

DYALOG

2021





Link Version 3.0

Morten Kromberg
CTO, Dyalog Ltd.

What is Link?

According to <https://github.com/dyalog/link>

 README.md 

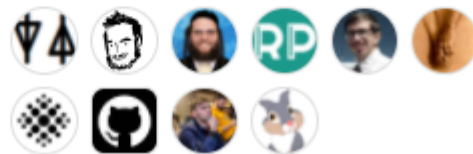
Link

Link enables the use of text files as the primary storage mechanism for APL source code, and is expected to replace the use of traditional binary workspaces and component files for many new APL projects.

Packages

No packages published
[Publish your first package](#)

Contributors 10



All you need to Know about Link...

To declare that your source is in C:\linkdemo

```
]link.create # C:\linkdemo
```

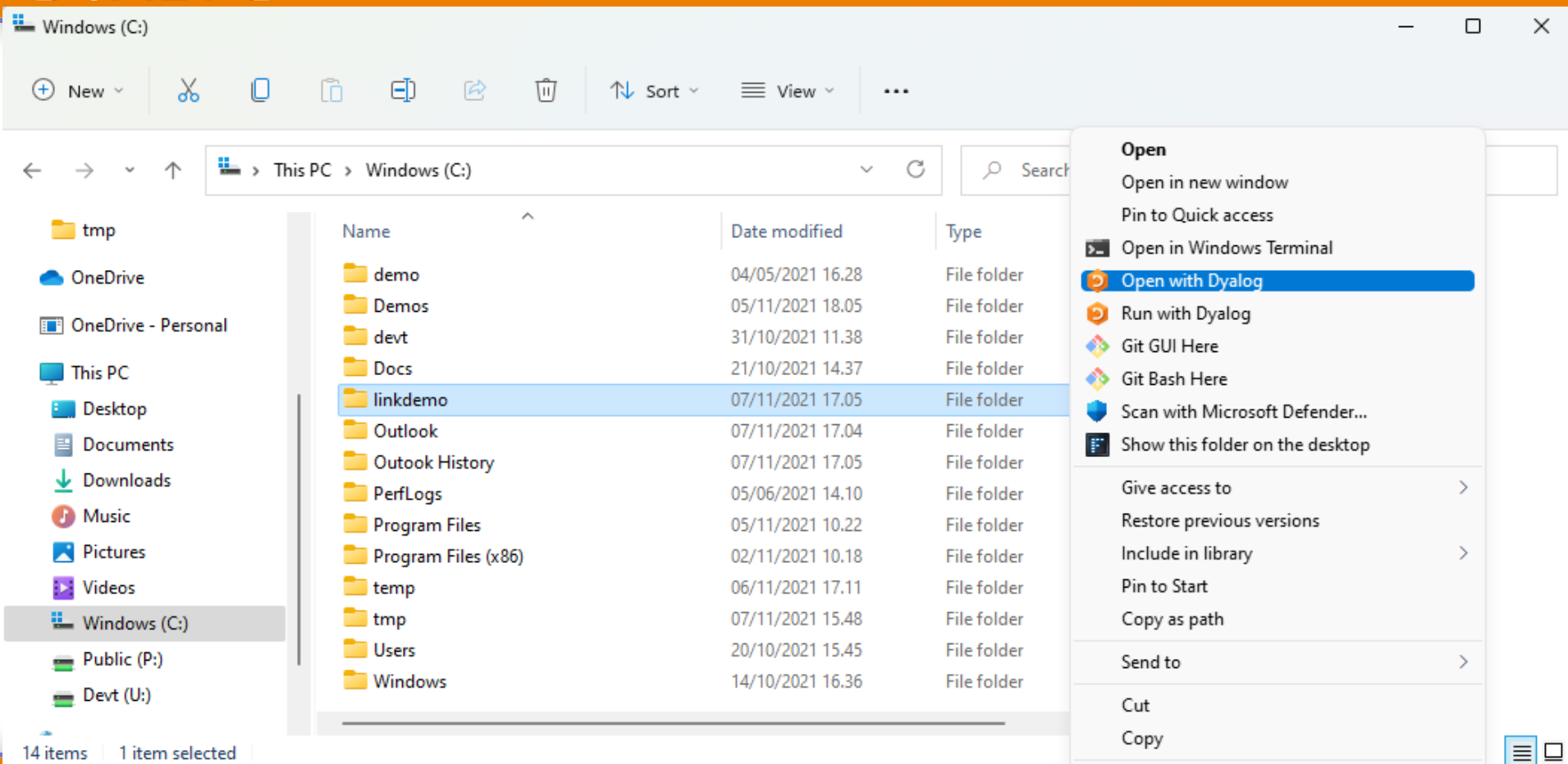
To import code that you do not intend to modify, from C:\linkdemo\myapp into the namespace "myapp":

```
]link.import myapp C:\linkdemo\myapp
```

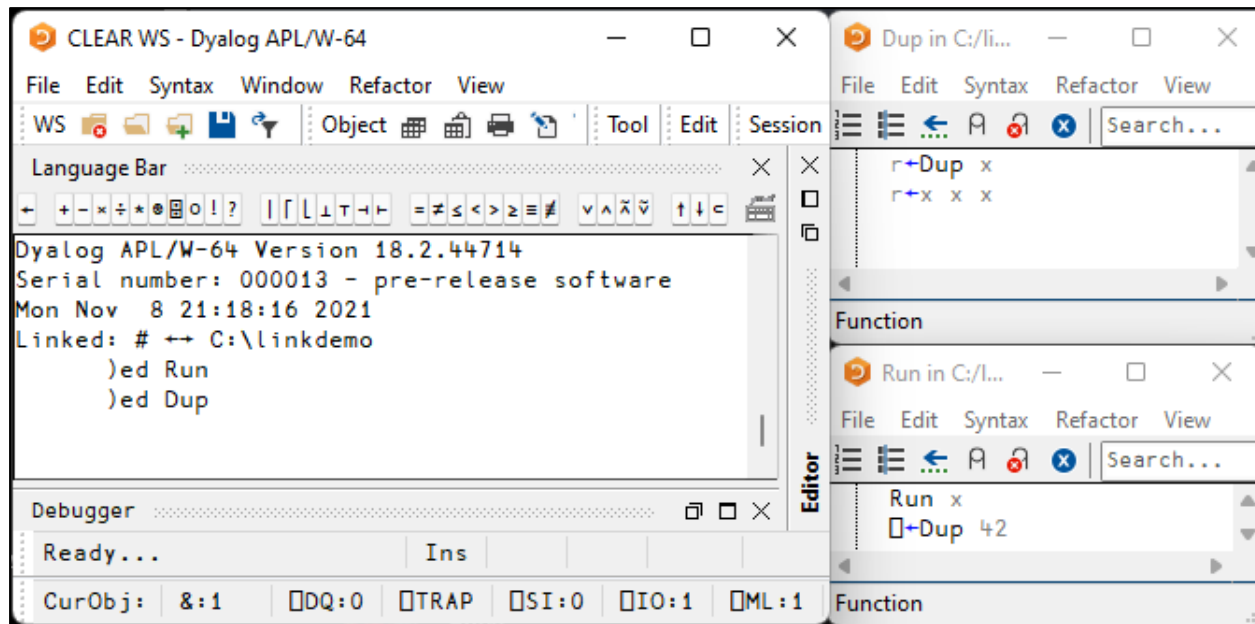
... and then proceed to use APL "normally"

... but without having to remember to type)SAVE at the end of your day.

