracts, see pages 20-25.

For user presentation abstracts, see pages 26-29.

NOTES:



Schedule: Thursday 15 October

All presentations take place in the *Central Hall (first floor)*

09:30 – 10:00	(D13) Behind the Curtain: The Creation of a New Operator Silas Poulson
10:00 – 10:40	(U10) Claude is the New Quad Mark Wolfson (Buyers Intelligence Group – BIG)
10:40 – 11:10	(D14) EWC in 2026 Aarush Bhat
11:10 – 11:40	TEA/COFFEE
11:40 – 12:10	(D15) Ten years Without Workspaces Adám Brudzewsky
12:10 – 12:40	(U11) The Newcomer's Experience: Creating a Video Game in APL Mikaela Hsu (independent)
12:40 – 13:10	(D16) APL Media Update 2026 Rich Park
13:10 – 14:10	LUNCH <i>Artisan Restaurant (mezzanine floor)</i>
14:10 – 14:40	(U12) Tacit in Practice Max Sun (InvestCloud)
14:40 – 15:10	(D17) Co-dfns Update (v6) Aaron Hsu
15:10 – 15:40	(D18) One Extension and Seven New Primitives Adám Brudzewsky
15:40 – 16:10	TEA/COFFEE
16:10 – 16:40	(D19) Testing the Untestable – GUIs, Fuzzers, and Dyalog APL Aarush Bhat



16:40 – 17:10	(D20) Plots and Tables: A Modern Approach Jesús Galán López
17:10 – 17:15	(D21) Closing Session Stine Kromberg
17:15 – 19:00	Free time
19:00 –	CLOSING DINNER Horizon Suite (mezzanine floor)

For Dyalog presentation abstracts, see pages 20-25.

For user presentation abstracts, see pages 26-29.

NOTES:



Schedule: Friday 16 October

09:30 – 13:00	Workshops (FA1) Beyond the Primitives: Using Iteration and Recursion in APL <i>Martello Suite (first floor)</i> (FA2) Problem Solving with APL <i>Pier Suite 2 (first floor)</i>
13:00 – 14:00	LUNCH <i>Artisan Restaurant (mezzanine floor)</i>
14:00 – 17:30	Workshops (FP1) Performance from the Bottom Up <i>Pier Suite 2 (first floor)</i> (FP2) Introduction to Link <i>Pier Suite 1 (first floor)</i> (FP3) Problem Solving with APL <i>Martello Suite (first floor)</i>

For workshop abstracts, see pages 14-19.



NOTES:



Abstracts: Workshops

These workshops have significant time set aside for participants to experiment with the subject matter.

Some of the workshops require materials that should be installed on your laptop before attending. These pre-requisites are listed by workshop at <https://usermeeting.dyalog.com/workshops/prerequisites/>. Please ensure that you meet these pre-requisites before attending a workshop.

MA1: Web Services in APL: An Introduction to REST and OpenAPI

Monday 09:30 – 13:00

A RESTful web service enables different applications to exchange data and functionality over the internet using a standardised request/response model. [OpenAPI](#) is a machine-readable way to describe such services, and has become the unofficial standard for doing so – from a single specification, tools can automatically generate client code, documentation, and tests. Thousands of public web services publish OpenAPI specifications; this hands-on workshop shows you how you can easily build APL clients for any of them using [Dyalog's OpenAPI tool](#).

In this workshop, you will:

- develop an understanding of REST and OpenAPI. An explanation of what makes a web service "RESTful" and how an OpenAPI specification describes one in a form that both humans and tools can read.
- consume web services from APL. Using Dyalog's OpenAPI tool, you'll automatically generate an APL client from the OpenAPI specification of a public RESTful webservice and make real requests from an APL session.
- provide OpenAPI-compatible web services from APL by building your own REST service using [Jarvis](#) together with [Stark](#) (a router for Jarvis that conveniently publishes its own OpenAPI specification) then exercise the service end-to-end, reading from and writing to it from a separate APL session using an automatically generated client.



