



An APL App End-to-End

Rich Park

APL Evangelist, Dyalog Ltd



Dyalog '24 APLSeeds '24 Dyalog '23 APL Seeds '23 Dyalog '22 Webinar APL Seeds '22 Dyalog '21 APL Seeds '21

Dyalog '20 Dyalog '19 Dyalog '18 Dyalog '17 Dyalog '16 Dyalog '15 Dyalog '14 Dyalog '13 Dyalog '12 Dyalog '11

APL Berlin 2010 Dyalog '09 Dyalog '08



Welcome to Dyalog '24 // Stine Kromberg // Dyalog '24

Dyalog User Meetings

Welcome to Dyalog '24



Opening Address by CEO
Stine Kromberg



Introduction



Welcome to
Dyalog '24
Stine Kromberg



The Road Ahead
Morten Kromberg

Behind the Scenes



Uilu – A Test
Framework for
Dyalog APL
Aarush Bhat

Filter by Date Range

Filter by Presenter

Filter by Event

2008

to

2026

For example: Morten Kromberg

Any



Sort:

Relevance

Per Page:

20



Segmented Scans and Nested Data Parallelism

- Andrzej Filinski

Oct 2012

in Dyalog '12

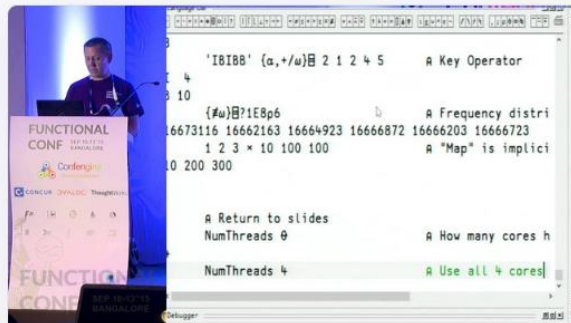


APL Worker Bees – Another Way to Run APL Tasks in Parallel

- Stig Nielsen

Oct 2023

in Dyalog '23



Parallel Programming in Dyalog using Futures and Isolates

- Morten Kromberg

Sept 2015

in Functional Conf 2015



Proof Verification with APL



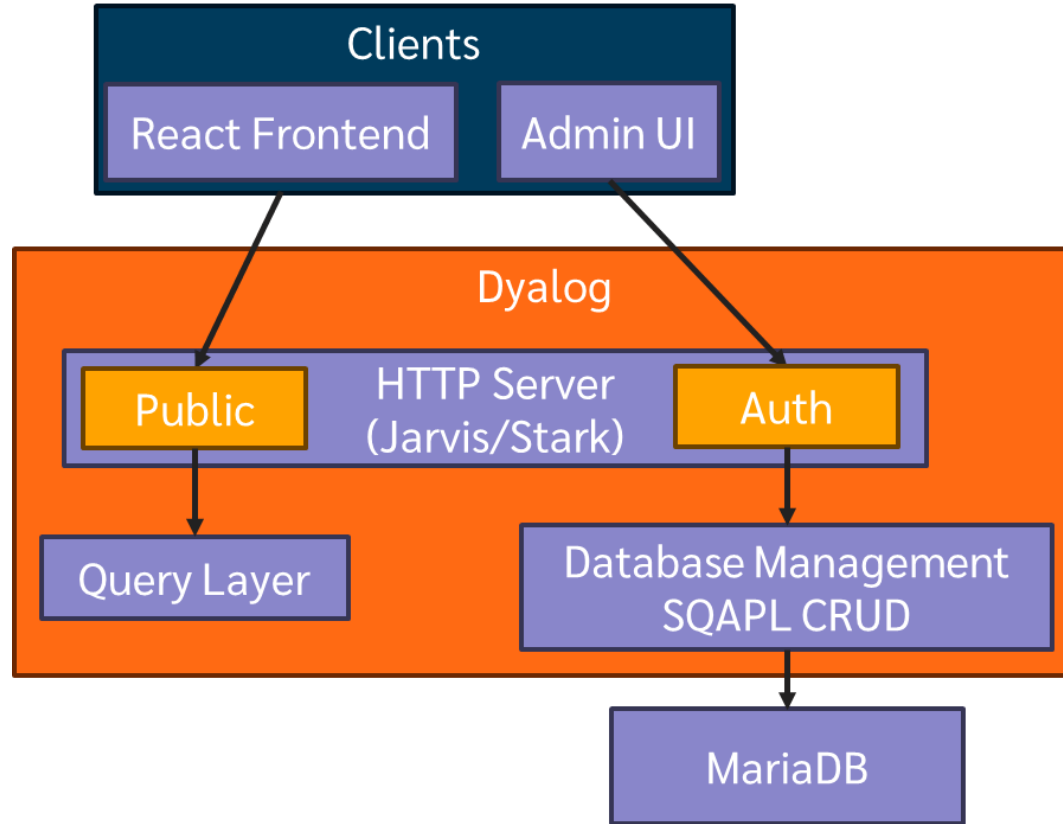
Comments

The "lamp" symbol (A) indicates the beginning of a comment:

Coming Up...

- ◆ Architecture Overview
- ◆ Development Setup
- ◆ Component Details
 - ◆ Database
 - ◆ HTTP Server
 - ◆ Search and Recommendation

Dyalog Content Management System (DCMS)



Development & Deployment

- ◆ Develop in Docker (write once, work anywhere)
- ◆ Mirror Production in Development
- ◆ Jenkins Automation for Deployment
- ◆ Testing
 - ◆ In APL session
 - ◆ In local Docker
 - ◆ In continuous deployment pipeline
- ◆ Reduce "works on my machine" issues

Development & Deployment



Stage View

		Checkout	Build container	Update MariaDB	Install dependencies	Test service	Publish DCMS	Create ENV file	Deploying with Docker Swarm
Average stage times: (full run time: ~2min 55s)		1s	1s	1s	10s	27s	58s	374ms	25s
#239 Apr 20 10:24 No Changes		931ms	1s	1s	10s	27s	56s	368ms	28s
#238 Apr 09 09:10 1 commit		1s	1s	1s	12s	27s	1min 1s	313ms	28s

Tatin & NuGet Packages

```
apl-dependencies.txt
```

```
dyalog-HttpCommand-5.9.3
```

```
dyalog-Jarvis-1.20.5
```

```
dyalog-NuGet-0.2.5
```

```
bkaw-Stark-0.1.10 (not yet public)
```


Tatin & NuGet Packages

```
]tatin.listpackages
Registry: https://tatin.dev          ≠ 70
Group & Name                          # major versions
-----
abrudz-ColSchemes                     1
abrudz-sort                           1
.....
sjt-translate                         1
tiamatica-Mutsu                       1
tiamatica-Notela                      1
```