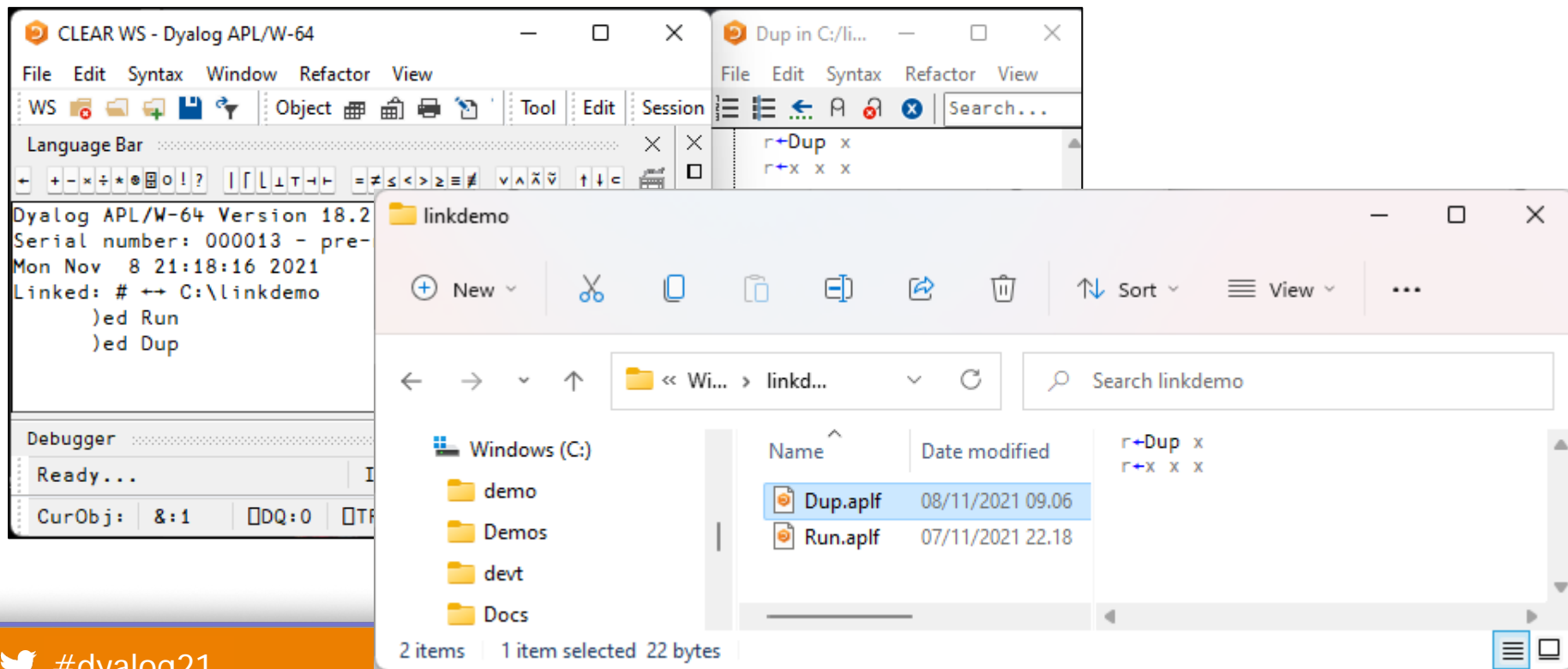




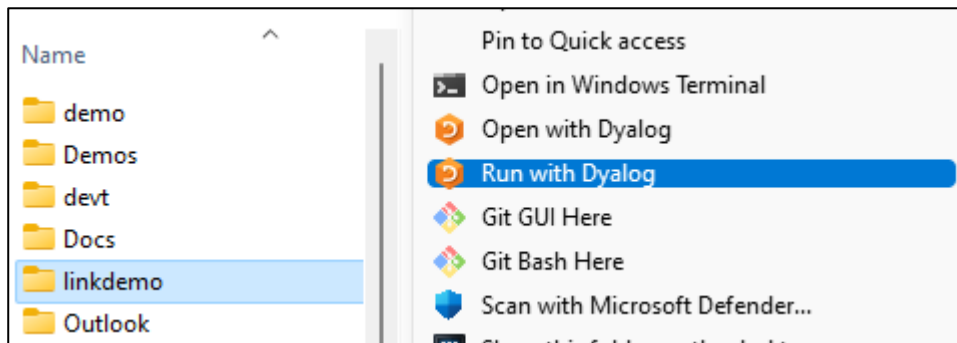
Workspace + Source are Synchronised



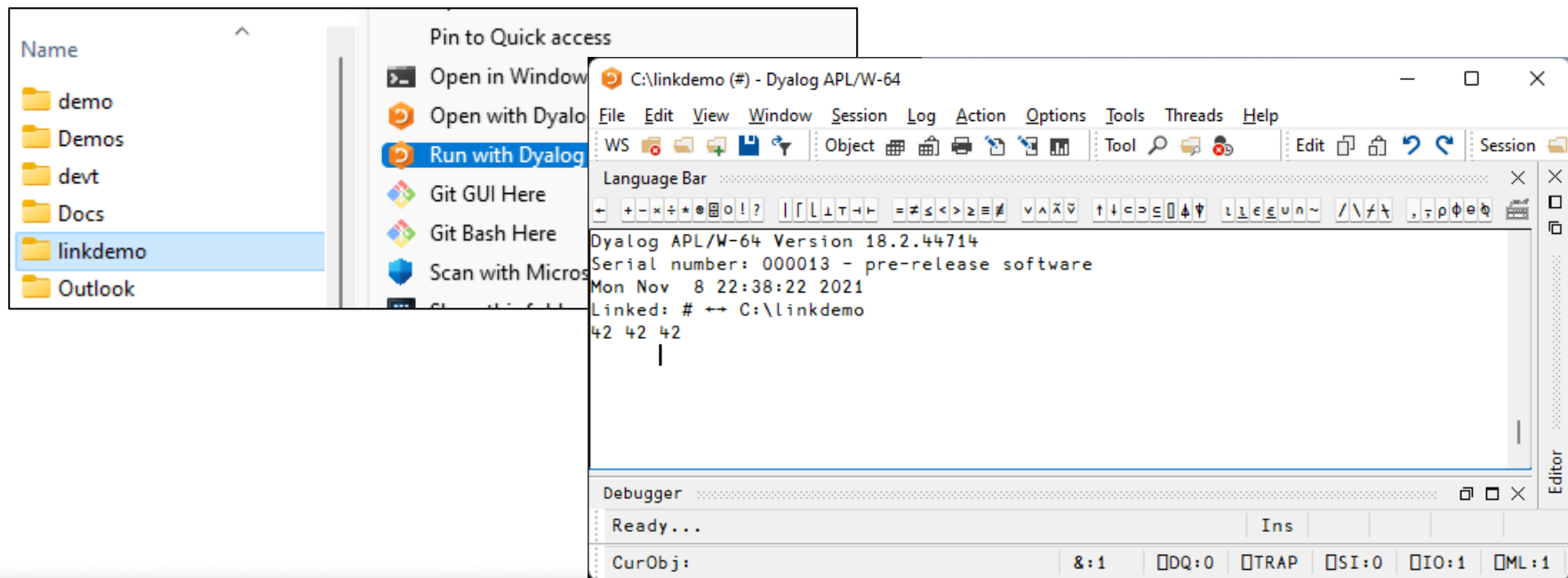
Workspace + Source are Synchronised



"Run with Dyalog" calls function Run

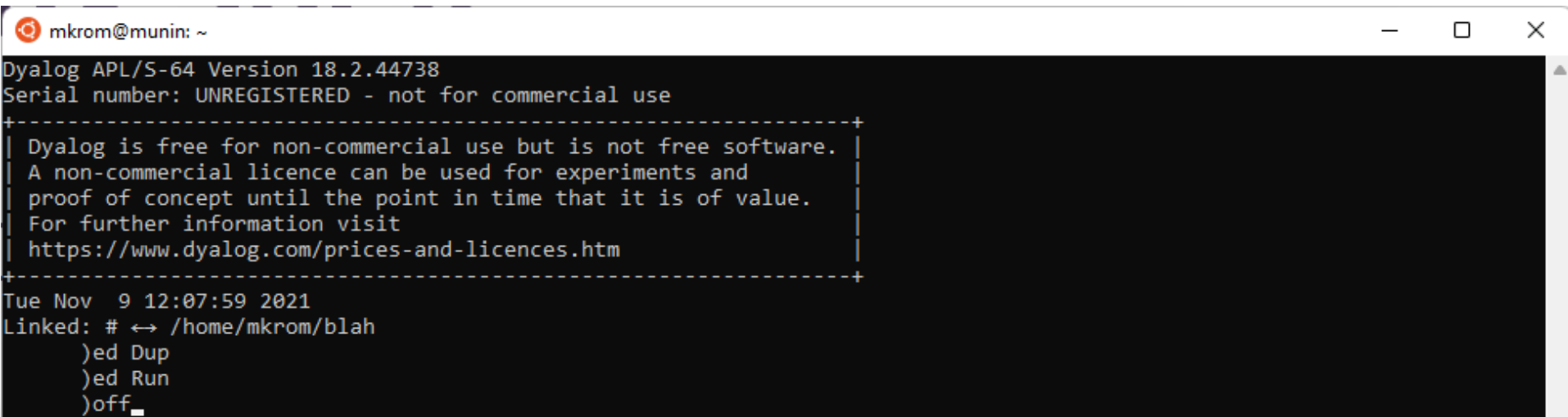


"Run with Dyalog" calls function Run



Linux Shell

```
$ LINK_DIR="blah" dyalog
```

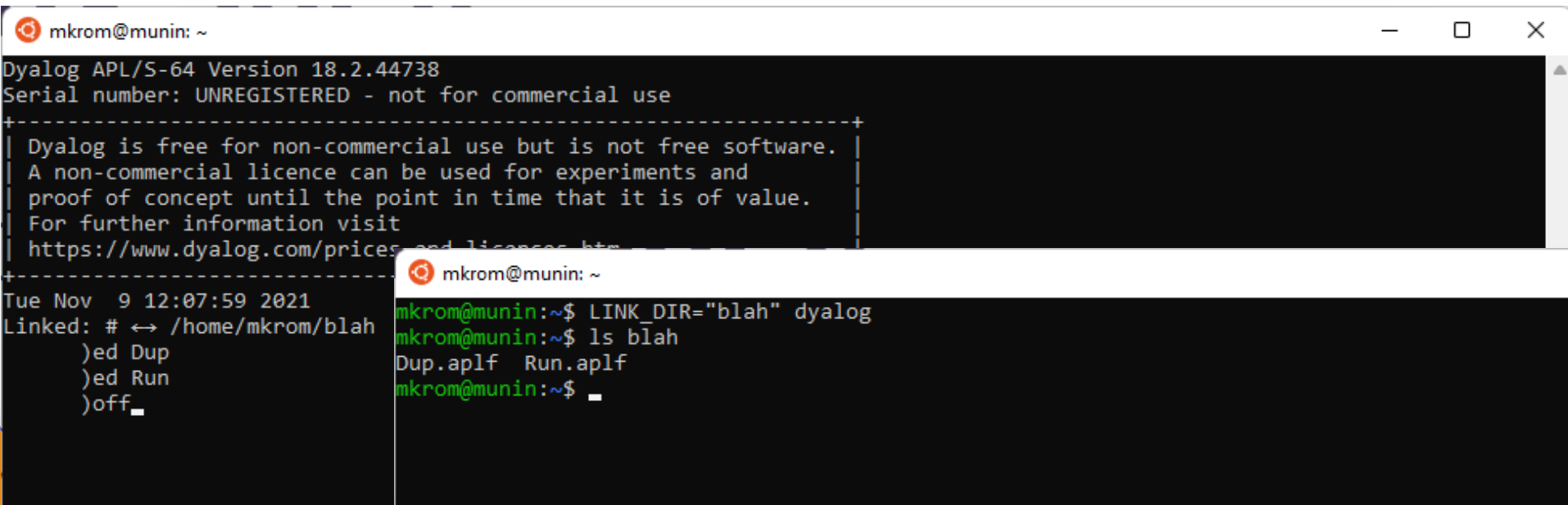


A terminal window titled 'mkrom@munin: ~' with standard window controls. The terminal output shows the Dyalog APL startup sequence, including version and serial number information, followed by a license notice enclosed in a dashed box. The user then enters the command 'LINK_DIR="blah" dyalog', which results in the prompt changing to '# ↔ /home/mkrom/blah'. The user then enters the command 'ed Dup', which is followed by 'ed Run' and 'ed off'.

```
mkrom@munin: ~  
Dyalog APL/S-64 Version 18.2.44738  
Serial number: UNREGISTERED - not for commercial use  
+-----+  
| Dyalog is free for non-commercial use but is not free software. |  
| A non-commercial licence can be used for experiments and      |  
| proof of concept until the point in time that it is of value.  |  
| For further information visit                                    |  
| https://www.dyalog.com/prices-and-licences.htm |  
+-----+  
Tue Nov 9 12:07:59 2021  
Linked: # ↔ /home/mkrom/blah  
    )ed Dup  
    )ed Run  
    )ed off_
```

Linux Shell

```
$ LINK_DIR="blah" dyalog
```

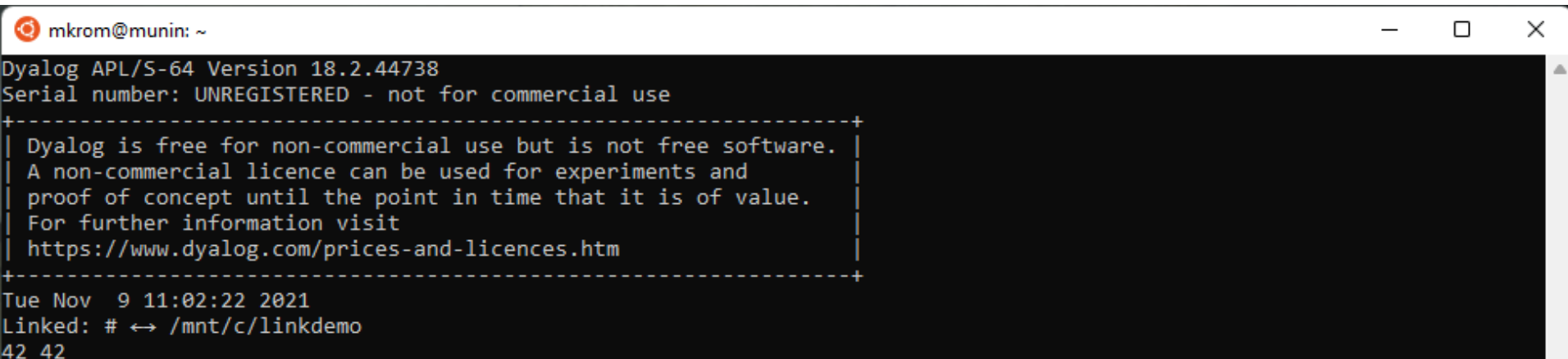


```
mkrom@munin: ~  
Dyalog APL/S-64 Version 18.2.44738  
Serial number: UNREGISTERED - not for commercial use  
-----  
| Dyalog is free for non-commercial use but is not free software. |  
| A non-commercial licence can be used for experiments and    |  
| proof of concept until the point in time that it is of value. |  
| For further information visit                                  |  
| https://www.dyalog.com/prices_and_licenses.htm              |  
-----  
Tue Nov  9 12:07:59 2021  
Linked: # ↔ /home/mkrom/blah  
    )ed Dup  
    )ed Run  
    )off_
```

```
mkrom@munin: ~$ LINK_DIR="blah" dyalog  
mkrom@munin:~$ ls blah  
Dup.aplf  Run.aplf  
mkrom@munin:~$ _
```