After this workshop you should feel confident consuming third-party REST APIs from APL and exposing your own APL code as a web service that other tools and languages can consume.

Keywords: *Web services, REST, OpenAPI, Stark, Jarvis*

MA2: Juggling Arrays and Namespaces

Monday 09:30 – 13:00

NOTE: This workshop is repeated as MP3.

Dyalog v20.0 made it significantly easier and safer to work with arrays and namespaces: array notation lets you write array literals, while the new system functions `⊖VGET` and `⊖VSET`, together with the enhanced `⊖NS`, let you read, write, and merge values by name, replacing most uses of *execute* (`⌞`).

In this hands-on workshop, you will explore the new syntactic forms and put them to work.

Keywords: *array notation, namespaces, exercises*

MA3: Concurrent Programming in Dyalog

Monday 09:30 – 13:00

APL's array-orientation means that is it an inherently data-parallel notation. For performing the same operation on multiple elements (SIMD), there's not much more you need. However, for expensive computations, calls to external services, or computing while your system continues to service external requests, you might want run multiple different functions at the same time, or run functions in different CPU processes or even different machines at the same time.

This workshop looks at the two main features of Dyalog made for concurrent programming: the *spawn* operator (`&`) and its associated `⊖Txxx` system functions; and the **isolate.dws** workspace. You will learn how to run functions in parallel, strategies for managing multiple threads, and how you can collect results.

Keywords: *performance, optimisation, concurrency, parallel language features*



MP1: Performance from the Bottom Up

Monday 14:00 – 17:30

NOTE: This workshop is repeated as FP1.

Many APL applications have a reputation for tremendous efficiency and performance, but users might be surprised to discover that their code does not run as fast as they would like, or that it uses too much memory. Because APL isn't like other languages, it's important to understand how to get the most from APL's unique design to maximise your performance while minimising your programming efforts.

In this workshop you will gain a strong foundation in using the APL primitives fluently to write performant programs. After going through the fundamentals of how the interpreter executes APL code, you will use that knowledge to build up a library of performance techniques. You will also have the opportunity to apply your knowledge with hands-on exercises and discussion.

Keywords: *Performance, algorithms, optimisation, array model*

MP2: Asynchronous Services: What, Why, and How

Sunday 14:00 – 17:30

The traditional request-response model has a fundamental limitation: the client must always speak first. This makes it awkward (and sometimes impossible) to build applications where the server needs to push updates, multiple parties exchange messages in real time, or work continues long after the initial request. Asynchronous web services solve this problem, and they're the backbone of dashboards, stock tickers, news feeds, multiplayer games, chat rooms, and long-running computations.

This hands-on workshop covers the three dominant techniques – [WebSockets](#), [Server-Sent Events \(SSE\)](#), and [webhooks](#) – and the Dyalog tools that bring them within easy reach: [HttpCommand](#), [Jarvis](#), [WebSocketClient](#), and [WebSocketServer](#).



In this workshop, you will:

- develop an understanding of when to use WebSockets, SSE, and webhooks, and how they differ.
- gain hands-on experience with Dyalog's asynchronous toolkit, that is, `HttpCommand`, `Jarvis`, `WebSocketClient`, and `WebSocketServer`.
- consume existing asynchronous services from APL.
- build and deploy your own asynchronous service.

Keywords: *Webhooks, WebSockets, Asynchronous, Jarvis, HttpCommand, WebSocketServer, WebSocketClient*

MP3: Juggling Arrays and Namespaces

Sunday 14:00 – 17:30

NOTE: This workshop is a repeat of workshop MA2.

Dyalog v20.0 made it significantly easier and safer to work with arrays and namespaces: array notation lets you write array literals, while the new system functions `⊖GET` and `⊖SET`, together with the enhanced `⊖NS`, let you read, write, and merge values by name, replacing most uses of `execute` (Φ).