Tatin & NuGet Packages

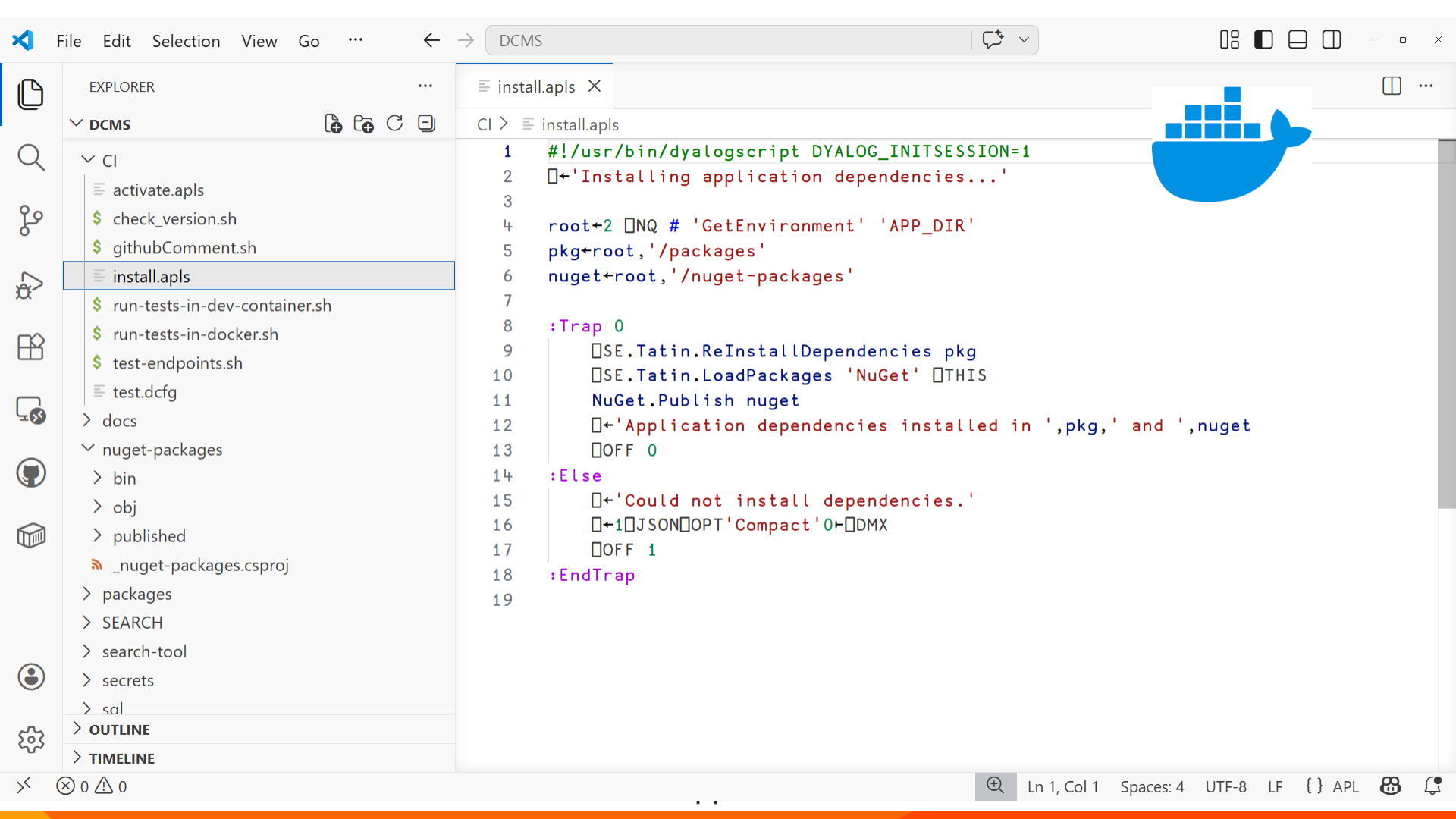
- ◆ NuGet: library of .NET tools
- ◆ Can use from APL with `⎕USING`
- ◆ Porter2Stemmer NuGet package

```
Stem¨(
    |analytic'
    |analytics'
    |analytically'
)
analyt  analyt  analyt
```

Tatin & NuGet Packages

```
NuGet.Setup '/app/nuget-packages '  
NuGet.Add 'Porter2Stemmer '  
DCMS.⊞USING←NuGet.Using '/app/nuget-packages '  
, /app/nuget-packages/published/Porter2Stemmer.dll  
, /app/nuget-packages/published/_nuget-packages.dll
```

```
Stem⊞(  
    analytic'  
    analytics'  
    analytically'  
)  
analyt  analyt  analyt
```