Linux Shell

```
$ LINK_RUN="/mnt/c/linkdemo" dyalog
```



A terminal window titled 'mkrom@munin: ~' with standard window controls. The output of the 'dyalog' command is displayed on a black background with white text. The output includes the version '18.2.44738', a notice about the unregistered serial number, a dashed box containing licensing information, the date 'Tue Nov 9 11:02:22 2021', the linked path, and the number '42 42'.

```
mkrom@munin: ~  
Dyalog APL/S-64 Version 18.2.44738  
Serial number: UNREGISTERED - not for commercial use  
+-----+  
| Dyalog is free for non-commercial use but is not free software. |  
| A non-commercial licence can be used for experiments and      |  
| proof of concept until the point in time that it is of value.  |  
| For further information visit                                    |  
| https://www.dyalog.com/prices-and-licences.htm |  
+-----+  
Tue Nov 9 11:02:22 2021  
Linked: # ↔ /mnt/c/linkdemo  
42 42
```

Converting an Existing System

```
)load c:\demos\linkdemo\myapp
```

```
)fns
```

```
Main      Mean      Root      StdDev
```

```
; "1 1 0 0<CR"<NL -3
```

```
Main;data
A Compute Mean and StdDev until user inputs an empty array

:Repeat
  <+ 'Enter some numbers:'
  :If 0<#data+<
    <+ 'Mean: ' ,1<Mean data
    <+ 'StdDev: ' ,1<StdDev data
  :EndIf
:Until 0<#data
```

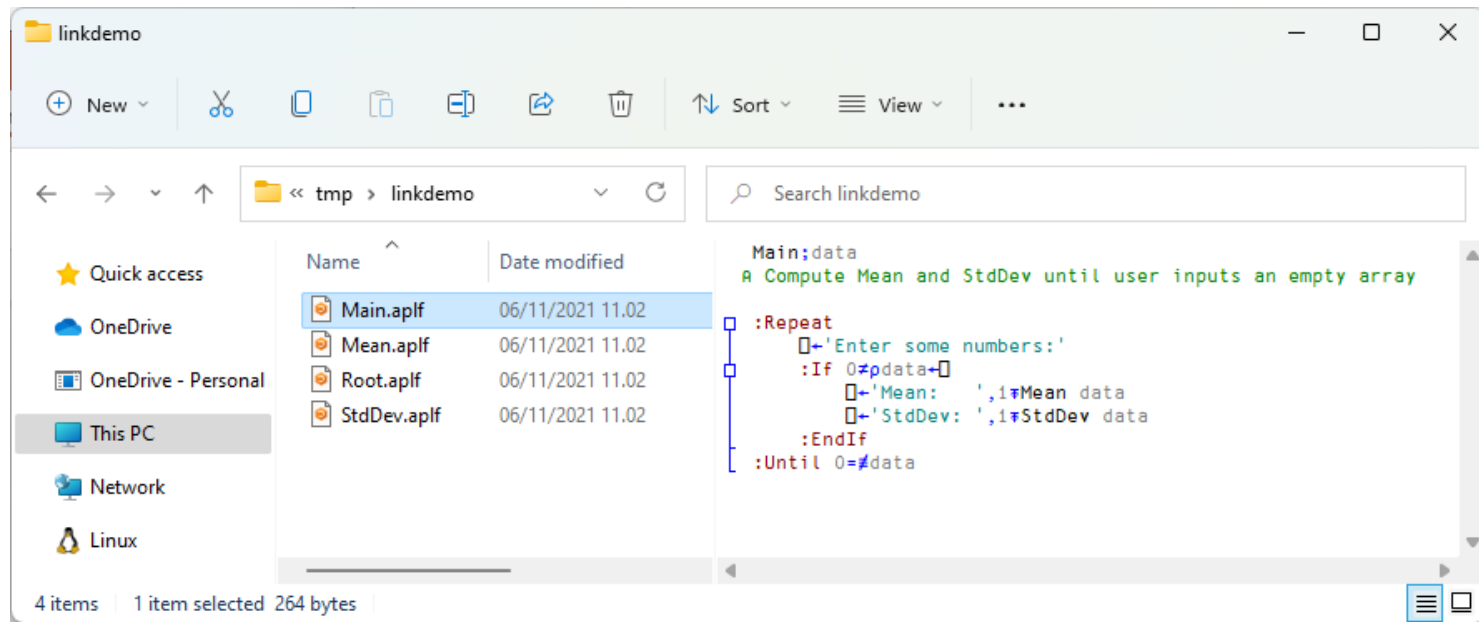
```
mean+Mean vals;sum
sum++/,vals
mean+sum÷1<#,vals
```

```
Root+{<+2
      <*<÷<}
```

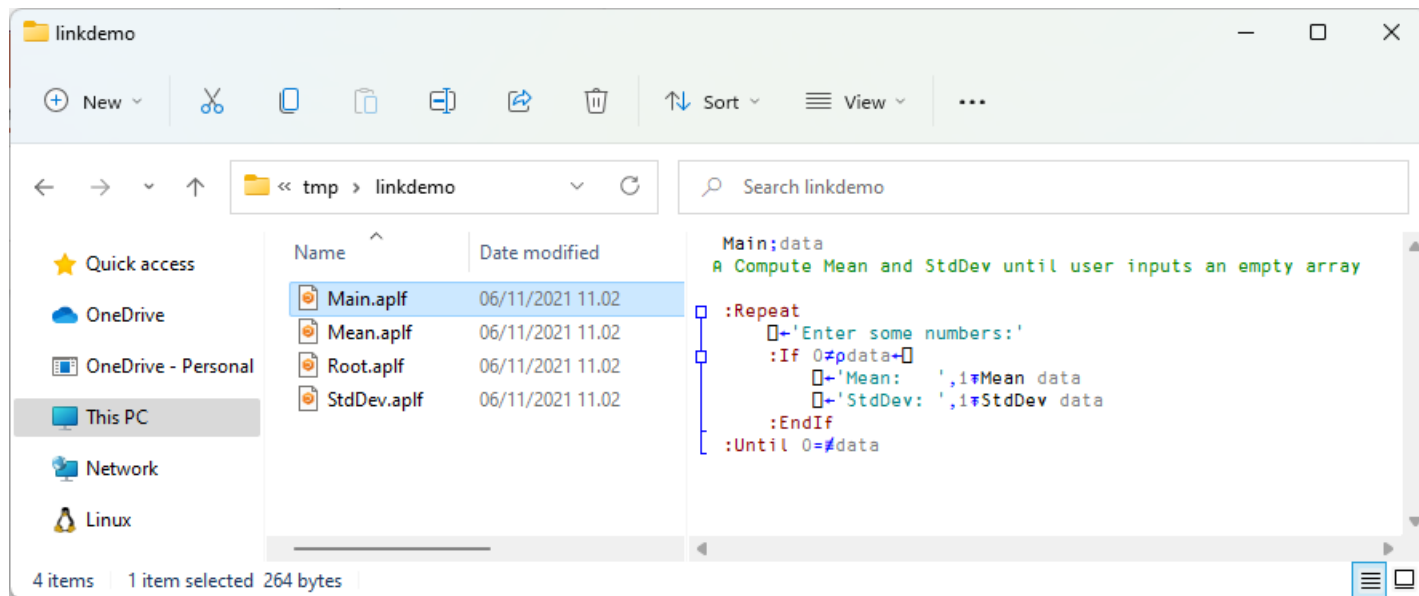
```
StdDev+{2 Root(+.<÷÷<),<-Mean <}
```

Exporting

```
]link.export # c:\tmp\linkdemo  
Exported: # → c:\tmp\linkdemo
```

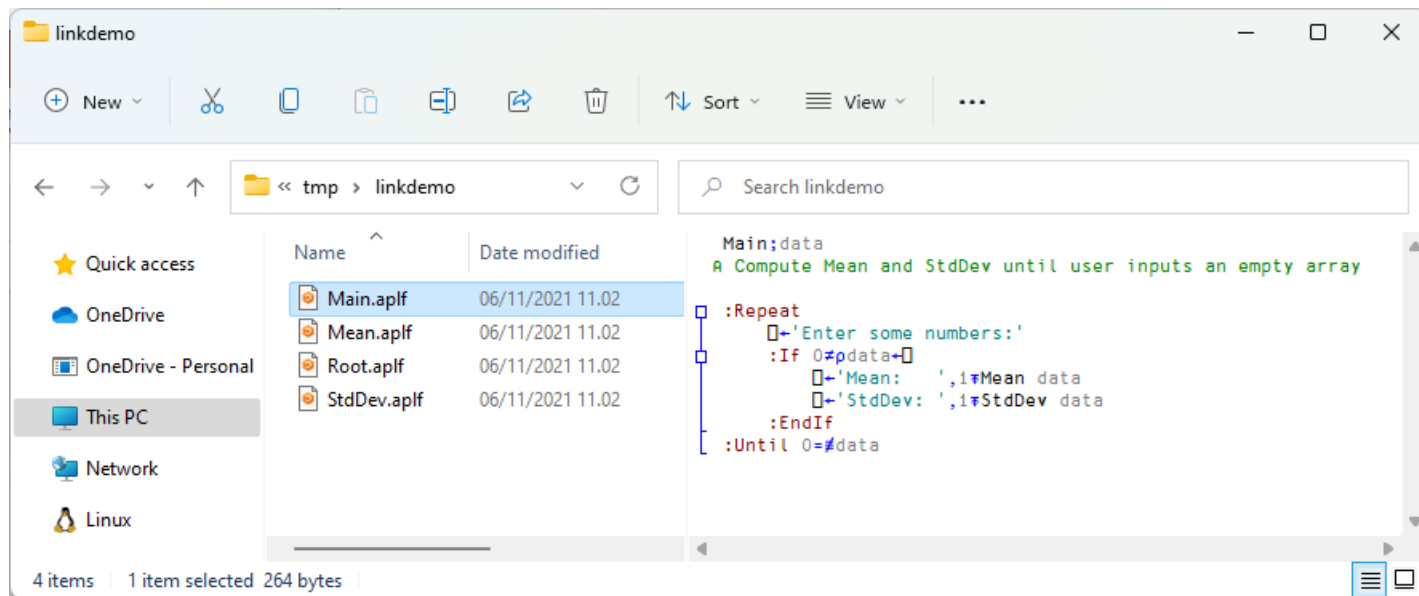


[System] Variables



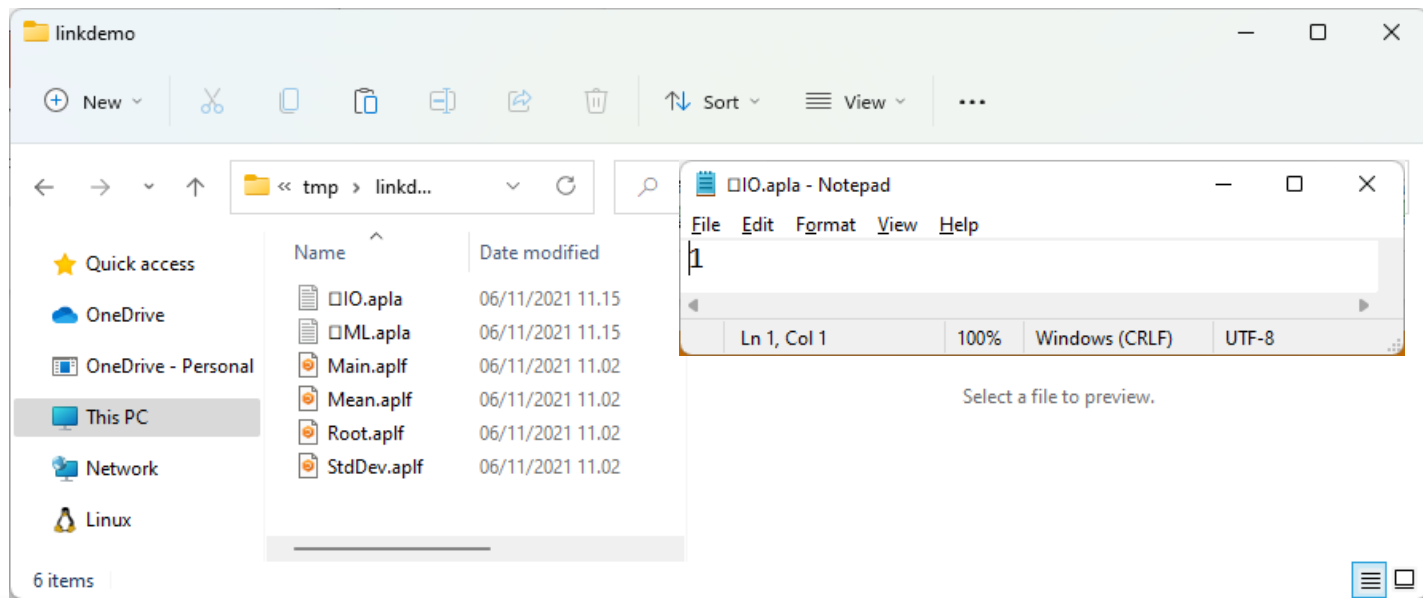
[System] Variables

```
]link.add IO ML  
Added: #.IO #.ML
```



