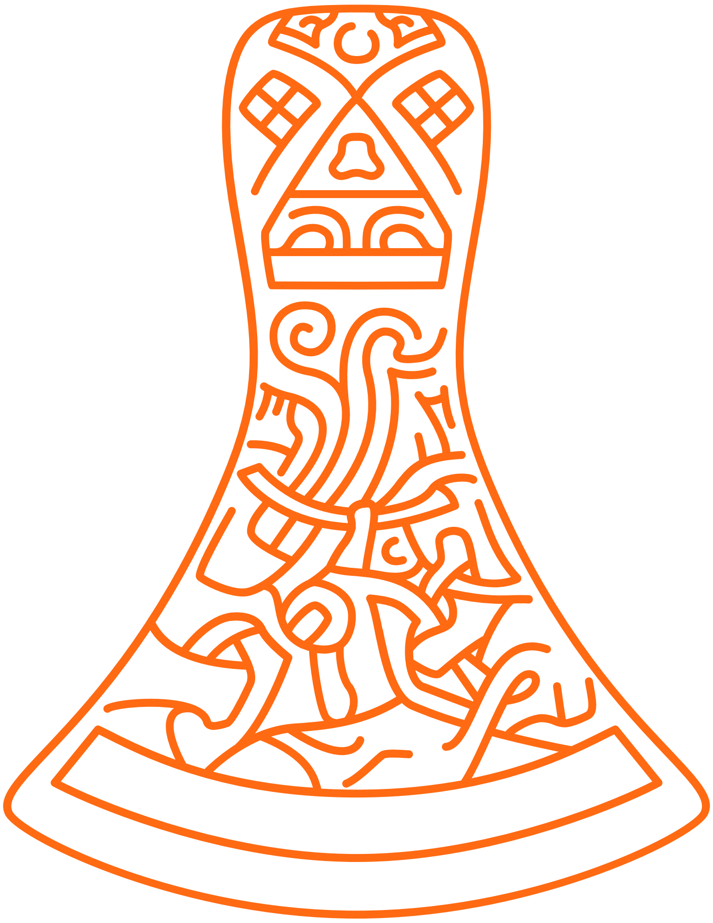
DYALOG

Glasgow 2024

Programme for Dyalog '24



Sunday 15 September – Thursday 19 September 2024

Table of Contents

A Message from Our Managing Director 2
Team Dyalog at Dyalog '24 2
About Dyalog '24 3
Evening Events 3
Schedule: Sunday 15 September 4
Schedule: Monday 16 September 6
Schedule: Tuesday 17 September 8
Schedule: Wednesday 18 September 10
Schedule: Thursday 19 October 12
Abstracts: Workshops 14
Abstracts: Dyalog Presentations 20
Abstracts: User Presentations 25
Practical Information 31

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 our annual user meeting and to Scotland!

These user meetings are very special to all of us at Dyalog Ltd, and we always look forward to these five days of training, presentations and direct encounters with our users, where we can tell you what we have done, what we have in the pipeline, and, not least, catch up on what you have been up to since we last met!

This year is very special to me, as it is my first at the helm. Those of you who are seasoned user meeting participants might have noticed that we have changed the format a bit this year; we have added an extra workshop slot and the banquet is now a closing dinner, but you will have to do without the Viking challenge.

We hope that these changes have improved the experience. Please give us feedback, both positive and negative, so we can make the user meetings even more special in the future.

I wish you a good user meeting, and hope you will enjoy yourself and that, most importantly, you will leave Scotland with a renewed network and that one good idea, insight, or contact, that will mean a difference to you and your organisation in the future!

Stine Kromberg

Team Dyalog at Dyalog '24

This year, Dyalog Ltd is represented by 24 members of the current team. Meet Team Dyalog at http://www.dyalog.com/meet-team-dyalog.htm.

The Dyalog forums (http://forums.dyalog.com/) are regularly visited by Dyalog Ltd staff as well as other stalwarts of the Dyalog community and are a great place to ask questions and share your expertise.

Useful Dyalog e-mail addresses:

- sales@dyalog.com – any non-technical questions related to Dyalog, for example, pricing information or licence purchase
- support@dyalog.com – any technical questions concerning Dyalog

About Dyalog '24

Materials from the presentations and workshop will be available on the Dyalog '24 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

Monday is quiz night! Join Jada and Karta for *Whyalog '24*, 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 '24 is taking place in the Restaurant after dinner (during dessert).