EXPLORER

DCMS

CI

activate.apls

\$ check_version.sh

\$ githubComment.sh

install.apls

\$ run-tests-in-dev-container.sh

\$ run-tests-in-docker.sh

\$ test-endpoints.sh

test.dcfg

> docs

nuget-packages

> bin

> obj

> published

_nuget-packages.csproj

> packages

> SEARCH

> search-tool

> secrets

> sal

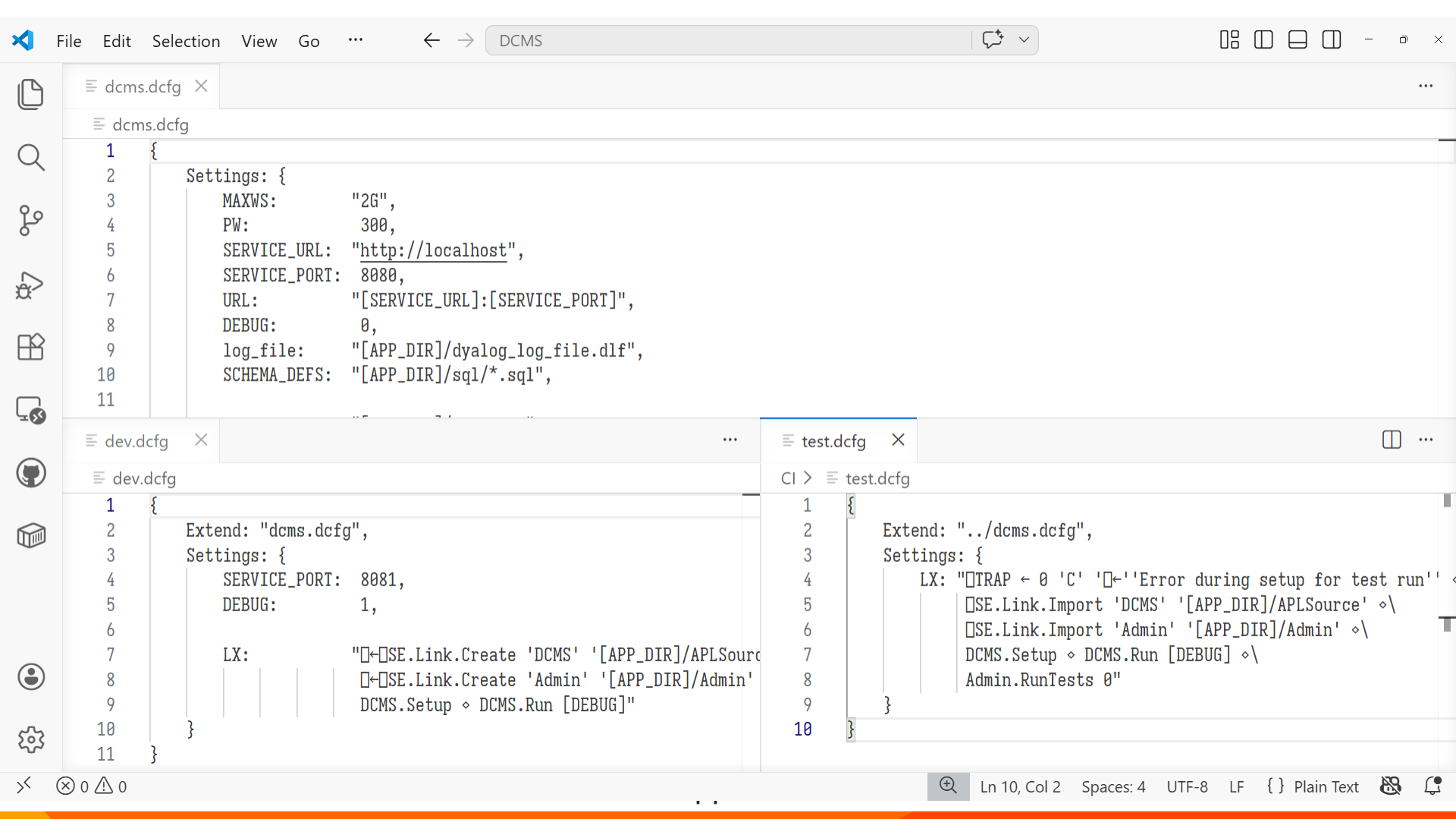
> OUTLINE

> TIMELINE

install.apls X

CI > install.apls

```
1  #!/usr/bin/dyalogscript DYALOG_INITSESSION=1
2  ⎕←'Installing application dependencies...'
3
4  root←2 ⎕NQ # 'GetEnvironment' 'APP_DIR'
5  pkg←root, '/packages'
6  nuget←root, '/nuget-packages'
7
8  :Trap 0
9      ⎕SE.Tatin.ReInstallDependencies pkg
10     ⎕SE.Tatin.LoadPackages 'NuGet' ⎕THIS
11     NuGet.Publish nuget
12     ⎕←'Application dependencies installed in ',pkg, ' and ',nuget
13     ⎕OFF 0
14 :Else
15     ⎕←'Could not install dependencies.'
16     ⎕←1⎕JSON⎕OPT'Compact'0-⎕DMX
17     ⎕OFF 1
18 :EndTrap
19
```



≡ dcms.dcfg ✕

≡ dcms.dcfg

```
1 {
2   Settings: {
3     MAXWS:      "2G",
4     PW:         300,
5     SERVICE_URL: "http://localhost",
6     SERVICE_PORT: 8080,
7     URL:        "[SERVICE_URL]:[SERVICE_PORT]",
8     DEBUG:      0,
9     log_file:    "[APP_DIR]/dyalog_log_file.dlf",
10    SCHEMA_DEFS: "[APP_DIR]/sql/*.sql",
11  }
```

≡ dev.dcfg ✕

≡ dev.dcfg

```
1 {
2   Extend: "dcms.dcfg",
3   Settings: {
4     SERVICE_PORT: 8081,
5     DEBUG:       1,
6
7     LX:          "␣←SE.Link.Create 'DCMS' '[APP_DIR]/APLSour
8                  |␣←SE.Link.Create 'Admin' '[APP_DIR]/Admin'
9                  |␣DCMS.Setup ⚡ DCMS.Run [DEBUG]"
10  }
11 }
```

≡ test.dcfg ✕

CI > ≡ test.dcfg

```
1 {
2   Extend: "../dcms.dcfg",
3   Settings: {
4     LX: "␣TRAP ← 0 'C' '␣←''Error during setup for test run'' ⚡
5         ␣SE.Link.Import 'DCMS' '[APP_DIR]/APLSource' ⚡\
6         ␣SE.Link.Import 'Admin' '[APP_DIR]/Admin' ⚡\
7         DCMS.Setup ⚡ DCMS.Run [DEBUG] ⚡\
8         Admin.RunTests 0"
9   }
10 }
```