[System] Variables

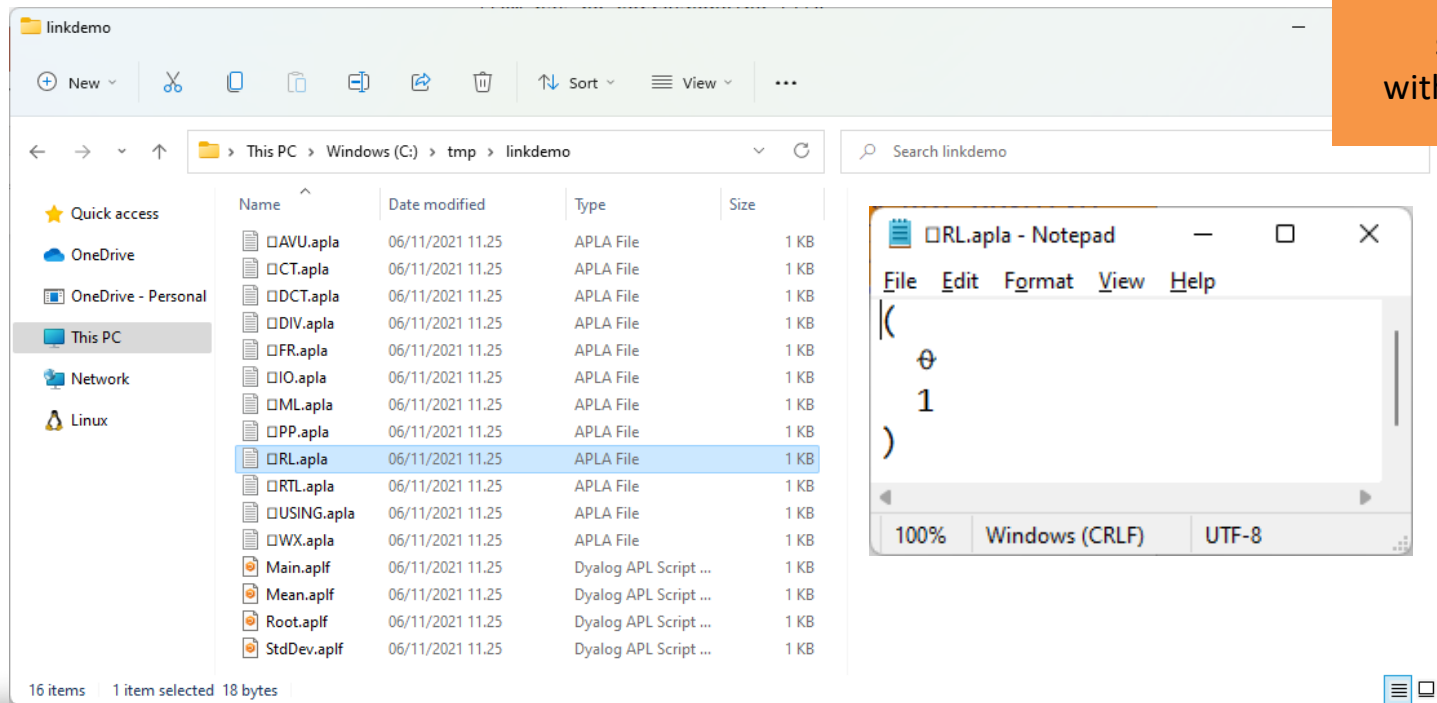
```
]link.add IO ML  
Added: #.IO #.ML
```



-sysvars

```
]export # c:\tmp\linkdemo -sysvars
Exported: # → c:\tmp\linkdemo
```

Exports all
system variables
with namespace scope



]link.create replaces)LOAD

```
]link.create # c:\tmp\linkdemo
Linked: # ↔ c:\tmp\linkdemo
```

```
)fns
```

```
Main      Mean      Root      StdDev
```

```
Main
```

```
Enter some numbers:
```

```
□:
```

```
1 2 3 4
```

```
Mean:      2.5
```

```
StdDev:    1.1
```

```
Enter some numbers:
```

```
□:
```

```
0
```



Shows that link is
Bi-directional

(Currently requires .NET)

Adding Structure

```
)fns
Main      Mean      Root      StdDev
```

```
; "1 1 0 0<CR"<NL -3
```

```
Main;data
A Compute Mean and StdDev until user inputs an empty array

:Repeat
  <+ 'Enter some numbers:'
  :If 0≠pdata+<
    <+ 'Mean:   ',1≠Mean data
    <+ 'StdDev: ',1≠StdDev data
  :EndIf
:Until 0=≠data
```

```
mean+Mean vals;sum
sum+≠,vals
mean+sum÷1[ρ,vals
```

```
Root+{α+2
      ω*÷α}
```

```
StdDev+{2 Root(+,×÷÷ρ),ω-Mean ω}
```

Adding Structure

CLEAR WS Exploring [#]

File Edit Options View Tools

Workspace Tree

- #
 - myapp
 - [Fns/Ops]
 - Main
 - stats
 - [Fns/Ops]
 - Mean
 - Root
 - StdDev

Contents of # (Global Scope)

Name	Location	Type
myapp	#	Namespace
stats	#	Namespace

2 object(s). 253.7Mb free. 9696 bytes used (0 bytes selected)

```

mean←Mean vals;sum
sum←+,vals
mean←sum÷1[ρ,vals

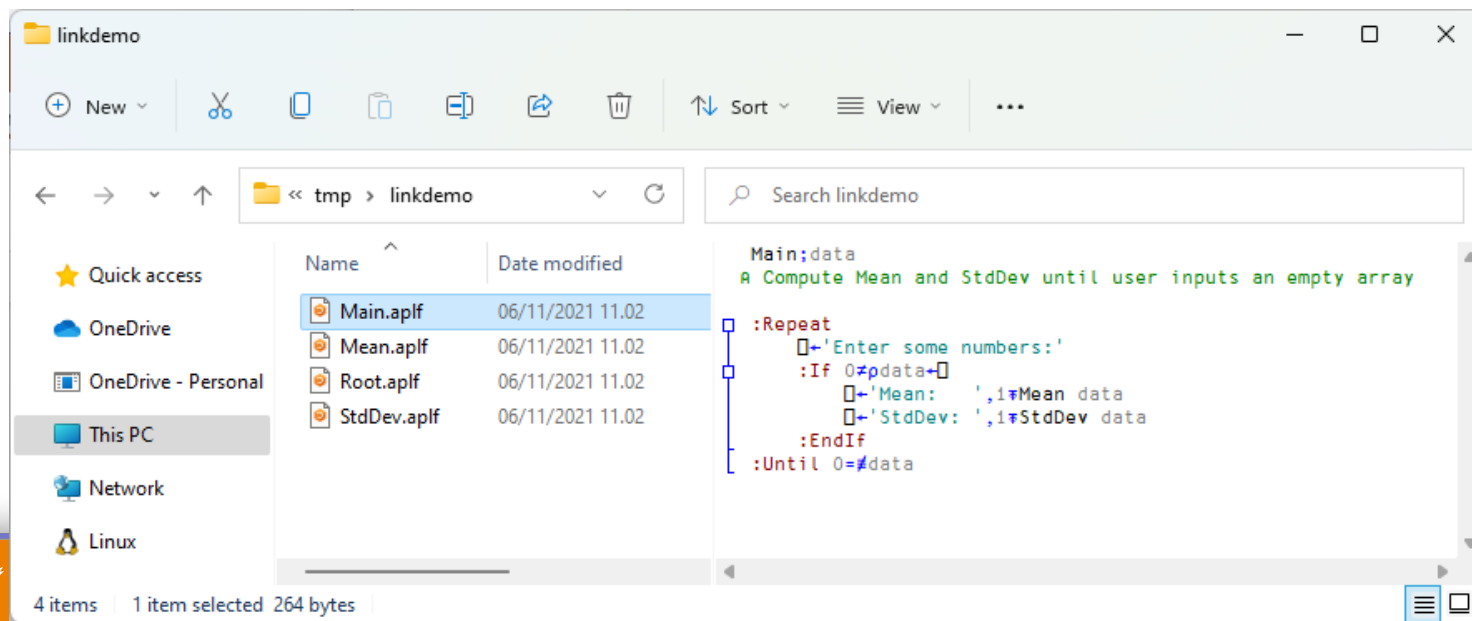
Root←{α+2
      ω*÷α}

StdDev←{2 Root(+,×÷÷ρ),ω-Mean ω}

```

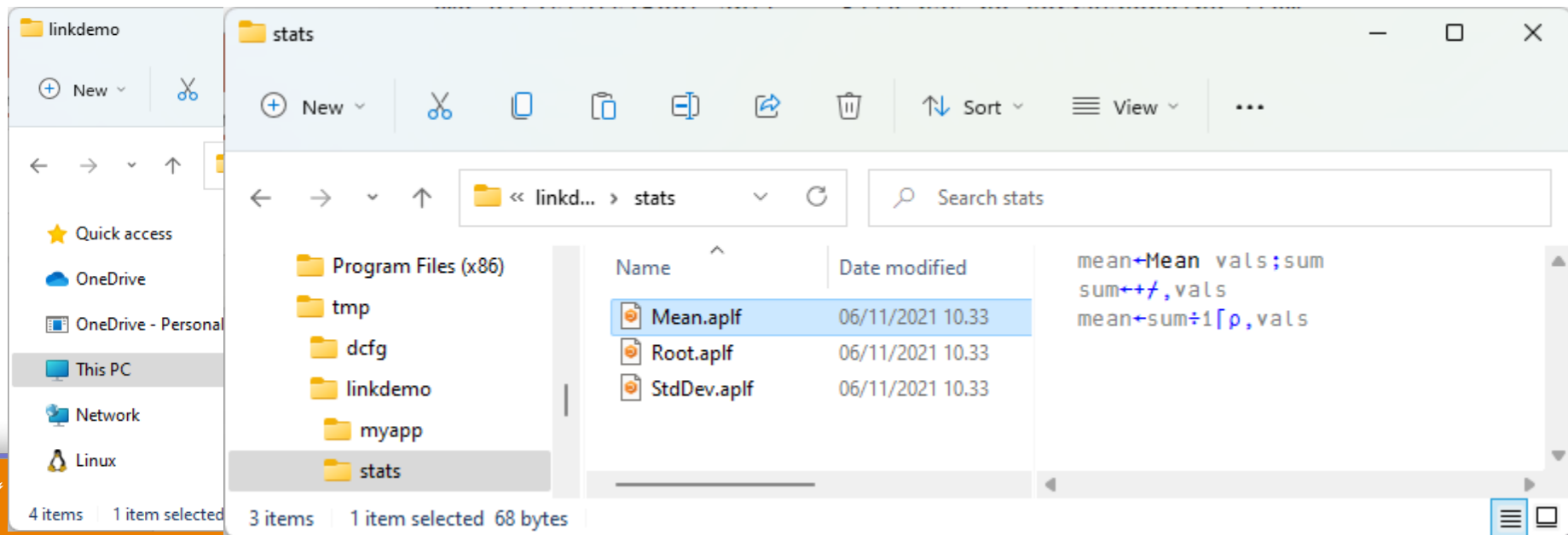
Adding Structure

- You could just use your file manager to restructure the directory



Adding Structure

- You could just use your file manager to restructure the directory



OR...