On **Tuesday** 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 Lounge Bar after dinner:

- IDE, Ride, VSCode, and so on – John Daintree
- Language Ideas – Adám Brudzewsky
- Teaching APL – Rich Park
- Future of the Dyalog user meeting, public outreach, and APL user groups – Stine Kromberg

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

Schedule: Sunday 15 September

08:30 – 09:30	Registration (Dyalog desk in the *Central Plaza*)
09:30 – 13:00 tea and coffee available throughout	Workshops **(SA1) Problem Solving with APL** *Room: Inspiration* Adám Brudzewsky (asst. Rich Park & Stefan Kruger) **(SA2) Performance Basics** *Room: Arcoona* Josh David and Karta Kooner (asst. Peter Mikkelsen) **(SA3) Web Services** *Room: Agememnon* Brian Becker (asst. Morten Kromberg and Michael Baas)
13:00 – 14:00	**LUNCH** *Restaurant*
14:00 – 17:30 tea and coffee available throughout	Workshops **(SP1) Tacit Programming in Dyalog** *Room: Inspiration* Adám Brudzewsky (asst. Rich Park & Stefan Kruger) **(SP2) Selected Primitives** *Room: Arcoona* Brian Becker (asst. Peter Mikkelsen & Silas Poulson) **(SP3) Migrating Your Win32 GUI to Linux, macOS, and the Cloud** *Room: Agamemnon* Morten Kromberg (asst. Karl Holt)
17:30 – 19:00	Free time
19:00	**DINNER** *Restaurant*

For workshop abstracts, see pages 14-19.

NOTES:

Schedule: Monday 16 September

All presentations take place in the *Auditorium*

08:30 – 09:00	Registration (Dyalog desk in the *Central Plaza*)
09:00 – 09:05	ANNOUNCEMENTS
09:05 – 09:35	**(D01) Welcome to Dyalog '24** Stine Kromberg, Managing Director (CEO)
09:35 – 10:10	**(D02) The Road Ahead** Morten Kromberg, Technical Director (CTO)
10:10 – 10:40	**(D03) Array Notation: A Journey of Discovery** John Daintree
10:40 – 11:10	TEA/COFFEE
11:10 – 11:40	**(D04) Setting and Getting Variable Values Mk II** Adám Brudzewsky
11:40 – 12:25	**(D05) WC Plugins** John Daintree
12:25 – 13:25	**LUNCH** *Restaurant*
13:25 – 13:55	**(U01) Dyalog APL in the Biggest Data Centres at the Heart of the Investment Management Industry** Oliver Lanz (SimCorp)
13:55 – 14:25	**(D06) Static Analysis of APL in APL** Brandon Wilson
14:25 – 14:55	**(U02) Telemetry and Protobuf** Gilgamesh Athoraya and Sandra Persson (Tiamatica AB)
14:55 – 15:15	TEA/COFFEE
15:15 – 15:45	**(U03) raylib-apl – A Simple Cross-Platform Library for Graphics, Sound, Interactivity, and more** Brian Ellingsgaard (*independent*)

15:45 – 16:30	**(D07) Everywhere WC** Morten Kromberg
16:30 – 19:00	Free time
19:00	**DINNER** *Restaurant* Followed by **Whyalog '24** (quiz)

For Dyalog presentation abstracts see pages 20-24.
For user presentation abstracts see pages 25-30.

NOTES:

Schedule: Tuesday 17 September

All presentations take place in the *Auditorium*

09:00 – 09:05	ANNOUNCEMENTS
09:05 – 09:35	**(U04) Taming Regression using APL** Stephen Mansour (Misericordia University)
09:35 – 10:05	**(U05) Dyalog for Data Science** Jesús Galán López, Ghent University
10:05 – 10:35	**(U06) Climbing Trees and Catching Bugs** Asher Harvey-Smith (University of Warwick)
10:35 – 10:55	TEA/COFFEE
10:55 – 11:25	**(U07) APL MOOC: An APL Course for University Students** Sergey Ichtchenko (University of Helsinki)
11:25 – 11:45	**(D08) The APL Forge – Introduction and Prize Ceremony** Stine Kromberg
11:45 – 12:25	**(U08) Planely Winning the APL Forge** Holden Hoover (independent)
12:25 – 13:25	**LUNCH** *Restaurant*
13:25 – 13:55	**(U09) Migrating to Dyalog from APL+Win: Workspace Transformation** Markos Mitsos, ERGO
13:55 – 14:25	**(U10) Deployment of the Dyalog Interpreter as an OLE Server for Interactive Use (Microsoft Windows Only)** Jürgen Wiedemann (Dittrich und Partner Consulting)
14:25 – 14:55	**(D09) Migrating APL+Win Applications** Morten Kromberg and Karl Holt
14:55 – 15:15	TEA/COFFEE

15:15 – 15:45	**(D10) Initialising and Starting from Text Files** Adám Brudzewsky
15:45 – 16:15	**(D11) New Tatin Packages** Brian Becker
16:15 – 16:45	**(D12) New Function for Shell Calls** Peter Mikkelsen
16:45 – 19:00	Free time
19:00	**DINNER** *Restaurant*
after dinner	**"Birds of a Feather" discussion groups:** • **IDE, Ride, VSCode, and so on** – John Daintree • **Language Ideas** – Adám Brudzewsky • **Teaching APL** – Rich Park • **Future of the Dyalog user meeting, public outreach, and APL user groups** – Stine Kromberg *Lounge Bar*

For Dyalog presentation abstracts see pages 20-24.
For user presentation abstracts see pages 25-30.