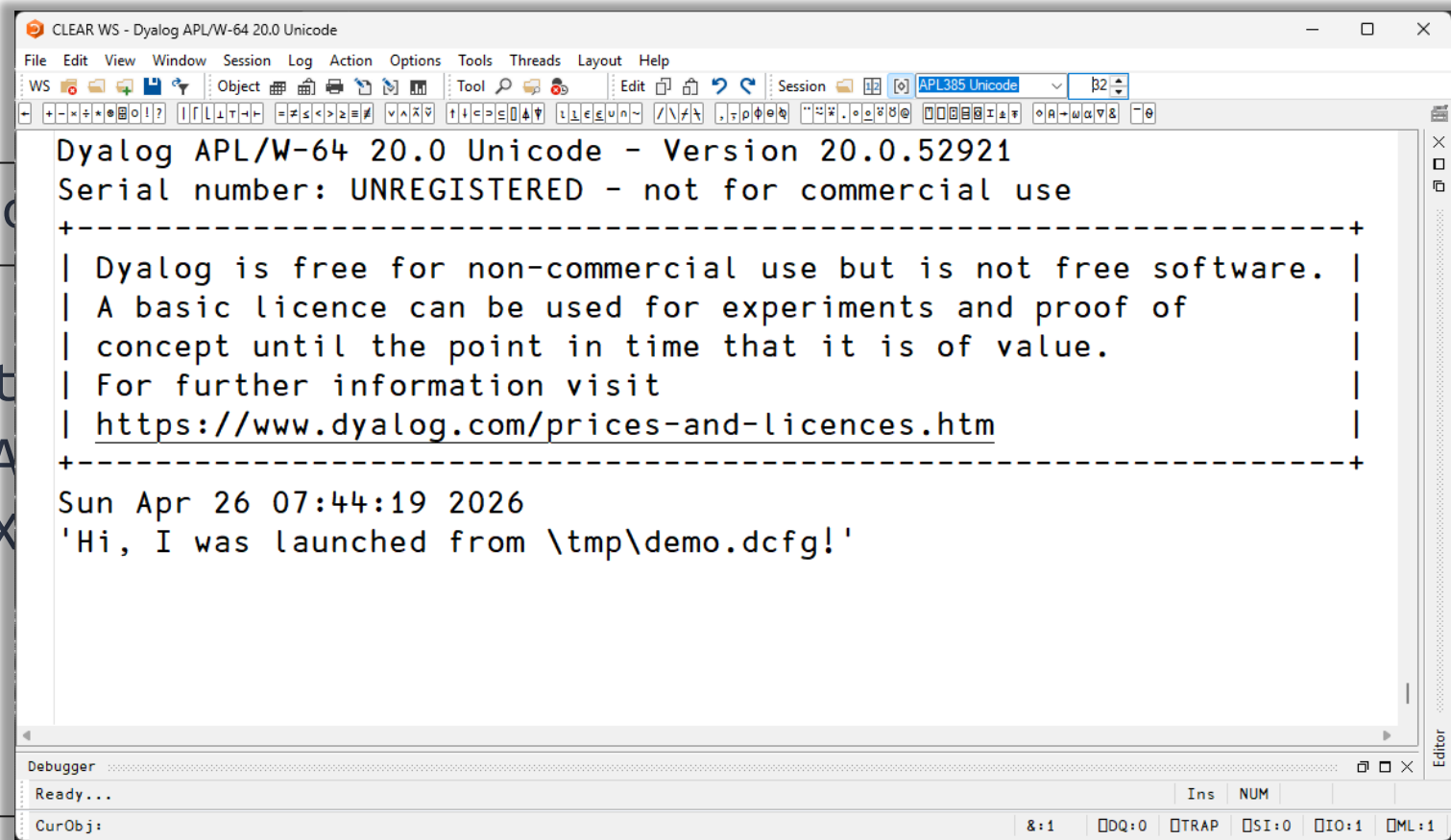
demo.dcfg

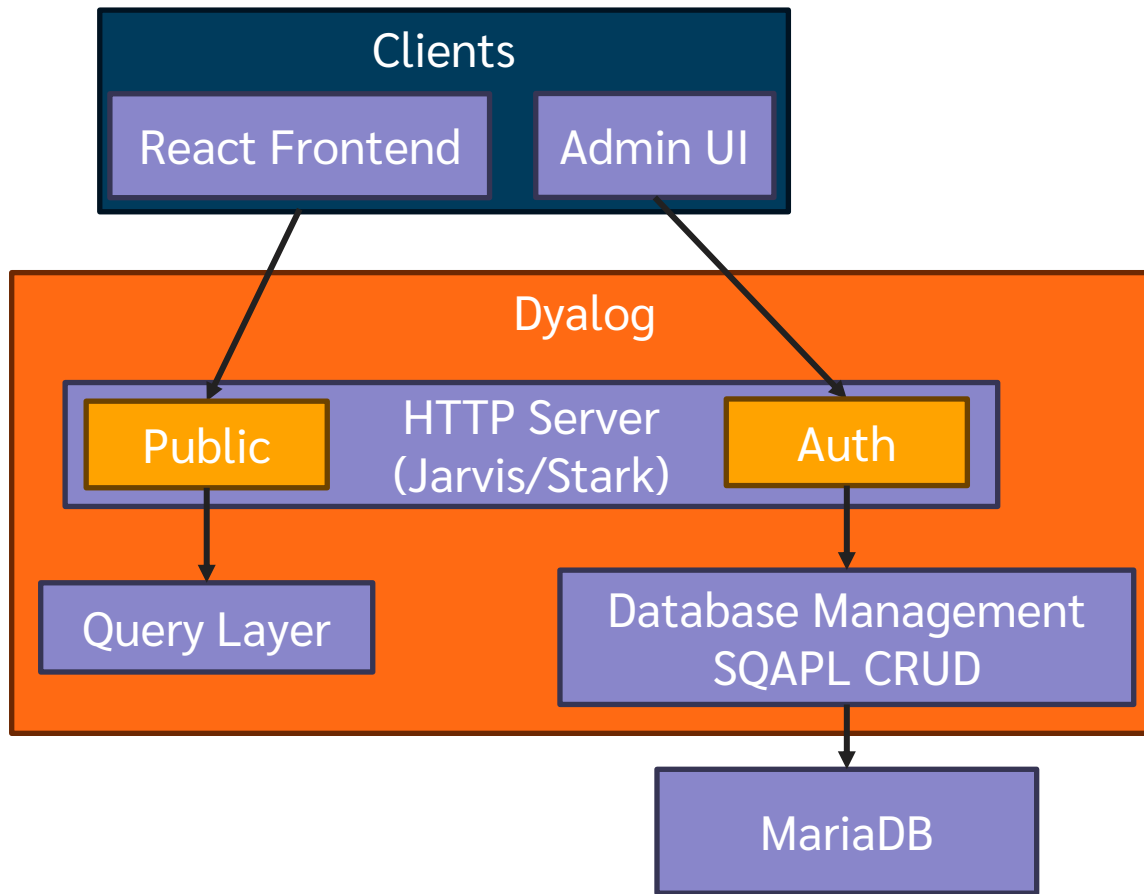
```
{  
  Settings: {  
    MAXWS: "2G",  
    LX: "␣←'Hi, I was launched from [CONFIGFILE]!'"  
  }  
}
```

demo.dcfg

```
{  
  Settings: {  
    MAXWS: "2G",  
    LX: "␣←'Hi, I was launched from [CONFIGFILE]!'"  
  }  
}
```

```
> .\dyalog.exe CONFIGFILE=\tmp\demo.dcfg
```





MariaDB

Filter by Date Range

Filter by Presenter

Filter by Event

2008

to

2026

For example: Morten Kromberg

Any



Sort:

Relevance

Per Page:

20



Segmented Scans and Nested Data Parallelism

- Andrzej Filinski

Oct 2012

in Dyalog '12

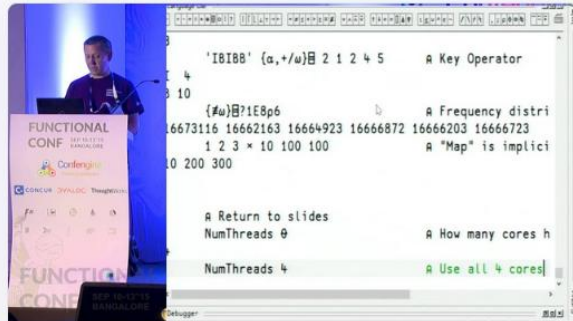


APL Worker Bees – Another Way to Run APL Tasks in Parallel

- Stig Nielsen

Oct 2023

in Dyalog '23



Parallel Programming in Dyalog using Futures and Isolates

- Morten Kromberg

Sept 2015

in Functional Conf 2015



Proof Verification with APL



Comments

The "lamp" symbol (A) indicates the beginning of a comment:

Filter by Date Range

Filter by Presenter

Filter by Event

2008

to

2026

For example: Morten Kromberg

Any



Sort:

Relevance

Per Page:

20



Segmented Scans and Nested Data Parallelism

- Andrzej Filinski

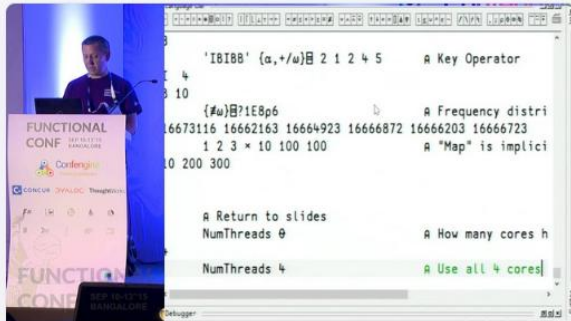
Oct 2012

in **Dyalog '12**

APL Worker Bees – Another Way to Run APL Tasks in Parallel

- Stig Nielsen

Oct 2023

in **Dyalog '23**

Parallel Programming in Dyalog using Futures and Isolates

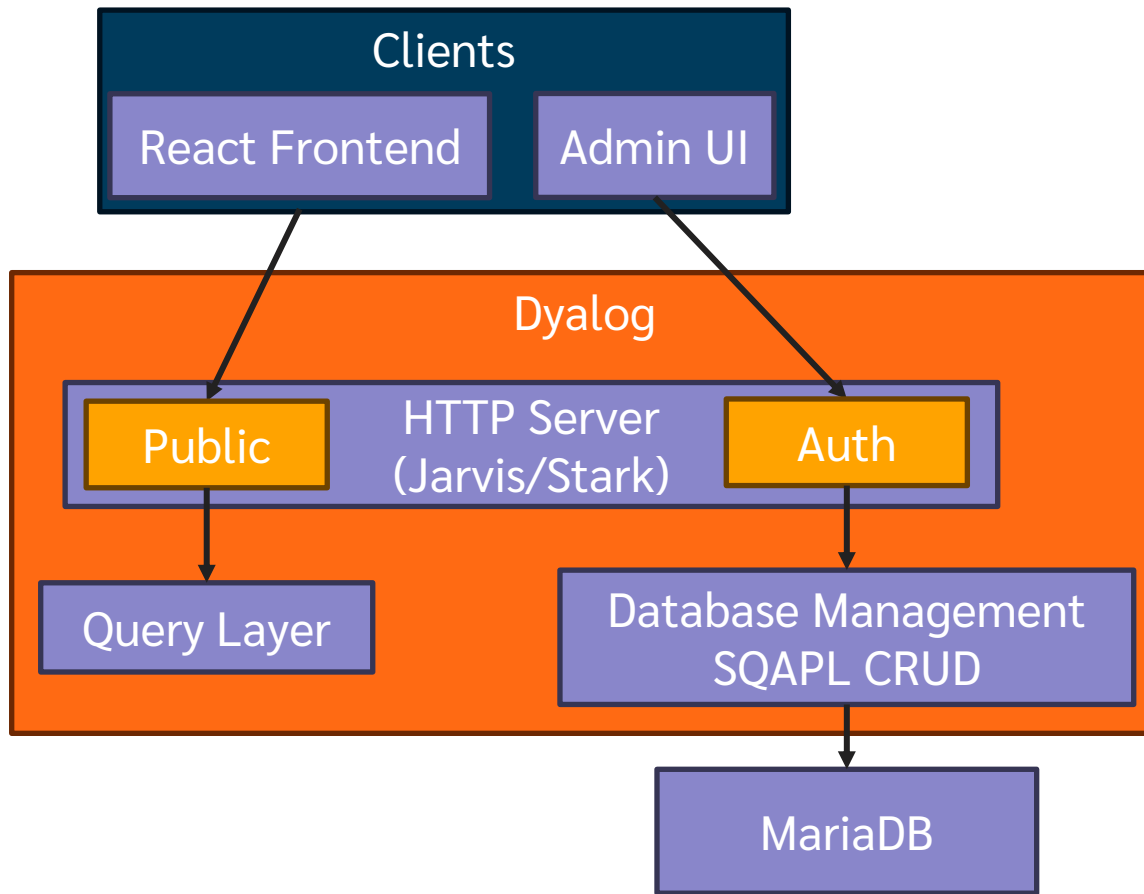
- Morten Kromberg

Sept 2015

in **Functional Conf 2015**

MariaDB

- ❖ Data integrity
 - ❖ Database normalization
 - ❖ Foreign key constraints
- ❖ Share data source with other apps



CRUD

API: Create, Read, Update, Delete

SQL: INSERT, SELECT, UPDATE/SET, DELETE

```
sql←'SELECT col1,col2 FROM table :>type1,type2:'  
SQA.Prepare cursor  
SQA.Exec cursor  
data←2>SQA.Fetch cursor
```