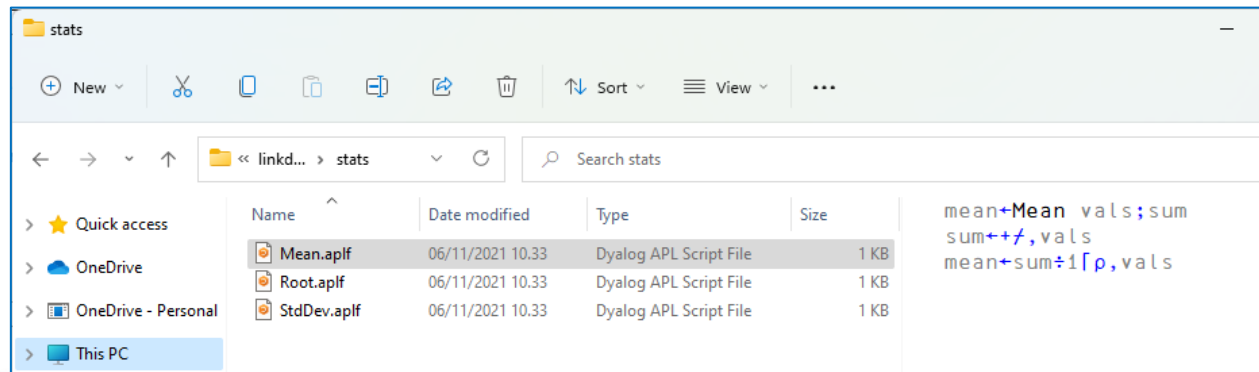
```

3 ⌘MKDIR 'c:\tmp\linkdemo\myapp'
3 ⌘MKDIR 'c:\tmp\linkdemo\stats'

]link.export Main c:\tmp\linkdemo\myapp
Exported: #.Main → c:/tmp/linkdemo/myapp/Main.aplf

statfns←'Mean' 'Root' 'StdDev'
;{⌘SE.Link.Export ω 'c:\tmp\linkdemo\stats'}"statfns
Exported: #.Mean → c:/tmp/linkdemo/stats/Mean.aplf
Exported: #.Root → c:/tmp/linkdemo/stats/Root.aplf
Exported: #.StdDev → c:/tmp/linkdemo/stats/StdDev.aplf

```



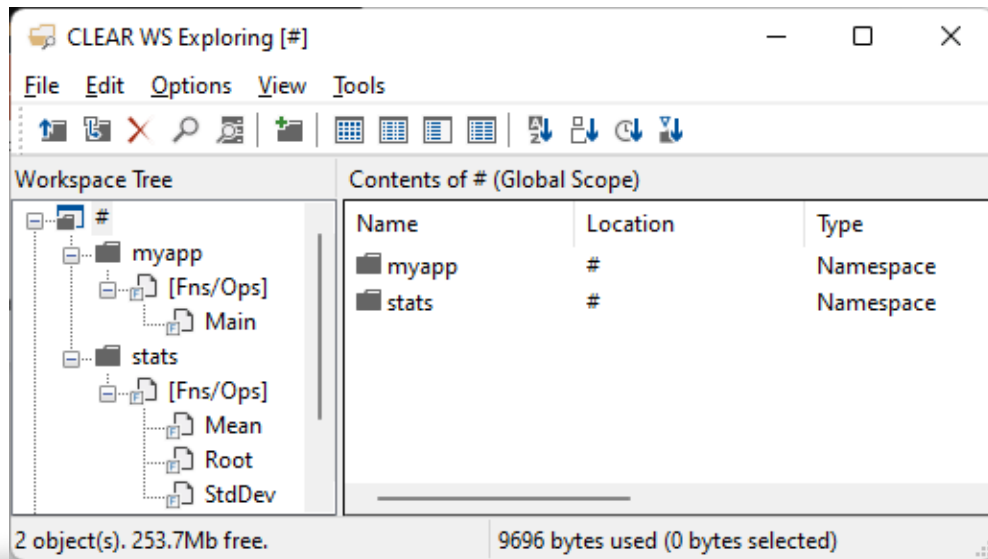
Adding Structure

Language Bar

Dyalog APL/W-64 Version 18.2.4714
 Serial number: 000013 - pre-release software
 Mon Nov 8 08:56:13 2021

```

    ]link.create # c:\tmp\linkdemo
Linked: # ↔ c:\tmp\linkdemo
    myapp.Main
Enter some numbers:
    10?10
VALUE ERROR: Undefined name: Mean
Main[6] ⍵←'Mean: ',1⍴Mean data
    ^
  
```



Namespace
structure
≡
Directory
Structure *

Adding Structure

Language Bar

Dyalog APL/W-64 Version 18.2.4714
 Serial number: 000013 - pre-release software
 Mon Nov 8 08:56:13 2021

```

    ]link.create # c:\tmp\linkdemo
Linked: # ↔ c:\tmp\linkdemo
myapp.Main
Enter some numbers:
[]:
    10?10
VALUE ERROR: Undefined name: Mean
Main[6] []←'Mean: ',1⍷Mean data
    ^
  
```

Debugger

Search...

Main;data;ST
 A Compute Mean and StdDev until user inputs an empty array

```

    ST+#.stats
  :Repeat
    []←'Enter some numbers:'
  :If 0≠pdata+[]
    []←'Mean: ',1⍷ST.Mean data
    []←'StdDev: ',1⍷ST.StdDev data
  :EndIf
  :Until 0=ndata
  
```

Modified Function Pos: 8/11,27

Ready...

CurObj: STStdDev (Undefined)

STstack (Tid: Txd:0)

Main[6]*[]←'Mean: '

Ins

8:1 DDQ:0 TRAP SI:

Link vs. Workspaces

The screenshot shows the Dyalog APL/W-64 IDE interface. The title bar indicates the file is 'c:\tmp\distraction.dws - Dyalog APL/W-64'. The menu bar includes File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, and Help. The toolbar has icons for workspace operations (WS), object operations (Object), and editing (Edit). The Language Bar shows various APL symbols. The main editor area contains the following code:

```

]link.create # c:\tmp\linkdemo
Linked: # ↔ c:\tmp\linkdemo
data←10?10
)save c:\tmp\distraction
c:\tmp\distraction.dws saved Mon Nov 8 10:00:23 2021
)off
  
```

The Debugger window at the bottom shows 'Ready...' and 'Ins' (inspect) mode. The status bar at the bottom indicates 'CurObj: distraction (Und &:1)' and various flags: ☐DQ:0, ☐TRAP, ☐SI:0, ☐IO:1, ☐ML:1.



Link vs. Workspaces

The image displays two overlapping screenshots of the Dyalog APL/W-64 IDE interface, illustrating the difference between workspaces and links.

Left Screenshot (Workspace Management):

- Title Bar:** c:\tmp\distraction.dws - Dyalog APL/W-64
- Menu Bar:** File Edit View Window Session Log Action Options Tools Threads Help
- Toolbar:** WS, Object, Tool, Edit, Session
- Language Bar:** Contains various APL symbols and navigation icons.
- Editor:**

```

]link.create # c:\tmp\linkdemo
Linked: # ↔ c:\tmp\linkdemo
data←10?10
)save c:\tmp\distraction
c:\tmp\distraction.dws saved Mon Nov 8 10:
)off

```
- Debugger:** Ready... CurObj: distraction (Und &:1 □DQ:0

Right Screenshot (Link Restoration):

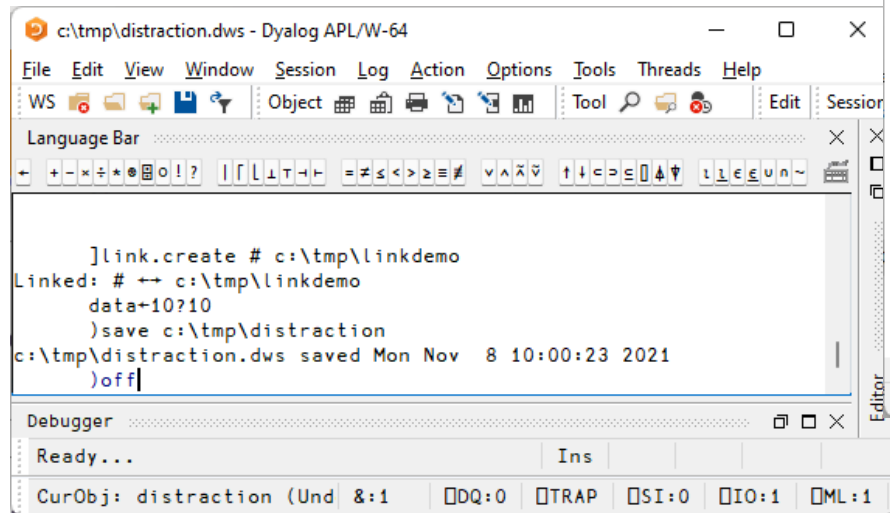
- Title Bar:** c:\tmp\distraction.dws - Dyalog APL/W-64
- Menu Bar:** File Edit View Window Session Log Action Options Tools Threads Help
- Toolbar:** WS, Object, Tool, Edit, Session
- Language Bar:** Contains various APL symbols and navigation icons.
- Editor:**

```

Dyalog APL/W-64 Version 18.2.44714
Serial number: 000013 - pre-release software
Mon Nov 8 10:13:03 2021
)load c:\tmp\distraction.dws
c:\tmp\distraction.dws saved Mon Nov 8 10:11:19 2021
Link Warning: 1 link restored: #
data
7 2 1 8 10 4 6 9 5 3
|

```
- Debugger:** Ready... Ins CurObj: data (Variable) &:1 □DQ:0 □TRAP □SI:0 □IO:1 □ML:1