NOTES:

Schedule: Wednesday 18 September

All presentations take place in the *Auditorium*

09:00 – 09:05	ANNOUNCEMENTS
09:05 – 09:35	**(D13) Co-dfns Roadmap and Updates** Aaron Hsu
09:35 – 10:05	**(D14) Data Parallel Proof Verification in APL** Brandon Wilson
10:05 – 10:35	**(U11) Developing in Dyalog with Modern Tools** Kai Jaeger (independent)
10:35 – 10:55	TEA/COFFEE
10:55 – 11:25	**(D15) ullu – A Test Framework for Dyalog APL** Aarush Bhat
11:25 – 11:55	**(U12) Noodle, A Cross-Platform, Object-Oriented Framework for Developing Single-User Desktop Dyalog GUI Applications** Pete Donnelly (independent)
11:55 – 12:25	**(D16) Interpreter Limits** John Daintree
12:25 – 13:25	**LUNCH** *Restaurant*
13:25 – 13:55	**(U13) PxEdit to GitHub – Refurbishing a Seasoned APL Application to Open Source** Veli-Matti Jantunen (Statistics Finland)
13:55 – 14:25	**(U14) Some APL Pioneers I Knew** Charles Brenner (DNA-View)
14:25 – 14:55	**(U15) Key Technical Decisions During the Development of Dyalog APL** Geoff Streeter (independent)
14:55 – 15:15	TEA/COFFEE

15:15 – 15:55	**(U16) Let's Put the Future Behind Us (Panel Discussion)** Host: Stephen Taylor (independent) Panellists: • Gitte Christensen (independent) • Pete Donnelly (independent) • Geoff Streeter (independent)
15:55 – 16:35	**(U17) The New Breed Plugs In (Panel Discussion)** Host: Stephen Taylor (independent) Panellists: • Gilgamesh Athoraya (Tiamatica AB) • Martina Crippa • Josh David • Sandra Persson (Tiamatica AB)
16:35 – 16:45	**(D17) Closing Session** Stine Kromberg
16:45 – 19:00	Free time
19:00	**CLOSING DINNER** *Restaurant*

For Dyalog presentation abstracts see pages 20-24.
For user presentation abstracts see pages see pages 25-30.

NOTES:

Schedule: Thursday 19 October

09:30 – 13:00 tea and coffee available throughout	Workshops **(TA1) Tacit Programming in Dyalog** *Room: Inspiration* Rich Park (asst. Stefan Kruger) **(TA2) Selected Primitives** *Room: Arcoona* Brian Becker (asst. Peter Mikkelsen & Silas Poulson) **(TA3) Namespaces in Dyalog** *Room: Agememnon* Adám Brudzewsky (asst. John Daintree)
13:00 – 14:00	**LUNCH** *Restaurant*
14:00 – 17:30 tea and coffee available throughout	Workshops **(TP1) Problem Solving with APL** *Room: Inspiration* Rich Park (asst. Stefan Kruger) **(TP2) APL Functions to Import, Export and Process Data in Files** *Room: Arcoona* Richard Smith (asst. Peter Mikkelsen) **(TP3) Link and the Basics of APL Source in Text Files** *Room: Agamemnon* Morten Kromberg (asst. Adám Brudzewsky)

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://www.dyalog.com/user-meetings/dyalog24/workshops/prerequisites.htm. Please ensure you meet these pre-requisites before attending a workshop.

SA1: Problem Solving with APL

Sunday 09:30 – 13:00
Adám Brudzewsky (asst. Rich Park and Stefan Kruger)

NOTE: This workshop will be repeated as workshop **TP1**.

APL has a wide range of techniques that can be used to solve a given problem, and its terseness and versatility makes it feasible to try different approaches 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

SA2: Performance Basics

Sunday 09:30 – 13:00
Josh David and Karta Kooner (asst. Peter Mikkelsen)

There are a few simple heuristics to consider when it comes to performance in Dyalog. This workshop is targeted at novice APLers, and focuses on the fundamental aspects of the array model and the interpreter that will have the most impact on performance when writing code.

You will learn how to profile functions to find especially slow parts of code, as well as how to write the same computations in a more efficient manner – this includes using flat arrays, specially optimised "spellings" (referred to as idioms), and reducing the number of computations overall where possible.