In this hands-on workshop, you will explore the new syntactic forms and put them to work.

Keywords: *array notation, namespaces, exercises*

FA1: Beyond the Primitives: Using Iteration and Recursion in APL

Friday 09:30 – 13:00

Traditional programming and idiomatic APL written in a data-parallel style can sometimes feel incompatible and impossible to use together. Dyalog APL provides a number of features around traditional iteration that can be used to great effect or that can greatly undermine APL's biggest advantages. Understanding how traditional concepts like iteration and recursion fit into the idiomatic APL landscape can greatly improve the architectural clarity and maintainability of your code, as well as giving you a better overview of large APL applications.



In this workshop you will explore patterns for writing iterative, recursive, and stateful code in APL. There will be a focus on how to take best advantage of the many constructs for control flow available in Dyalog, such as structured statements, labels, and recursion; you will learn how to use these features to write programs that are easy to read, debug, and maintain while avoiding the pitfalls that can undermine the quality of your code.

Keywords: *Control flow, recursion, loops, idioms*

FA2: Problem Solving with APL

Friday 09:30 – 13:00

NOTE: This workshop is repeated as FP3.

APL users have access to a wide range of techniques to solve any given problem; the terseness and versatility of APL means that different approaches can be tried to see what works best in a particular scenario. In this workshop, you will solve a variety of problems; we will review our solutions together to enhance our problem solving skills.

Keywords: *problem solving, exercises*

FP1: Performance from the Bottom Up

Friday 14:00 – 17:30

NOTE: This workshop is a repeat of workshop MP1.

Many APL applications have a reputation for tremendous efficiency and performance, but users might be surprised to discover that their code does not run as fast as they would like, or that it uses too much memory. Because APL isn't like other languages, it's important to understand how to get the most from APL's unique design to maximise your performance while minimising your programming efforts.

In this workshop you will gain a strong foundation in using the APL primitives fluently to write performant programs. After going through the fundamentals of how the interpreter executes APL code, you will use that knowledge to build up a library of performance techniques. You will then apply your knowledge with hands-on exercises and discussion.

Keywords: *Performance, algorithms, optimisation, array model*



FP2: Introduction to Link

Friday 14:00 – 17:30

Most actively-maintained APL systems have switched or are switching to using text files for source, rather than workspaces or component files. This allows integration with source code management systems, build and test frameworks, and of course to experiment with "agentic AI". If you have not yet experienced the benefits of text-based source, or you understand the benefits but need help to get started, this workshop is for you.

[Link](#) allows you to use Unicode text files to store APL source code, rather than "traditional" binary workspaces. The workshop includes:

- how to find and use the documentation.
- the types of APL objects that can be handled, and how each is represented as text.
- how Link synchronises changes made inside and outside the active workspace.
- migrating an existing workspace to text-based source.
- using GitHub to manage "Linked" source code.

Keywords: *text files, text-based source, source code management, Link*

FP3: Problem Solving with APL

Friday 14:00 – 17:30

NOTE: This workshop is a repeat of workshop FA3.

APL users have access to a wide range of techniques to solve any given problem; the terseness and versatility of APL means that different approaches can be tried to see what works best in a particular scenario. In this workshop, you will solve a variety of problems; we will review our solutions together to enhance our problem solving skills.

Keywords: *problem solving, exercises*



Abstracts: Dyalog Presentations

D01: Welcome to Dyalog '26

Stine Kromberg, Managing Director (CEO)

Dyalog Ltd's Managing Director opens the user meeting.

D02: The Road Ahead

Morten Kromberg, Technical Director (CTO)

In accordance with tradition, Morten looks briefly back over his shoulder before turning his gaze to the future, presenting his view of the road that lies before Dyalog Ltd and users of Dyalog, and setting the scene for the technical presentations that follow.