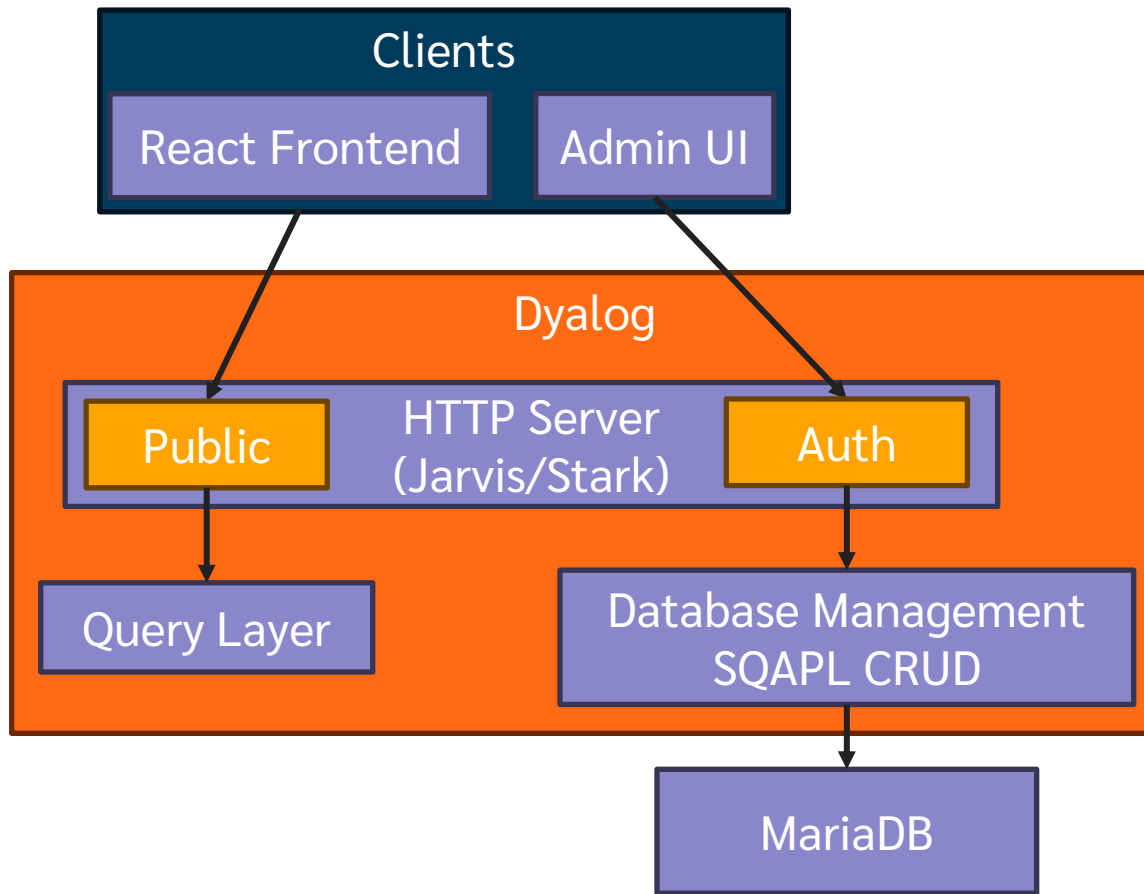
CRUD

API: Create, Read, Update, Delete

SQL: INSERT, SELECT, UPDATE/SET, DELETE

```
SQL.Select(table:table_name ♦ columns:fields)
```


CRUD



HTTP Server

Jarvis REST mode

REST: Principles for resource management

HTTP: POST GET PATCH DELETE

API: Create Read Update Delete

Stark REST router (experimental)

```
'/crud/{table}' router.Get 'CRUD.Read'
```

+ OpenAPI spec generator!

OpenAPI

- ◆ Standardized framework for REST API specification
- ◆ Well supported tooling and interoperation

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

▼ info:

title: "Dyalog Content Management System"
version: "4.0.0"
openapi: "3.0.3"

▼ paths:

▼ /crud/{table}:

▶ get: { operationId: "listRows", summary: "List rows from table", parameters: (3)[...], ... }

▼ post:

operationId: "createRows"
▶ parameters: [{...}]
▶ requestBody: { required: true, content: {...} }
▶ responses: { 201: {...}, 400: {...}, 401: {...}, ... }
summary: "Create rows"
▶ tags: ["Table CRUD"]

▶ /crud/{table}/schema: { get: {...} }
▶ /crud/{table}/{id}: { delete: {...}, get: {...}, patch: {...} }
▶ /events: { get: {...} }
▶ /presenters: { get: {...} }
▶ /teapot: { get: {...} }
▶ /version: { get: {...} }
▶ /videos: { get: {...} }
▶ /videos/{youtube_id}: { get: {...} }
▶ /videos/{youtube_id}/recommendations: { get: {...} }

SQL Tables to JSON Objects

- ◆ APL matrix/table from SQL database
- ◆ REST service provides JSON
- ◆ JSON wrappers!



Wrappers

A wrapper is an enclosed vector of the form:

```
<code> special
```

The nature of the `special` data structure is identified within the wrapper by a leading numeric code. Code 1 is used to identify JSON values such as `null`, `true` and `false`. Codes 2, 3 and 4 are used to identify different forms of datasets.

This wrapper mechanism has been chosen to identify special treatment because a scalar enclosure cannot be represented in JSON/JavaScript.

A wrapper may be specified directly in the right argument to `JSON` and/or as part of the array structure specified by the right argument, as a sub-array or in a namespace. This allows a special array to be processed appropriately as part of a general data structure that is to be rendered in JSON notation.

Wrappers for special JSON values

Wrappers may be used to export JSON special values such as `null`, `true` and `false` using code 1. This mechanism is supplementary to the use of enclosed character vectors. See

On this page

[JSON Import \(X is 0\)](#)

[Import as Data \(Format 'D'\)](#)

[Import as Matrix \(Format 'M'\)](#)

[JSON Export \(X is 1\)](#)

[Raw Text](#)

[Wrappers](#)

[JSON Name Mangling](#)

```
Table2JSON←{
  1 ⍳JSON⊂2,⊂ω
}
```

name	age	is_cool
Tom	42	false
Dick	56	true
Asher	37	true

json←Table2JSON matrix



```
[{
  age: 42,
  is_cool: false,
  name: "Tom",
},
{
  age: 56,
  is_cool: true,
  name: "Dick",
},
{
  age: 37,
  is_cool: true,
  name: "Asher",
}]
```



```
Table2JSON←{  
    1 ⍳JSON⊂2,⊂ω  
}
```