Keywords: optimisation, performance, profiling, array-oriented programming

SA3: Web Services

Sunday 09:30 – 13:00
Brian Becker (asst. Morten Kromberg and Michael Baas)

NOTE: This workshop is a re-run of workshop SP3 from Dyalog '23, but updated to cover some of the most recently-added features of both HttpCommand and Jarvis as well as using the OpenAI API with HttpCommand

A *web service* is a software system designed to enable communication and interoperability between different applications or systems over the internet. It provides a standardised way for different software applications to exchange data and perform various functions. Web services are commonly used for a wide range of purposes, such as integrating different systems, sharing data between applications, automating business processes, and building distributed applications. Web services provide a scalable and platform-independent way of enabling communication and data exchange between disparate software systems. These functionalities can be accessed and utilised by other software applications, irrespective of the programming languages or platforms they are built on.

In this hands-on workshop, you'll learn how to:

- consume web services using *HttpCommand*.
  HttpCommand is a utility used to issue HTTP requests and receive responses. It can interact with practically any web service. We will:
  - cover the most commonly used features of HttpCommand.
  - learn how to develop and debug web service requests.
  - access a variety of web services including web services that require authentication.
- provide web services using the *Jarvis* web service framework.
  Jarvis makes it easy to make many Dyalog applications available over the internet. We will:
  - develop a simple web service in 5 minutes.
  - cover the essential features of Jarvis.
  - examine and experiment with a sample web service.
  - demonstrate how to access our web service from a browser and using HttpCommand.

Keywords: web services, HttpCommand, Jarvis, OpenAI API

SP1: Tacit Programming in Dyalog

Sunday 14:00 – 17:30
Adám Brudzewsky (asst. Rich Park and Stefan Kruger)

NOTE: This workshop will be repeated as workshop **TA1**.

In tradfns, function arguments must be named in the header. In dfns, they are named for us as ⍺ and ⍵. With tacit programming, we express a function directly as a composition of other functions, without reference to the arguments – the arguments are handled implicitly (tacitly!) according to syntactic rules. This can result in terse and memorable function expressions that can be drawn upon when needed, just like other idioms.

In this hands-on workshop, you will learn how to read, write, and use function trains and compositional operators.

Keywords: function composition, function trains, tacit, operators

SP2: Selected Primitives

Sunday 14:00 – 17:30
Brian Becker (asst. Peter Mikkelsen and Silas Poulson)

NOTE: This workshop will be repeated as workshop **TA2**.

Dyalog has been adding primitives to APL at a steady, but careful, pace. If you would like to get a better grasp of the newest functionality, then this workshop is for you!

In this hands-on workshop you will explore functions like *tally* (monadic ≢), *where* (monadic ⍸), *interval index* (dyadic ⍸), *unique mask* (monadic ≠), and *left*/*right*/*same* (respectively, dyadic ⊣, dyadic ⊢, and monadic versions of both of these), in addition to operators like *at* (@), *constant* (⍨), and *key* (⌸). There will be lots of exercises to aid understanding.

Keywords: language enhancements, exercises

SP3: Migrating Your Win32 GUI to Linux, macOS, and the Cloud

Sunday 14:00 – 17:30
Morten Kromberg (asst. Karl Holt)

Users, IT departments, and auditors are pushing APL developers to move existing Win32 applications to the cloud, to simplify deployment, improve security, and increase accessibility.

In this workshop, you will discover how existing Microsoft Windows desktop application User Interfaces based on ⎕WC objects can be converted to web applications with a minimum of effort and with the added bonus that it becomes easier to update their appearance.

Keywords: Microsoft Windows, WinForms, Graphical User Interface, GUI

TA1: Tacit Programming in Dyalog

Thursday 09:30 – 13:00
Rich Park (asst. Stefan Kruger)

NOTE: This workshop is a repeat of workshop **SP1**.

In tradfns, function arguments must be named in the header. In dfns, they are named for us as α and ω. With tacit programming, we express a function directly as a composition of other functions, without reference to the arguments – the arguments are handled implicitly (tacitly!) according to syntactic rules. This can result in terse and memorable function expressions that can be drawn upon when needed, just like other idioms.

In this hands-on workshop, you will learn how to read, write, and use function trains and compositional operators.

Keywords: function composition, function trains, tacit, operators

TA2: Selected Primitives

Thursday 09:30 – 13:00
Brian Becker (asst. Peter Mikkelsen and Silas Poulson)

NOTE: This workshop is a repeat of workshop **SP2**.

Dyalog has been adding primitives to APL at a steady, but careful, pace. If you would like to get a better grasp of the newest functionality, then this workshop is for you!

In this hands-on workshop you will explore functions like *tally* (monadic ≢), *where* (monadic ⍸), *interval index* (dyadic ⍸), *unique mask* (monadic ≠), and *left*/*right*/*same* (respectively, dyadic ⊣, dyadic ⊢, and monadic versions of both of these), in addition to operators like *at* (@), *constant* (⍨), and *key* (⌸). There will be lots of exercises to aid understanding.