D03: Slowly Getting Faster

Asher Harvey-Smith

One of the promises of array programming is fast execution of algorithms written in a high-level notation. Asher describes how performance improvements planned for the next Dyalog releases continue to deliver on this promise.

D04: Improving APL Threads

Peter Mikkelsen

APL threads can be a convenient way to structure concurrent applications. As the use of threading has grown, the implementation has started to show signs of strain, and issues can become apparent as the number of simultaneous threads increases. Peter presents some of the work being undertaken that aims to improve the performance and behaviour of APL thread switching, and the thread related language features such as `:Hold` and `⌈TGET`.

D05: Dyalog and AI

Stefan Kruger

Large Language Models (LLMs) are having significant impacts on our industries, in both good and bad ways. It's a fast-moving field. Stefan's presentation is a "state of the LLM nation" snapshot of how we as APLers can take advantage of the good parts of what LLMs can do, and how we can



avoid some of the downsides. He demonstrates how to enable communication between LLMs and the APL session, and takes us on a tour of the current models, tools, and techniques.

D06: Inside Dyalog Labs – From Prototype to Project

Martina Crippa

[Dyalog Labs](#) is an open-source initiative for tools, libraries, and experimental projects written in and for Dyalog APL. It has been established in response to new ways of developing software, and provides a place where ideas can be prototyped and turned into real projects.

Martina introduces the initiative, showcases some of the projects developed so far, and discusses the role of Dyalog Labs in growing the ecosystem of modern tools for the APL community.

D07: “Cooler Than Keanu”

John Daintree

John is a big fan of "Dark Mode"; he thinks it is very cool and has enabled it on all his devices. Unfortunately, Dyalog (and, by extension, your applications) doesn't have a "Dark Mode". What can we do? How cool can we make Dyalog APL? (Hint: there's a clue in the title).

D08: Web Services Overview

Brian Becker

As applications increasingly need to interact with other systems, data sources, and cloud-based services, web services have become a standard mechanism for communication. The Dyalog ecosystem provides a range of tools for consuming and exposing web services, enabling APL applications to participate in modern, distributed architectures and integrate seamlessly with the broader software landscape. In this presentation, Brian discusses the capabilities and common use cases of Jarvis, HttpCommand, and WebSockets.



D09: Making Web Services More Accessible

Holden Hoover

REST (Representational State Transfer) is the most widely used architectural style for web service APIs, allowing applications to exchange data and perform operations using standard HTTP protocols. OpenAPI is an industry-standard, machine-readable, specification for describing REST APIs, making it possible to automatically generate documentation, client libraries, testing tools, and other integration assets. Holden introduces the Dyalog OpenAPI Client Generator, which can generate APL client interfaces directly from OpenAPI specifications, greatly simplifying access to thousands of existing web services. He also introduces *Stark*, a framework that streamlines the development of Jarvis REST services and can automatically generate OpenAPI specifications for them, making those services easier to document, discover, and consume.

D10: Time's Up

Richard Smith

The I-Beam function that formats date-times as text has been merged into `⌈DT` in Dyalog v21.0 and will be removed from a future Dyalog release. Richard discusses why the functions have been separate until now and the various enhancements that have been made to `⌈DT`. He also explains why and when features are deprecated and removed, and demonstrates a tool to help you prepare for when this happens.

D11: Dyalog Static Analysis Tool – In Practice and the Future

Brandon Wilson

Dyalog's Static Analysis Tool (DSAT) now enjoys real world usage. Brandon presents an overview of DSAT in practice and discusses what its future may look like. In particular, he looks at Continuous Integration/Continuous Delivery/Distribution checks, session integration, an emerging VSCode extension, and applications to AI tooling.



D12: The APL Forge – Introduction and Prize Ceremony

Stine Kromberg

What is [The APL Forge](#)? How do you compete, and who won the 2026 and 2025 rounds? Stine introduces this competition and discusses some upcoming changes for the 2027 round, culminating in the award ceremony for the winners.