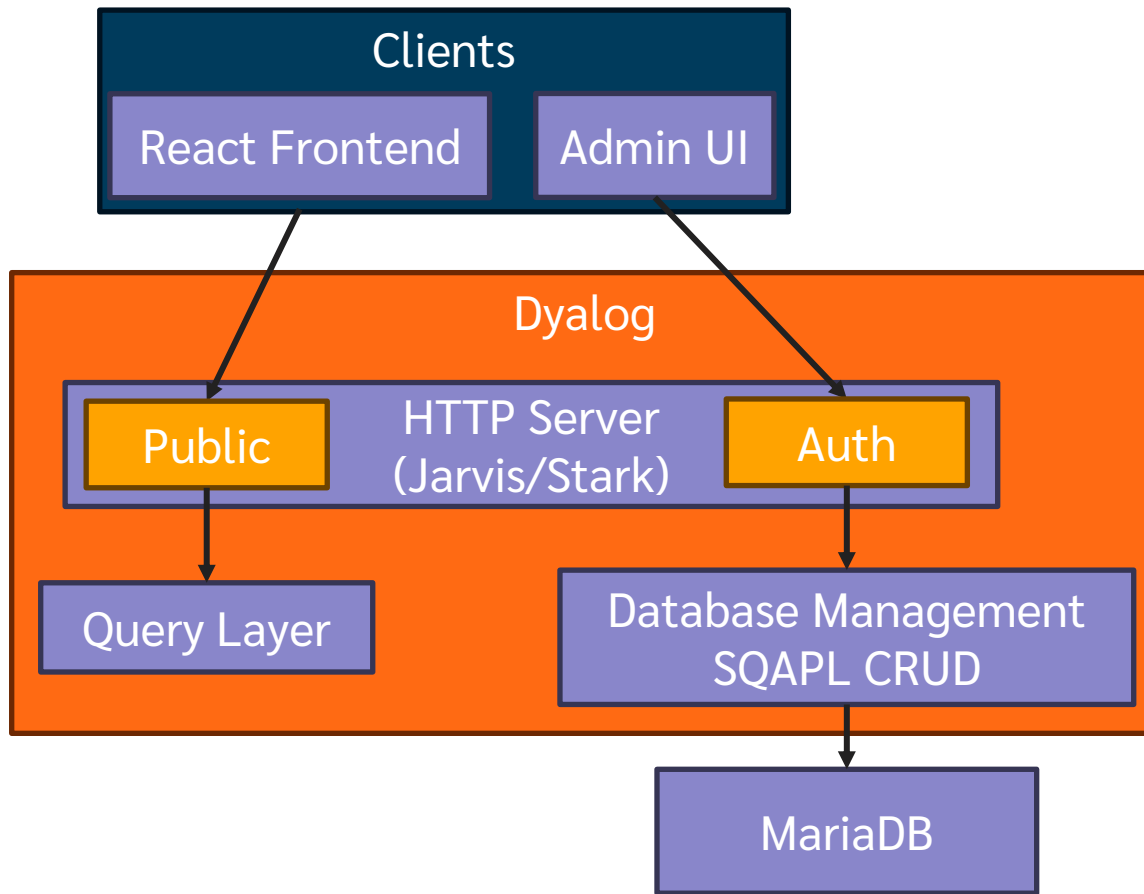
CRUD/Create.aplf

```
our_values←↑req.Payload ⌵VGET column_names
```

```
inserted←SQL.Insert(  
  table:table_name ⋄ columns:column_names  
  values:our_values ⋄ returning:our_cols  
)
```

```
r←Table2JSON our_cols;inserted
```

CRUD



Query Layer

Search

Filter

Recommendations

Search

Term-frequency inverse-document-frequency

Common words count less towards results

$$tf_idf \leftarrow tf + . \div df$$

Search

```
term_count ← ⍋(+/⊖ docs) terms
relevant_vids ← ⍋v/0 < term_count
tf ← term_count[relevant_vids;]
df ← +/tf
tf_idf ← tf +. ÷ df
relevant_vids[⍋tf_idf]
```

Query Layer

Search

Filter

Recommendations

Filter Date Range

```
⊞←ddn←DT2DDN '2026-04-27 09:30:00'  
46138.39583
```

```
(from ≤ ^ ≤ to) ddn
```

Filter Date Range

```
DT2DDN←{1 ⌈DT 2>∘' - : '∘⌈VFI∘⊆ω}
```

```
FormatDateTime←{α(1200⌈)ω}
```

```
⌈←ddn←DT2DDN '2026-04-27 09:30:00'
```

```
46138.39583
```

```
'YYYY-DD-MM hh:mm:ss' FormatDateTime ddn
```

```
2026-27-04 09:30:00
```

Filter Date Range

```
DT2DDN←{'YYYY-MM-DD hh:mm:ss' 1 ⍴DT⊆ω}
```

```
FormatDateTime←⍴DT
```

```
⍴←ddn←DT2DDN '2026-04-27 09:30:00'  
46138.39583
```



```
'YYYY-DD-MM hh:mm:ss' FormatDateTime ddn  
2026-27-04 09:30:00
```

Filter Date Range

```
DT2DDN←{'%ISO%' 1 ⍵DT≤ω}
```

```
FormatDateTime←⍵DT
```

```
⍵←ddn←DT2DDN '2026-04-27T09:30:00'  
46138.39583
```



```
'%ISO%' FormatDateTime ddn  
2026-27-04T09:30:00
```

Query Layer

Search

Filter

Recommendations

Recommendations

Setting and Getting Variable Values

Adám Brudzewsky

November 2023

in [Dyalog '23](#)

Assignment and using the value of a variable are important elements of programming, but, in more complex systems, the name of a variable is sometimes dynamic, and determined separately. Current solutions to these situations suffer from poor readability and performance. Adám proposes an alternative approach in the form of system functions for getting name-value pairs (`⊞NV`), getting a name (`⊞NG`) and setting a name (`⊞NS`). If you recognise the last one, luckily his proposal is a consistent extension of the current `⊞NS` system function.

00:00 Common usage scenarios

01:05 Set variable values is awkward and slow

03:43 Getting the value of a variable is also

06:35 Comparing performance of error guard vs explicit checking

07:15 Getting name value pairs from a namespace

08:03 Name set `⊞NS`

09:14 Name get `⊞NG`

10:13 Name value pairs `⊞NV`

12:00 Basic usage examples

12:24 Set default left argument to `tradfn`

12:46 Merge values from multiple namespaces

13:20 Query data objects with defaults for missing values

Suggested Videos:



Setting and Getting Variable Values Mk II

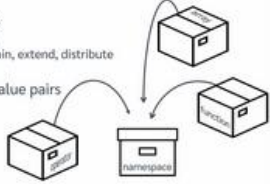
- Adám Brudzewsky

Sept 2024 in Dyalog '24

Namespaces in Dyalog APL

A way to organise code
Encourage modularity
Easier to debug, maintain, extend, distribute