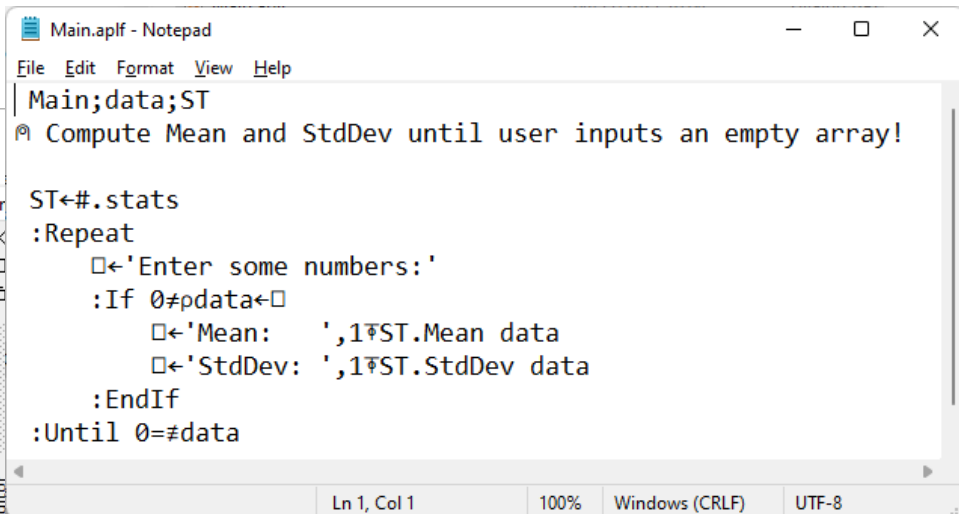
Saved Workspaces



The screenshot shows the Dyalog APL/W-64 workspace editor. The title bar indicates the file is 'c:\tmp\distraction.dws'. The menu bar includes File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, and Help. The toolbar contains icons for workspace operations. The main editor area displays the following APL code:

```
]link.create # c:\tmp\linkdemo  
Linked: # ↔ c:\tmp\linkdemo  
data←10?10  
    )save c:\tmp\distraction  
c:\tmp\distraction.dws saved Mon Nov 8 10:00:23 2021  
    )off|
```

The bottom panel shows the Debugger status, which is 'Ready...'. The status bar at the bottom indicates the current object is 'distraction (Und &:1)' with various flags set to 0 or 1.



The screenshot shows a Notepad window titled 'Main.aplf - Notepad'. The menu bar includes File, Edit, Format, View, and Help. The text content is as follows:

```
Main;data;ST  
Ⓜ Compute Mean and StdDev until user inputs an empty array!  
  
ST←#.stats  
:Repeat  
    □←'Enter some numbers:'  
    :If 0≠pdata←□  
        □←'Mean: ',1⌈ST.Mean data  
        □←'StdDev: ',1⌈ST.StdDev data  
    :EndIf  
:Until 0=#data
```

The status bar at the bottom indicates the cursor is at 'Ln 1, Col 1', the zoom is '100%', the line ending is 'Windows (CRLF)', and the encoding is 'UTF-8'.

Saved Workspaces

The screenshot displays the Dyalog APL/W-64 workspace management interface. The main window, titled 'c:\tmp\distraction.dws - Dyalog APL/W-64', shows a list of saved workspaces. The 'Language Bar' at the top indicates the current workspace is 'distraction.dws'. The 'Debugger' pane at the bottom shows 'Ready...' and 'CurObj: distraction'.

A Notepad window titled 'Main.aplf - Notepad' is open, showing the following code:

```
Main;data;ST
Compute Mean and StdDev until user inputs an empty array!
```

The workspace list shows the following details:

Name	Direction	File	Comments
#.myapp.pmat	+		Item has no corresponding file
#.myapp.Main	+	c:/tmp/linkdemo/myapp/Main.aplf	File now dated Now, WS copy is dated 27 minutes ago

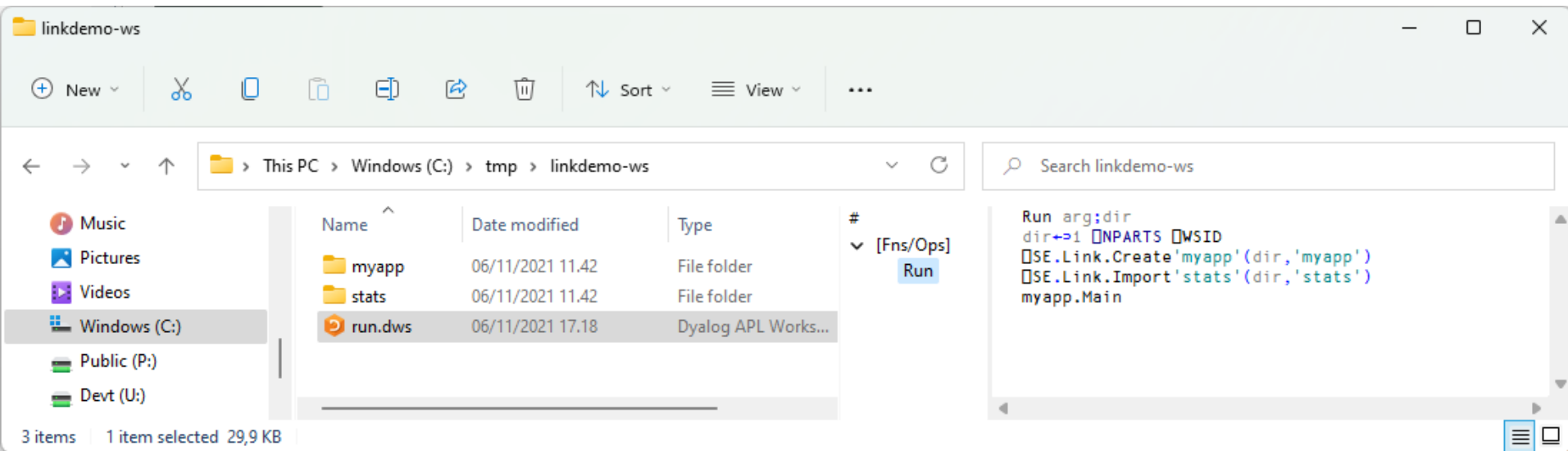
The workspace list also shows the following details:

Name	Direction	File	Comments
#.myapp	+		
#.myapp.pmat	+		
#.myapp.Main	+	c:/tmp/linkdemo/myapp/Main.aplf	File now dated Now, WS copy is dated 27 minutes ago

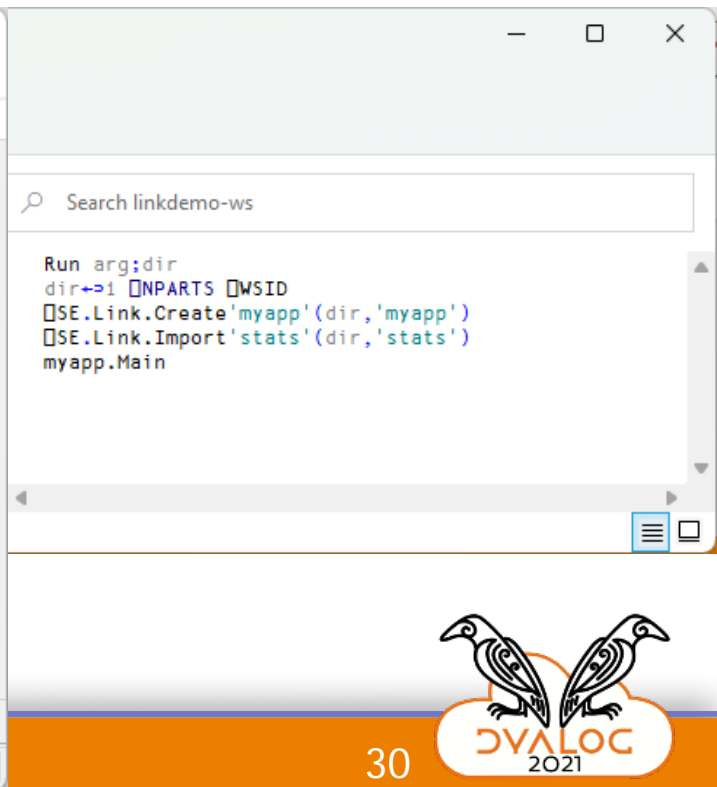
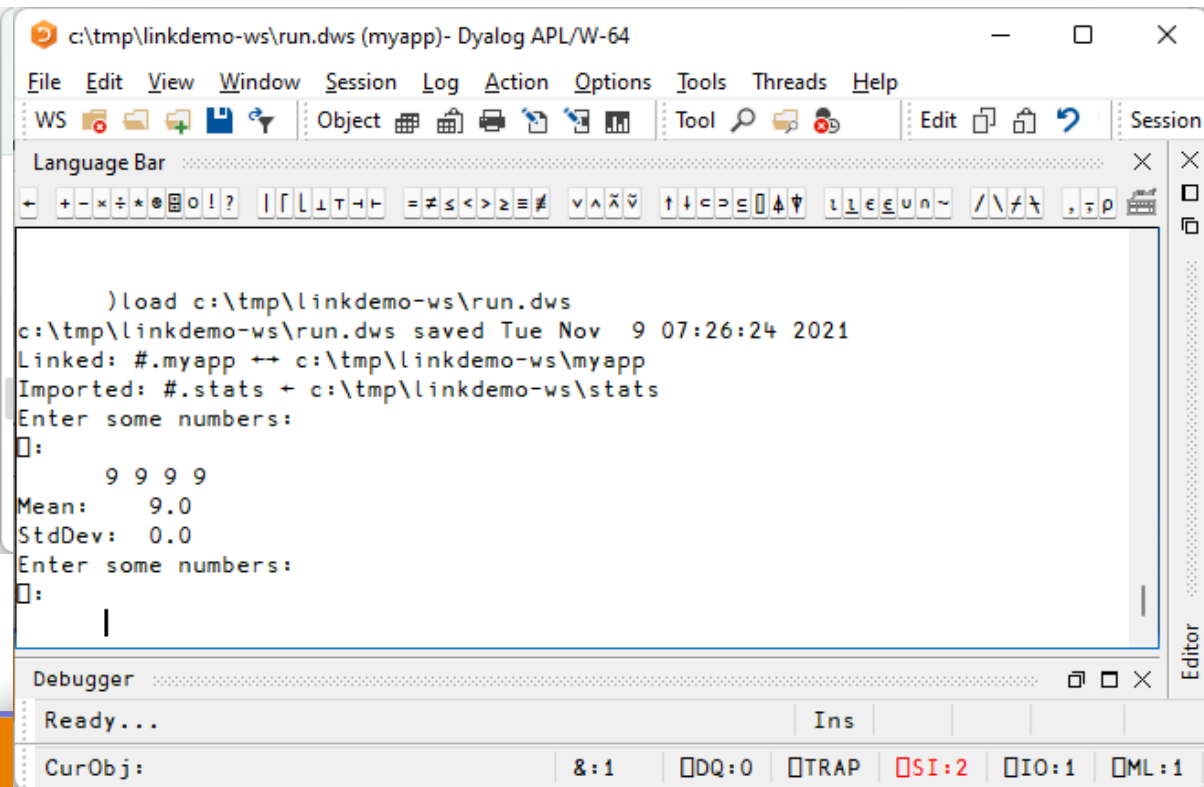
The workspace list also shows the following details:

Name	Direction	File	Comments
#.myapp	+		
#.myapp.pmat	+		
#.myapp.Main	+	c:/tmp/linkdemo/myapp/Main.aplf	File now dated Now, WS copy is dated 27 minutes ago