D13: Behind the Curtain: The Creation of a New Operator

Silas Poulson

Dyalog v20.0 introduced a new operator, [behind](#) (`⋈`). Silas explains some of the processes involved in adding a new language element like an operator, using the *behind* operator and the lessons learnt during its implementation as a reference.

D14: EWC in 2026

Aarush Bhat

EWC is Dyalog's cross-platform implementation of the `⌈WC` GUI family. Your APL applications are constructed using the same forms as under `⌈WC`; EWC renders them as a real GUI on Linux, macOS, or Microsoft Windows, either in a desktop window or a browser tab.

Aarush demonstrates how EWC works; his demonstration includes examples of applications that have already been re-platformed without changes and others that have been built using EWC from the start. He explains what's changed since [Morten's Dyalog '24 presentation on EWC](#), and shares the plan for the future.

D15: Ten Years Without Workspaces

Adám Brudzewsky

The workspace is 60 years old, and at Dyalog '18 [Paul Mansour declared it dead](#). Adám reviews the decade since. He traces the history of APL source – from binary workspaces and `⌈ORs` in component files, through transfer files and SALT's scripted namespaces, to Link, Acre, Cider, Tatin, and Git – and shows how the interpreter has taken over jobs that once needed a bootstrap workspace. He considers what the workspace is still good for, and looks ahead to modules, single-file source, and source handling built into the interpreter.



D16: APL Media Update 2026

Rich Park

It can be difficult for professional users to make time to learn about everything happening in the world of Dyalog. The user meeting offers a great opportunity, but there's no need to wait for the next one to learn about new features, projects, and techniques. Rich provides an overview of what APLers have been saying and doing in this whirlwind tour of APL community projects and publications.

D17: Co-dfns Update (v6)

Aaron Hsu

[Co-dfns](#) is an APL compiler that is designed to let a programmer use APL in more places and for more applications. When compared with the standard Dyalog interpreter, it can enable better integration with other programming environments and high-performance domains (such as scientific computing, machine learning, and graphics), and improved performance in certain situations. A lot has been happening with the Co-dfns compiler since Aaron's last update at Dyalog '24; he introduces the new version 6 release and provides a roadmap of what's next on the horizon for APL compilation.

D18: One Extension and Seven New Primitives

Adám Brudzewsky

Many everyday APL expressions are longer than they need to be. Adám presents a draft proposal for eight small language changes, each of which shortens a common idiom: monadic $\leq$ and $\geq$ to sort ascending and descending, monadic $\wedge$ and $\vee$ to promote and demote rank, monadic τ and $\perp$ to encode and decode binary, monadic $\geq$ (a companion to monadic $\Rightarrow$) to pick the last, and an extension to dyadic ρ allowing one axis length to be inferred. He demonstrates each in use and invites discussion.

D19: Testing the Untestable – GUIs, Fuzzers, and Dyalog APL

Aarush Bhat

Every Dyalog release is preceded by a lot of testing that users never see. Aarush introduces through the test suites that run continuously against the interpreter: the long-standing unit tests, ullu's primitive-by-primitive tests, coverage-instrumented builds that report exactly which lines of the interpreter's C source each suite exercises, and fuzzing, which compiles an



instrumented interpreter and then spends entire weekends inventing APL expressions designed to crash it. Crashes are automatically minimised, deduplicated, and added to a growing body of tests that are replayed against every platform and edition on later builds.

He then introduces CITA, which lets our APL tools be tested against many interpreter versions, editions, and platforms on real hardware.

Aarush concludes by looking at the testing of GUIs, which have always been the part everyone struggles to automate. He demonstrates how EWC and `□WC` are now covered by several hundred tests with pixel-tight visual regression running on every pull request. Much of this suite was written by LLM agents exploring the running application, and he concludes with his thought on this approach might change the way in which test suites are built and maintained.