Keywords: language enhancements, exercises

TA3: Namespaces in Dyalog

Thursday 09:30 – 13:00
Adám Brudzewsky (asst. John Daintree)

In Dyalog, namespaces are unusual scalar values that are often under-utilised by the APLer. They are often treated as only a convenient way to organise a workspace, but they can also be used as containers for objects and data. Even classes, whether implemented in APL, external (such as ⎕WC GUI objects), or imported from Microsoft .NET, reside in namespaces.

This workshop demonstrates the value of namespaces both as structured containers to organise code and as self-contained dictionaries of name-value pairs.

Keywords: namespaces, code organisation, dictionaries, object orientation

TP1: Problem Solving with APL

Thursday 13:00 – 17:50
Rich Park (asst. Stefan Kruger)

NOTE: This workshop is a repeat of workshop **SA1**.

APL has a wide range of techniques that can be used to solve a given problem, and its terseness and versatility makes it feasible to try different approaches 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

TP2: APL Functions to Import, Export and Process Data in Files

Thursday 13:00 – 17:50
Richard Smith (asst. Peter Mikkelsen)

NOTE: This workshop is a re-run of workshop TP3 from Dyalog '23.

The interpreter provides several built-in functions to import, export, and process data in files, and to manage the host file system. These are invaluable tools in their own right, and can be the building blocks for powerful APL applications that are largely independent of the host system on which they run. This workshop examines these functions in depth, starting with a simple introduction and then showing their Dyalog v19.0 enhancements.

Keywords: import data, export data, built-in portable file functions

TP3: Link and the Basics of APL Source in Text Files

Thursday 13:00 – 17:50
Morten Kromberg (asst. Adám Brudzewsky)

The Dyalog community is transitioning from binary formats to Unicode text files as the preferred mechanism for storing the source code of APL applications. Dyalog includes a tool known as *Link*, which maintains the connection between the names in the active workspace and the corresponding source files.

This workshop is aimed both at users who are new to Dyalog and wondering how to start a project, and existing users who would like to transition to using text source. It covers:

- Part 1: Getting started with Link
- Part 2: Moving an existing APL application to Link
- Part 3: Project, Package and Source Code Management tools – Git, Cider, Tatin, and NuGet

Keywords: Link, Cider, Tatin, Git, NuGet

Abstracts: Dyalog Presentations

D01: Welcome to Dyalog '24

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. How many times will he mention Large Language Models?

D03: Array Notation: A Journey of Discovery

John Daintree

John's never been 100% convinced about the proposed Array Notation. He's been on a voyage of discovery...find out where he ended up.

D04: Setting and Getting Variable Values Mk II

Adám Brudzewsky

At Dyalog '23, Adám proposed an approach to assignment and using the value of variables with dynamic names, to replace the current solutions that suffer from poor readability, performance, and safety. The design has now evolved to be both more intuitive and more versatile. Adám presents an update and demonstrates usage.

D05: WC Plugins

John Daintree

John demonstrates extensions (and some simplifications) to the DWA interface that allow the creation of external ⎕WC objects. Dyalog Ltd intends to use this mechanism to re-implement the HTMLRenderer object.

D06: Static Analysis of APL in APL

Brandon Wilson

APL as an interpreted language provides unique challenges for verifying correctness. Recently, Brandon has started work on static analysis of APL code using the Co-dfns interpreter as a back end; the aim of this work is to provide an ergonomic tool to statically specify desired properties of APL codebases. Brandon reports on the initial steps of his proof-of-concept.

D07: Everywhere WC

Morten Kromberg

Many existing applications are written for the Microsoft Windows desktop using the ⎕WC family of system functions (WC stands for Window Create). EWC is a project that aims to allow many – maybe, eventually, most – GUI forms written in this style to be moved to the web, or to non-Windows desktops, using the HTMLRenderer, with minimal changes to application code.

In addition to reproducing capabilities of ⎕WC, the EWC project adds a handful of JavaScript-based objects that do not exist in the Microsoft Windows implementation; these include a ribbon control, an ApexCharts widget, and experimental use of "Flex" to position controls as an alternative to setting position and size.

EWC is an open-source project, and one design goal is to allow users to add additional widgets as required.

D08: The APL Forge – Introduction and Prize Ceremony

Stine Kromberg

What is The APL Forge? How do you compete, and who won the 2024 round? Stine introduces this new competition, which has replaced the International APL Problem Solving Competition (Phase 2) that ran for the last 15 years, culminating in the award ceremony for the winner.