Launching using a Workspace



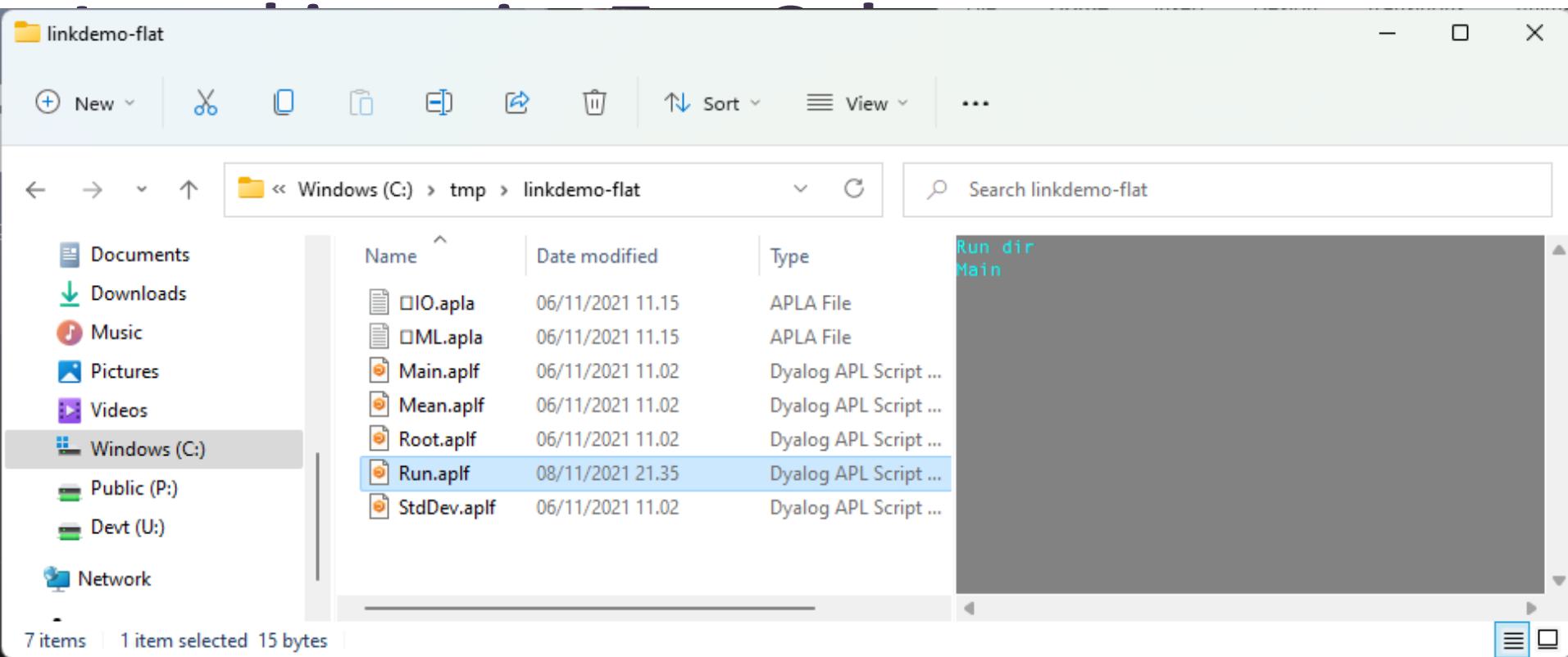
Launching using a Workspace

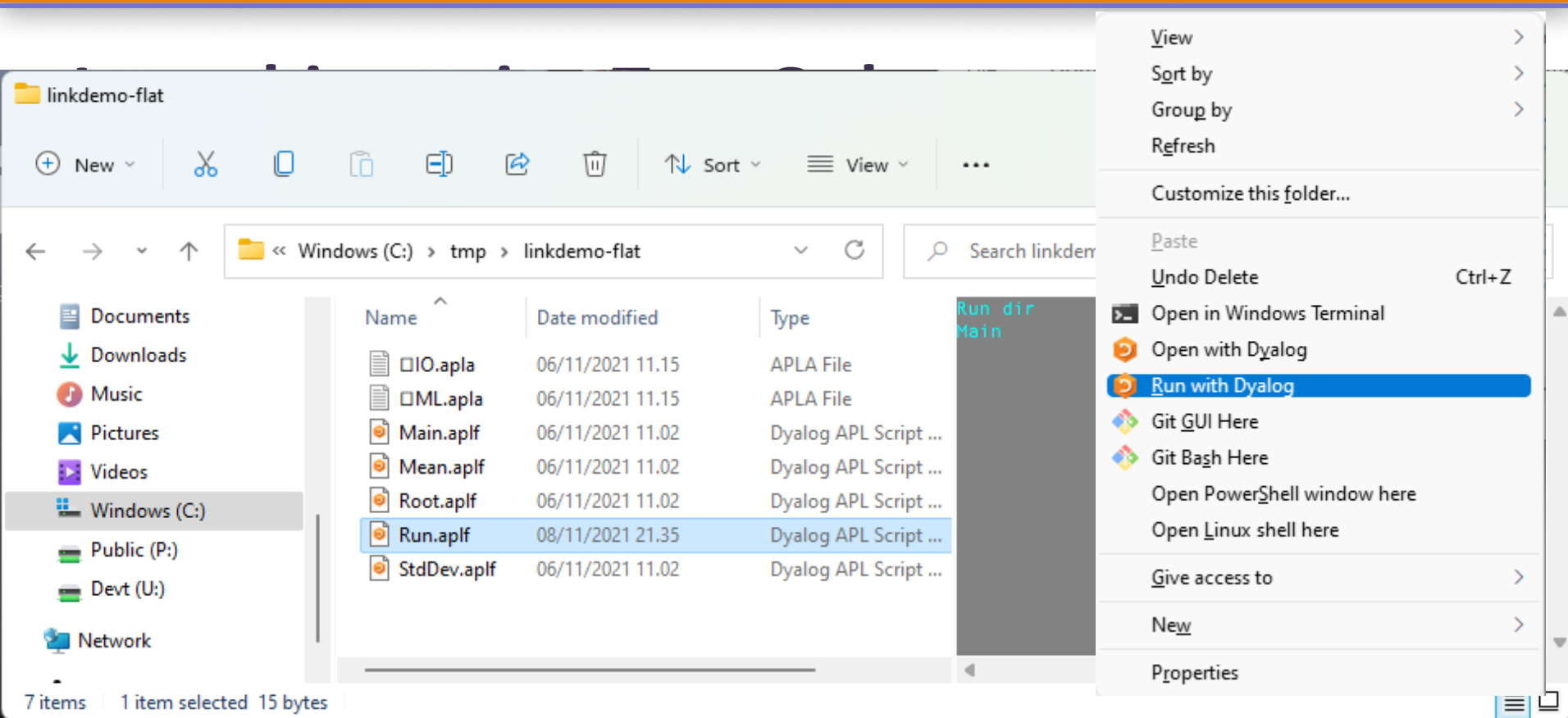


Saved Workspaces

- ✧ OK for pausing and crash analysis
- ✧ Not recommended as permanent storage for source code

Launching using Text Only





linkdemo-flat

New ▾

Sort ▾ View ▾

← → ▾ ↑

« Windows (C:) » tmp » linkdemo-flat

Search linkdemo

Documents

Downloads

Music

Pictures

Videos

Windows (C:)

Public (P:)

DevT (U:)

Network

Name	Date modified	Type
IO.apla	06/11/2021 11.15	APLA File
ML.apla	06/11/2021 11.15	APLA File
Main.aplf	06/11/2021 11.02	Dyalog APL Script ...
Mean.aplf	06/11/2021 11.02	Dyalog APL Script ...
Root.aplf	06/11/2021 11.02	Dyalog APL Script ...
Run.aplf	08/11/2021 21.35	Dyalog APL Script ...
StdDev.aplf	06/11/2021 11.02	Dyalog APL Script ...

7 items | 1 item selected 15 bytes

View ▾

Sort by ▾

Group by ▾

Refresh

Customize this folder...