D20: Plots and Tables: A Modern Approach

Jesús Galán López

It is usually convenient to represent data visually, for example, as a table or as some kind of technical plot, and Dyalog already offers several options for doing this. Jesús introduces a different approach that uses some of Dyalog APL's features: user commands are organised in Tatin packages, configuration options and data are defined as namespaces, and external JavaScript libraries take care of the actual rendering. As a lighter option, a text-based alternative renderer is provided, which makes it possible to display simple plots and tables directly in your session in any environment.

D21: Closing Session

Stine Kromberg

Dyalog Ltd's Managing Director closes the user meeting.



Abstracts: User Presentations

U01: Think in Arrays: Design, Iterate, and Build in the Age of AI

Dave Thomas, [Bedarra Corporation](#) (Canada)

Dave presents a brief review of the essentials of array thinking and its impact before discussing how array practitioners move from thinking to program design and implementation. Jumping forward to the current Age of AI, he explores the role of arrays – what some call 'Iverson's Ghost', visible today in NumPy and PyTorch – in today's Machine Learning (ML) and Large Language Models (LLMs). This leads naturally into approaches for the generative development of array programs. Dave concludes with a discussion of promising compiler and hardware research.

U02: Dyalog in the Global Investment Management Architecture

Oliver Lanz, [SimCorp A/S](#) (Denmark)

Dyalog Ltd has been SimCorp's trusted partner for core processing throughout all major technology transitions. Oliver explains what SimCorp do in the context of their latest transitions from desktop applications to cloud-native backend-systems and the emergence of useful AI use cases.

U03: Coming Up for Air: What I'd Like to See from Dyalog

Gilgamesh Athoraya, [Tiamatica AB](#) (Sweden)

After five years focused on a large migration project, Gilgamesh takes a step back and looks at Dyalog with fresh eyes. He shares his personal wishlist of features and improvements that would make life easier for developers and help keep Dyalog relevant in a rapidly-changing software landscape.

U04: Why CI is (Also) Important in the Age of D-AI-a-log

Lars Stampe Villadsen, [SimCorp A/S](#) (Denmark)

Lars explores how Continuous Integration (CI) discipline helps you in agent (and human) driven development, and explains why local execution and procedures should be built to steer both humans and agents.

**U05: The APL Forge (Winner) – APLearn, Machine Learning in APL**

Borna Ahmadzadeh, student at [York University](#) (Canada)

In recent years, the world of artificial intelligence has been dominated by Generative Artificial Intelligence (GenAI). However, traditional models like linear regression and decision trees remain the powerhouse behind machine learning (ML) applications dealing with structured data such as tables. Borna demonstrates [APLearn](#), an APL library implementing common ML algorithms for a variety of tasks such as supervised learning, clustering, and dimensionality reduction. He explores array programming's relevance in the context of ML, its unique advantages and occasional drawbacks, and trains a couple of models on real data to get a taste for what ML in APL might look like in practice.

U06: The APL Forge (Winner) – Forging Voxels with Dyalog APL

Kyle Croarkin, independent (U.S.A.)

Kyle has developed a project utilising APL to make a Minecraft-like game called "[A Voxel Game](#)". He presents an overview of his project, describes his experiences as he went through the process of game development in APL, and explains how APL contributed to different parts of the architecture.

U07: The APL Trust – An Update

Diane Hymas, independent (U.S.A.)

Inspired by the vision of Gitte Christensen, [The APL Trust US](#) was established as a non-profit organization in 2025 to support the growth and impact of APL in education, research, science, engineering, and mathematics. Diane provides a brief update on the APL Trust's background, mission, current status, and early grant-funded projects. Whether you are new to the APL Trust or have followed its progress from the beginning, she invites you to learn how this initiative is helping support innovative APL work and how you can take part in its future.



U08: The Next APL Font Family

Madeline Vergani, independent (Italy)

Madeline talks about her work on the [APL387 font](#), focusing on the most interesting design and technical choices and her hopes about the future of the font. She is particularly interested in community opinions on perhaps the most important part of the font family...its name!