D09: Migrating APL+Win Applications

Morten Kromberg and Karl Holt

Dyalog Ltd has been engaged to port an APL+Win application to Dyalog. An important part of the project is the creation of open-source tools that will be made available free-of-charge for similar migrations. These include tools to:

- move APL+Win source code to text files, and use Git to share the same source code between the APL+Win and Dyalog versions of a code base.
- automatically rewrite code to handle many language differences.

- emulate ⎕WI GUI and other system functions.
- continue to use APL+Win component files during the migration process.

Morten and Karl introduce these tools and explore future development in this area.

D10: Initialising and Starting from Text Files

Adám Brudzewsky

With the release of Dyalog version 19.0, it is now possible to both initialise your session and start your application straight from text files, without any involvement of binary files such as session files and saved workspaces. Adám demonstrates how you can make your favourite development tools available and launch your code — all directly from text sources.

D11: New Tatin Packages

Brian Becker

Tatin is a package manager for Dyalog. Brian introduces some of the new Tatin packages introduced in the last year:

- *NuGet* simplifies loading and incorporating .NET packages in your applications.
- *APLProcess* makes it easy to launch additional APL processes to distribute workload or run in code in an isolated environment.
- *OpenAI* implements an interface from Dyalog to OpenAI's API, which enables the integration of AI capabilities in your applications. OpenAI can also serve as a template for how to use HttpCommand to implement interfaces to other AI frameworks.

(For more information on Tatin, see Morten's Dyalog '22 presentation "The P words…Projects and Packages")

D12: New Function for Shell Calls

Peter Mikkelsen

At Dyalog '23, Peter identified some issues in monadic `⎕SH`/`⎕CMD` and proposed a design for a new system function, `⎕SHELL`, that fixes those issues. Since then, the design has been refined and improved based on feedback. Peter demonstrates the revised `⎕SHELL` system function that will appear in Dyalog version 20.0.

D13: Co-dfns Roadmap and Updates

Aaron Hsu

Aaron presents work that has been done in the Co-dfns compiler to improve integration and support for traditional APL code, as well as a roadmap for these features into the future. He discusses the performance and features of present releases together with caveats, recommendations for using the compiler, and some unique things that Co-dfns can help you to do with APL.

D14: Data Parallel Proof Verification in APL

Brandon Wilson

Formal proof verification of large mathematical databases requires a lot of processing power. Metamath is a simple and flexible computer-processable language that supports rigorously verifying, archiving, and presenting mathematical proofs, and is particularly efficient when processing data in a data-parallel manner. APL provides opportunities to perform verification in a data-parallel manner. Brandon shows how APL can be used to perform data-parallel proof verification of Metamath, something that is not offered by existing Metamath systems.

D15: Ullu – A Test Framework for Dyalog APL – Aarush

Aarush Bhat

ullu is a new testing framework for Dyalog APL that rigorously tests the functionality of each of the primitives. One of the central aims is to get close to 100% source code coverage, to verify the accuracy and reliability of Dyalog APL.

Aarush introduces this new framework, and demonstrates how he has achieved almost 100% code coverage for the primitives that have been incorporated so far.

D16: Interpreter Limits

John Daintree

There are many limits in the Interpreter, some of which you might know about (for example, a maximum rank of 15) and some of which you might not (for example, a depth limit of 256 parentheses in a function). Some limits are easy for us to change, others less so. John discusses the whats, whys, and hows of these limits, how we might change them, and the impacts of doing so.

D17: Closing Session

Stine Kromberg

Dyalog Ltd's Managing Director closes the user meeting.

Abstracts: User Presentations

U01: Dyalog APL in the Biggest Data Centres at the Heart of the Investment Management Industry

Oliver Lanz, SimCorp (Denmark)

Dyalog has a firm place in the heart of SimCorp Dimension, which is the backbone of more than 50% of the top investment management firms worldwide; it manages over €15 trillion combined. Oliver offers some insights into why SimCorp like Dyalog, the position it has within their architecture, and their plans for positioning it better in relation to increased demands for security and cost of ownership. He gives a high level view on the architecture of the SimCorp Dimension product, and discusses SimCorp's position on concepts such as shared code files, the .NET bridge, multi-threading, type checking, code signing, and more.

U02: Telemetry and Protobuf

Gilgamesh Athoraya and Sandra Persson, Tiamatica AB (Sweden)

Monitoring your software's performance and behaviour by collecting telemetry data is nothing new. The open source project OpenTelemetry aims to standardise how this is done. Gilgamesh presents how OpenTelemetry can be used from Dyalog and demonstrates a sample application that emits telemetry data.

Telemetry data can, optionally, be emitted using Protobuf (Protocol Buffers), a mechanism for serialising structured data. Sandra presents an APL implementation of Protobuf and a plugin that generates APL code from schema files.