A collection of name-value pairs
Like a dictionary
Named parameters



Namespaces in Dyalog APL

- Rich Park

Jun 2022 in Tutorials



One-Time Pure Mathematician Corrupted by Exposure to APL

- Charles Brenner

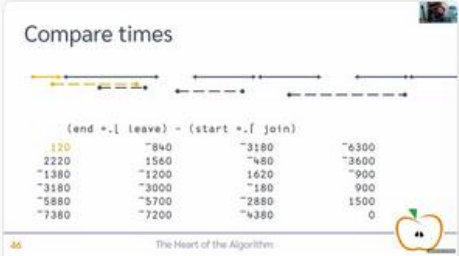
Oct 2022 in Dyalog '22



Creating and managing your own User Commands

- Adám Brudzewsky

May 2018 in Dyalog Webinar



Compare times

$(end \cdot \leftarrow \leftarrow leave) - (start \cdot \leftarrow \leftarrow join)$

120	~840	~3180	~6300
2220	1560	~480	~3600
~1380	~1200	1620	~900
~3180	~3000	~180	900
~5880	~5700	~2880	1500
~7380	~7200	~4380	0



Giving Key a Vocabulary

Adám Brudzewsky

DYALOG
Elsinore 2023

Multi-line documents

- Process line-by-line or entire document
- Line-by-line...
 - Less memory usage
 - Useful semantic
- Entire document...
 - Search across lines
 - Replace operations may eliminate lines



Interval Index

Dyalog `ai` looks up items of `ai` in the intervals defined by the sorted sequence `a`



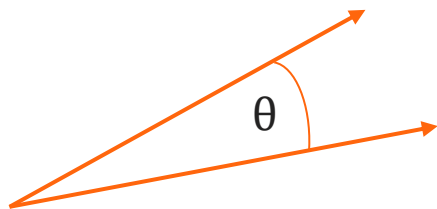
Benefits:

- Classification of data is a very common operation
- With Interval Index, this becomes simple and very fast

DEMO!



Cosine Similarity



$$A \cdot B = |A||B| \cos \theta$$

$$\cos \theta = \frac{A \cdot B}{\sqrt{\sum A^2} \times \sqrt{\sum B^2}}$$

$$(c + . \times \phi c) \div o . \times \sim 0.5 * \sim + / c * 2$$

Cosine Similarity

Isn't that expensive?


Yes

Isolates


Isolates

Core Documentation

These documents describe the details of the language and program construction; they are not specific to an operating system.

  Dyalog APL Language Reference Guide ([summary](#))


  Dyalog Programming Reference Guide ([summary](#))

 .NET Interface Guide ([summary](#)) NOTE: Dyalog Unicode edition only

 Comparison of .NET/.NET Framework Interfaces

 Compiler User Guide ([summary](#))

 **Parallel Language Features** ([summary](#))

 Shared Code Files User Guide ([summary](#)) NOTE: Dyalog Unicode edition only

https://www.dyalog.com/documentation_200.htm

Isolates

```
#.⊂CY'isolate'  
II←#.II ⋄ Values←#.isolate.Values
```

```
vids←CACHE.videos
```

```
data←vids.index_cols⊂~cvids.fields⊂'title' 'description' 'presenter'
```

```
recommendations←videos.ComputeRecommendations II data
```

```
:If 0=⊂NC'vids.recommendations'  
    vids.recommendations←1p~s,0⌈-1+s←≠vids.values  
:EndIf
```

Meanwhile...

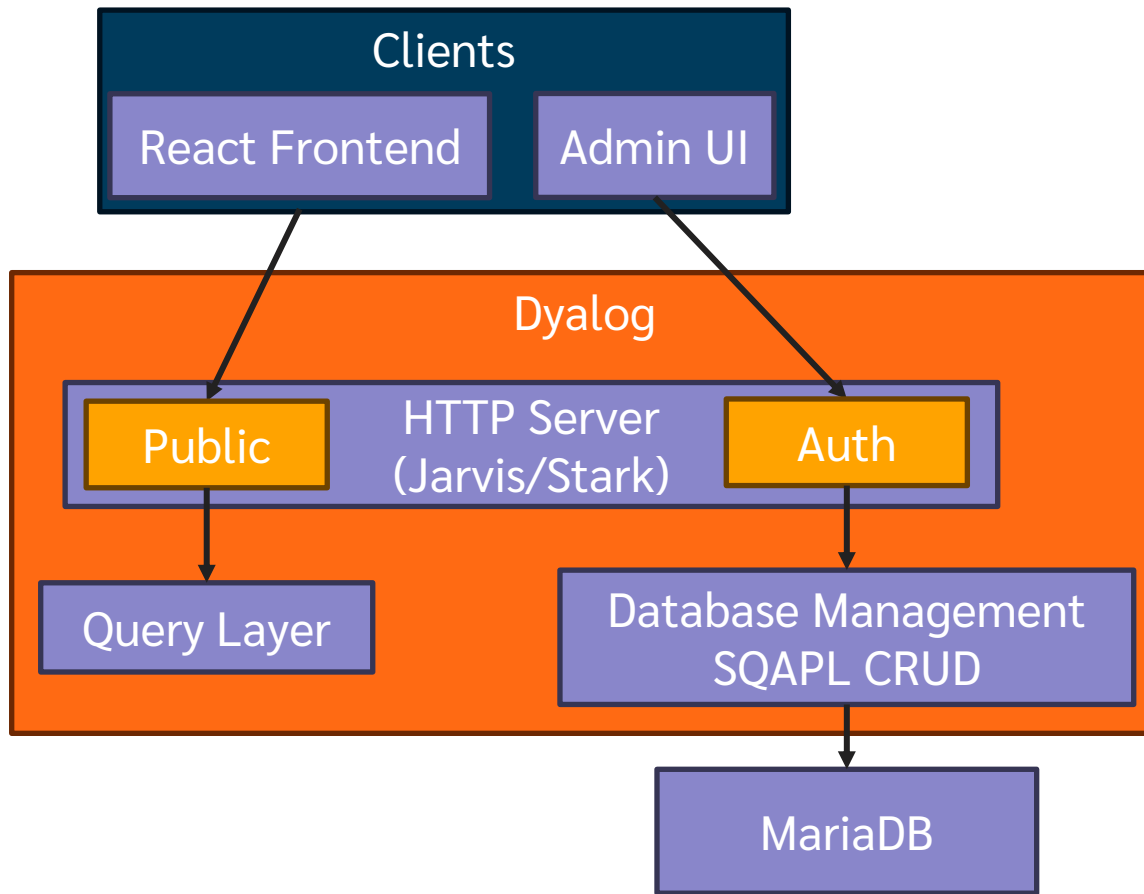
... in the Recommendations endpoint function

```
Values←#.isolate.Values
```

```
⍺ Use existing values unless the ComputeRecommended
```

```
⍺ isolate has finished processing
```

```
vids.recommendations←>vids.recommendations Values'##.recommendations'
```



github.com/Dyalog/DCMS

dyalog.com/tools/tools-and-code-libraries.htm