U09: Trying to Make a Point: APL Glyphs in Action

Katya Solstad, KFT / The Book of Glyphs / Applied Meanings (U.S.A.)

Katya's presentation will unfold not as an explanation but as a theatrical performance in which the glyphs leave the keyboard.

Iverson called notation a tool for thought – this presentation goes to the thinking itself.

U10: Claude is the New Quad

Mark Wolfson, [Buyers Intelligence Group \(BIG\)](#) (U.S.A.)

Rich Park, [Dyalog Ltd](#) (U.K.)

APL enables knowledgeable people to express solutions as simple, direct, code. In an ideal world, we could live in the Session, elegantly transforming from array to array. In the real world, we often need additional expertise to interface with other technologies, to utilise existing solutions, or to make our own accessible to less technical end users.

In APL, quad (⎕) denotes the machine and can be thought of as the window to the outside world. System functions have long been the primary mechanism for that interfacing. At their best, they are simple enough for a domain expert to build an entire system unaided. However, their limits are the implementor's limits, and beyond those you still need to know the external technology. Large Language Model (LLM)-assisted development is now lowering that barrier; Rich and Mark demonstrate examples of APL development and interfacing work accelerated by LLM assistance.

U11: The Newcomer's Experience: Creating a Video Game in APL

Mikaela Hsu, independent (U.S.A.)

Mikaela describes the process, difficulties, and upsides of learning APL, as well as the mindset behind learning to apply artistic creativity to technology while creating a video game.



U12: Tacit in Practice

Max Sun, [InvestCloud](#) (Canada)

Tacit programming is a paradigm in which the programmer, rather than writing a dfn or tradfn that explicitly references arguments, builds a single derived function and relies on operators and trains to control the flow of arguments. While purely tacit programming is best used only for recreational confusion, everyone uses derived functions. Max demonstrates common idioms that he uses – split, inspect, read, write, and select. He explains how they're useful to him, and summarises the tools and resources that help him to write tacit code.

Trademarks

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

Oracle® and Java™ are registered trademarks of Oracle and/or its affiliates.

macOS® and OS X® (operating system software) are trademarks of Apple Inc., registered in the U.S. and other countries.

All other trademarks are the property of their respective owners.



Practical Information

To make your stay as pleasant as possible, we would like to draw your attention to some practical details.

Check In and Out

Your room is available from **15:00** on the day of arrival; you must check out by **10:30** (weekdays) and **12:00** (weekends) on the day of departure.

Internet Access

WiFi is available in all areas and is free of charge.

Meals and Drinks

The user meeting fee includes all meals and breaks on user meeting days.

- Tea/coffee is available at all breaks and after lunch in the Breakout Area outside the Central Hall and workshop rooms.
- Dinner includes tea/coffee. Wine and other drinks are available to purchase with dinner. Wine is included with the closing dinner.
- All bedrooms have tea/coffee making facilities, and room service is available.

Breakfast is served **7:00** to **10:00** on weekdays and **7:30** to **10:30** on weekends. The hotel's Vista Bar Café is on the ground floor and is open from **10:00** each day. Don't forget to settle your bar bill when you check out!

Hotel Facilities

All rooms have an en-suite bathroom, tea/coffee making facilities, TV, air conditioning, an iron/ironing board, a safe, a desk and a hairdryer. Overnight accommodation includes breakfast. The small gym on the mezzanine level has 2 treadmills, 1 cross trainer, 1 bike, 1 cable machine plus a bench and weights. It is always open. The hotel reception can supply information about local restaurants/shops, activities and things to do in the area.

Smoking Policy

The View Hotel operates a strict non-smoking policy, and smoking is not allowed in the main building or the bedrooms (including on the balcony). There are allocated smoking areas outside the hotel and on the mezzanine level balcony.