U03: raylib-apl – A Simple Cross-Platform Library for Graphics, Sound, Interactivity, and more

Brian Ellingsgaard, independent (Faroe Islands)

raylib-apl makes the C library *raylib* available to the APL programmer on all platforms. Like APL, raylib was originally used for teaching, in its case, games programming. However, raylib has expanded to include 2D, 3D, and VR graphics, with custom fonts, animated 3D models, shaders, images, textures, mathematical functions, sound, and more. It has also evolved its own vibrant ecosystem.

Brian describes how you can take advantage of this functionality, and when it is appropriate to do so.

U04: Taming Regression using APL

Stephen Mansour, Misericordia University (U.S.A.)

There is a natural progression in mathematics from arithmetic to algebra to linear regression. Define a linear function $f(x)=mx+b$ and let $y=f(x)$. Using simple arithmetic, one can determine the y-value given x. We can use high-school algebra to find x given y. In linear regression we know both x and y; we want to find the function f. Dyalog allows us to create this function under program control using namespaces and defined operators. Stephen also demonstrates extensions to this concept including multiple and non-linear regression.

U05: Dyalog for Data Science

Jesús Galán López, Ghent University (Spain)

Jesús explains how Dyalog can be used in the field of data science. He starts by solving several typical introductory problems using only the APL primitives, to prove that they offer a powerful toolset for data analysis, before exploring the use of more advanced features in Dyalog for the development of libraries. Finally, he investigates how APL is applied to the solution of real data oriented problems in the field of materials science.

U06: Climbing Trees and Catching Bugs

Asher Harvey-Smith, University of Warwick (U.K.)

A few brave APLers have forged a path forward for working with trees in APL, and Asher wants to bring their findings to the masses. He presents his tutorial for working with trees (represented with parent vectors), describing both its content and the tools he used to combine illustrations, APL code, and prose.

U07: APL MOOC: An APL Course for University Students

Sergey Ichtchenko, University of Helsinki (Finland)

APL is not well known among the younger generation – it's time to make a change! For the past year, Sergey and his team have been developing an online APL MOOC (Massive Open Online Course) for the University of Helsinki. He demonstrates its progress, and shows the impact it can have on how we teach a new generation of APL thinkers.

U08: Planely Winning the APL Forge

Holden Hoover, independent (Canada)

Automatic Dependent Surveillance–Broadcast (ADS-B) is an aviation surveillance technology that allows an aircraft to determine its position using GPS or other methods and continuously broadcast its position and status over a radio signal. These raw signals can be received by ground antennas and transmitted to a central server for parsing and aggregation. Holden demonstrates how he won the 2024 APL Forge with his Radar Ingest System. This APL application is designed to connect multiple antennas and feed real-time raw data from aircraft into the system, where the data is then parsed, stored, and distributed using a web API.

U09: Migrating to Dyalog from APL+Win: Workspace Transformation

Markos Mitsos, ERGO (Germany)

Markos describes the migration of a sheaf of APL+Win workspaces to Dyalog, a project that he has undertaken using the Link framework. "Migration" in this context encompasses restructuring into namespaces, using multiline headers, taking advantage of new primitives, and so on. He also took the opportunity to make further changes to the code, such as improving error handling and enhancing modularisation.

U10: Deployment of the Dyalog Interpreter as an OLE Server for Interactive Use (Microsoft Windows Only)

Jürgen Wiedemann, Dittrich und Partner Consulting (DPC) (Germany)

In connection with a migration project from APL+Win to Dyalog, DPC developed a solution for integrating the Dyalog interpreter in the form of an OLEServer object. The ActiveX server for APL+Win was the model for the solution.

The main requirement for the development of the interface was not only the pure interface to the interpreter for code execution, but also the necessity for access to the development environment in its entirety, with editors and tracer for debugging. In concrete application scenarios, the interface is used for programming and maintaining heterogeneous applications in which different code sources and development environments are used. Jürgen explains the development undertaken by DPC.

U11: Developing in Dyalog with Modern Tools

Kai Jaeger, independent (Germany)

Kai demonstrates how he develops using the latest tools and technologies; he opens a project using *Cider*, investigates a bug report, fixes the issue and reduces the technical debt, produces a new version, runs tests, and publishes the updated version as a Tatin *package*.

This includes using tools such as *Git*, *GitHub*, *Cider*, and *Tatin*, performing a search-and-replace with `]Fire`, using a comparison utility, and leveraging user commands like `]APL2Git` and `]GitHub`, among others.

U12: Noodle, A Cross-Platform, Object-Oriented Framework for Developing Single-User Desktop Dyalog GUI Applications

Pete Donnelly, independent (U.K.)

A few years ago, Pete decided to port *Filos* (Greek for friend), a Greek/English Language system that he had developed for PocketAPL, to mainstream Dyalog. He chose to use the HTMLRenderer to replace the Microsoft Windows GUI, mainly for portability, and this choice forced him to learn HTML, JavaScript, and CSS. He decided to use an object-oriented approach, partly because the HTML DOM (Document Object Model) is object oriented and partly because an OO approach suited his application. At some point, he realised that he had developed the basis of a general framework that could be used to develop any type of Dyalog GUI application, and *Noodle* is the result.

The Microsoft Windows GUI objects (such as Button, Edit, and Grid) are replaced by HTML elements (such as Button, Input, and Table), but Noodle also supports jQuery user interface widgets, and now the commercial jQWidgets library. It provides a built-in layout tool, so users do not need to specify the position and size of GUI elements. Just as the Dyalog Windows

GUI requires no knowledge of the basic Windows API, Noodle users need know nothing about HTML, CSS, or JavaScript. Despite the fact that Noodle uses the HTMLRenderer, it is NOT a web server application.

Pete demonstrates how to use the Noodle Framework to develop the GUI for an application and some of the widgets currently available before discussing future developments.

U13: PxEdit to GitHub – Refurbishing a Seasoned APL Application to Open Source

Veli-Matti Jantunen, Statistics Finland (Finland)

PxEdit started as a temporary application for just one specific purpose, but quickly gained lots of users. It has become an all-purpose tool for creating and managing Px files containing statistical tables, which is crucial for the many organisations around the world who are using PxWeb databases.

Over the years, the application has grown in an uncontrolled fashion, with new features added when needed, quick hacks, and fixes over patches. With an end goal of making PxEdit open source, Veli-Matti gives a short introduction to how the code, infested with almost every APL sin, *might* be rearranged so that another human being could understand it. Possibly.

U14: Some APL Pioneers I Knew

Charles Brenner, DNA-View (U.S.A.)

When Charles graduated from university, Larry Breed invited him to join the APL group at IBM – comprising Ken Iverson *et al.* – in developing an APL interpreter and, to an extent, APL itself. These were inspiring people to work with, as remains the case today.

U15: Key Technical Decisions During the Development of Dyalog APL

Geoff Streeter, independent (U.K.)

Geoff takes a historical walk through some of the key technical decisions made during the development of Dyalog APL. Expect some nostalgia, some repentance, some triumphalism, and some serendipity.

U16: Let's Put the Future Behind Us (Panel Discussion)

Host: Stephen Taylor, independent (U.K.)

Panellists: Gitte Christensen, independent (Denmark), Pete Donnelly, independent (U.K.), Geoff Streeter, independent (U.K.)

Decades ago we expected that by now computers would mostly talk APL, which would have taken over the world. Stephen asks Gitte, Geoff, and Pete how they took care of APL in the real world and kept the dream alive.

U17: The New Breed Plugs In (Panel Discussion)

Host: Stephen Taylor, independent (U.K.)

Panellists: Gilgamesh Athoraya, Tiamatica AB (Sweden), Martina Crippa, Dyalog Ltd (Denmark), Josh David, Dyalog Ltd (U.S.A.), Sandra Persson, Tiamatica AB (Sweden)

For its first two decades, APL was most new users' first encounter with programming – and an exciting plunge into personal computing. These days, most new APLers know several programming languages already. Stephen asks Gilgamesh, Martina, Josh, and Sandra what APL looked like to them at first, and why they stuck with it.

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 will need to check out of the room by **11:00** on the day of departure.

Internet Access

Free WiFi is available in all areas. The WiFi network is *GJCH-Public WiFi* – you will need to accept their terms and conditions to connect. There are PCs with internet access and printers near the reception area that all delegates can use.

Meals and Drinks

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

- The Lunch buffet includes soft drinks and tea/coffee.
- Dinner includes soft drinks and tea/coffee. Wine is available to buy.
- Wine is included with the closing dinner.
- Guest rooms have tea/coffee making facilities.

There is a bar on the ground floor of the hotel that is open every day from 10:00 to 00:00. Please remember to settle your bar bill when you check out!

Hotel Facilities

All rooms have an en-suite bathroom with shower, tea/coffee making facilities, an HD Smart TV, air conditioning, an iron and ironing board, a desk and a hairdryer. Overnight accommodation includes breakfast.

The hotel has a Health Suite on site with a fully equipped gym, steam room, sauna and swimming pool. It is open from 06:30 to 22:00 on weekdays and 08:00 to 20:00 at weekends (last entry 1 hour before closing) and is available to hotel guests. Exercise classes are also included – the timetable for these varies.

The hotel has parking on site but this is shared with the hospital next door so can become congested at times. There are regular trains and buses into central Glasgow – please ask at the hotel reception for details.

Smoking Policy

Smoking is not permitted anywhere in the hotel or its grounds. Guests can smoke at the end of the entrance drive, which is a 2-3 minute walk from the hotel main entrance.