Paste

Undo Delete Ctrl+Z

Open in Windows Terminal

Open with Dyalog

Run with Dyalog

Git GUI Here

Git Bash Here

Open PowerShell window here

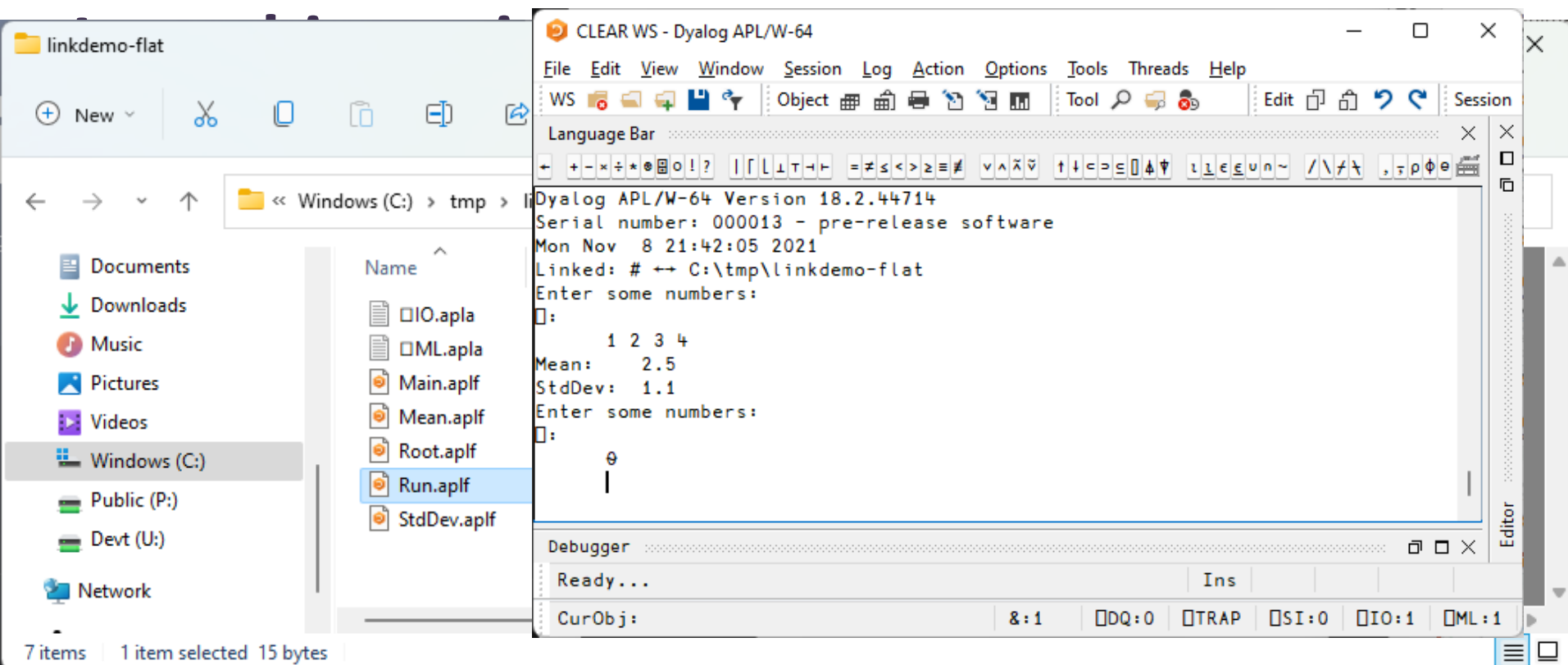
Open Linux shell here

Give access to ▾

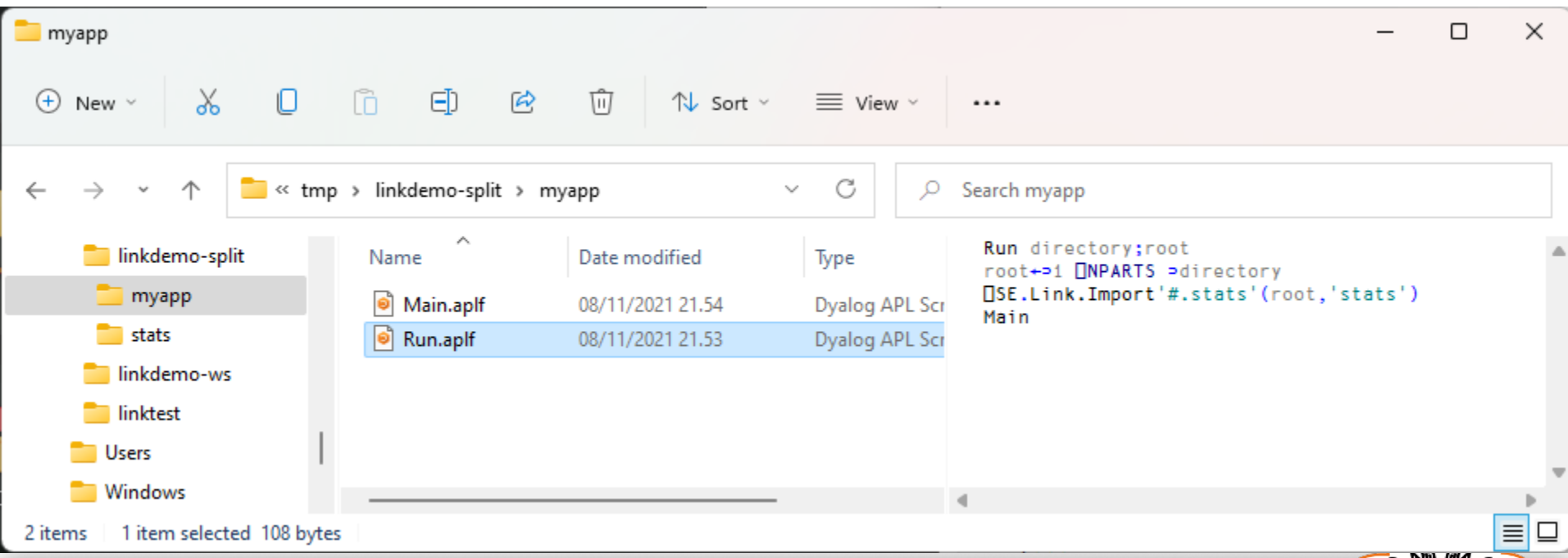
New ▾

Properties

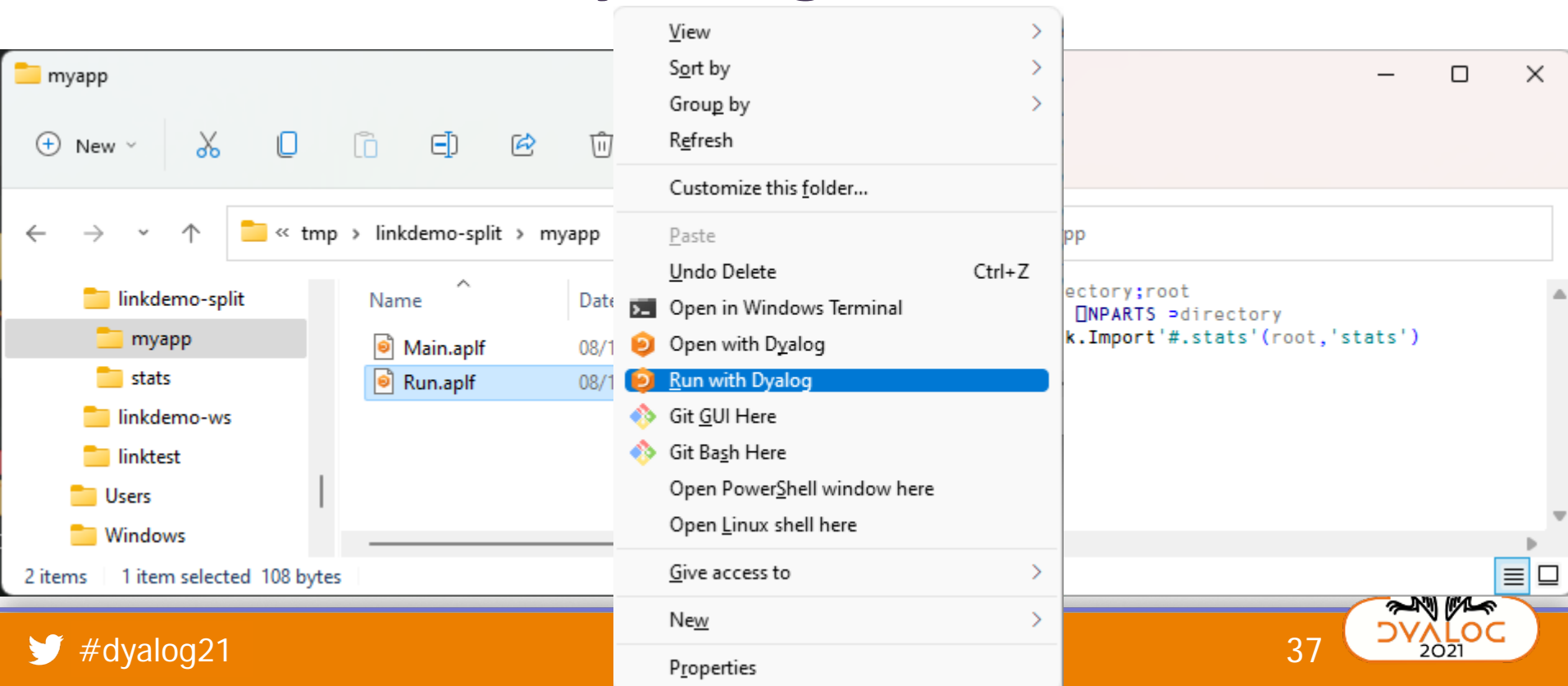
```
Run dir
Main
```



Launch Directory in argument to Run



Launch Directory in argument to Run



The screenshot shows a Windows File Explorer window with the following details:

- Address bar:** < tmp > linkdemo-split > myapp
- Left pane (Navigation):** linkdemo-split, myapp (selected), stats, linkdemo-ws, linktest, Users, Windows
- Right pane (Details):**

Name	Date
Main.aplf	08/1
Run.aplf	08/1
- Status bar:** 2 items, 1 item selected, 108 bytes

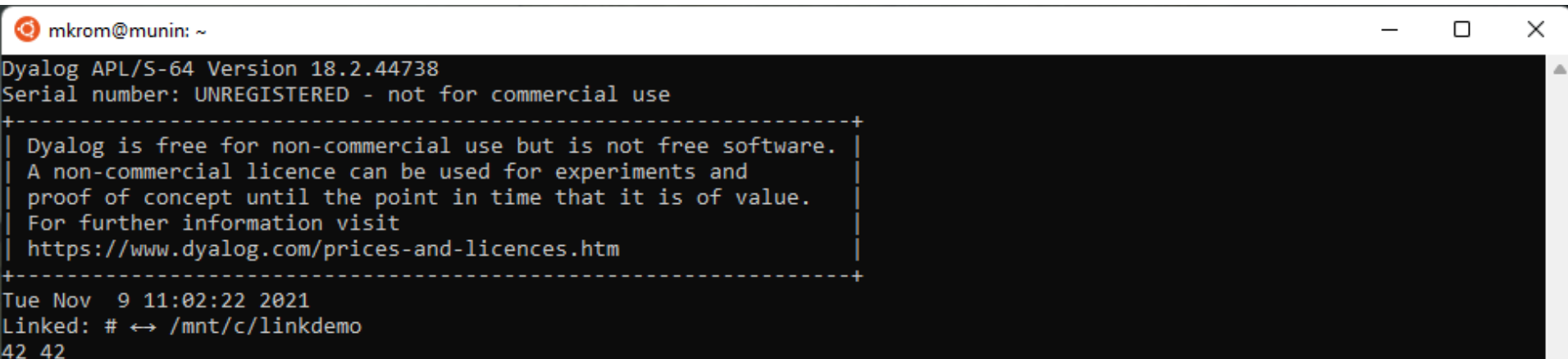
The context menu is open over the 'myapp' folder, with 'Run with Dyalog' highlighted. The background shows a Dyalog IDE window with the following code:

```
directory:root
[NPARTS >directory
k.Import'#.stats'(root,'stats')
```

Launching under Linux / AIX

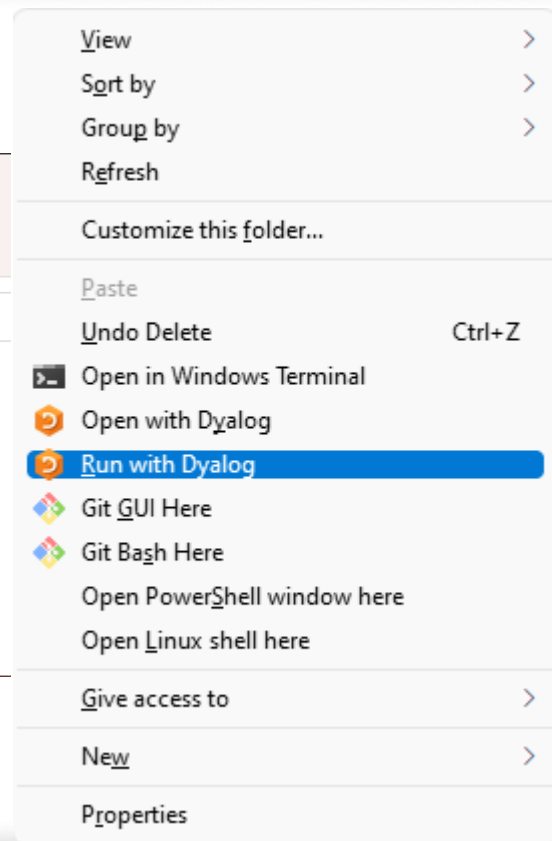
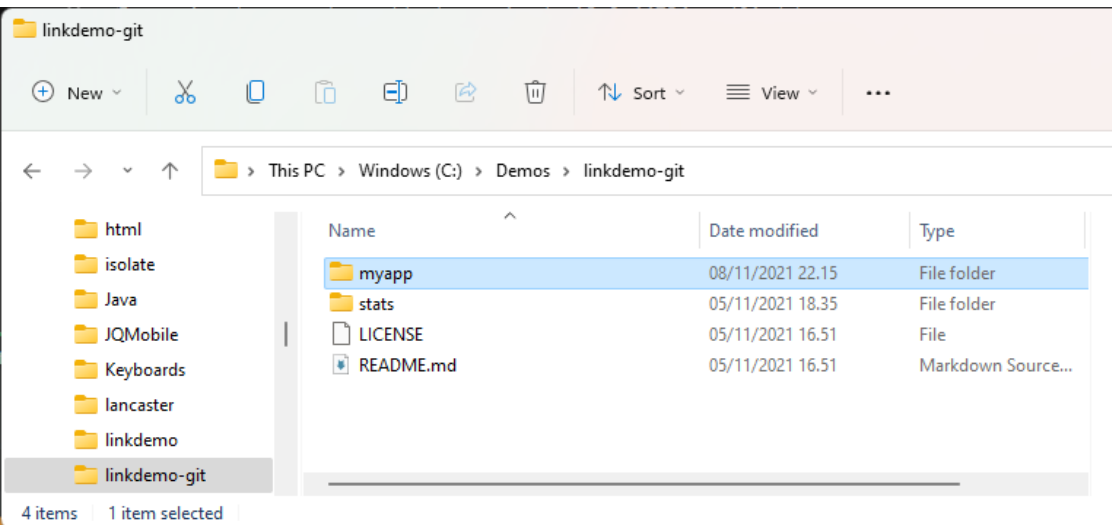
Linux:

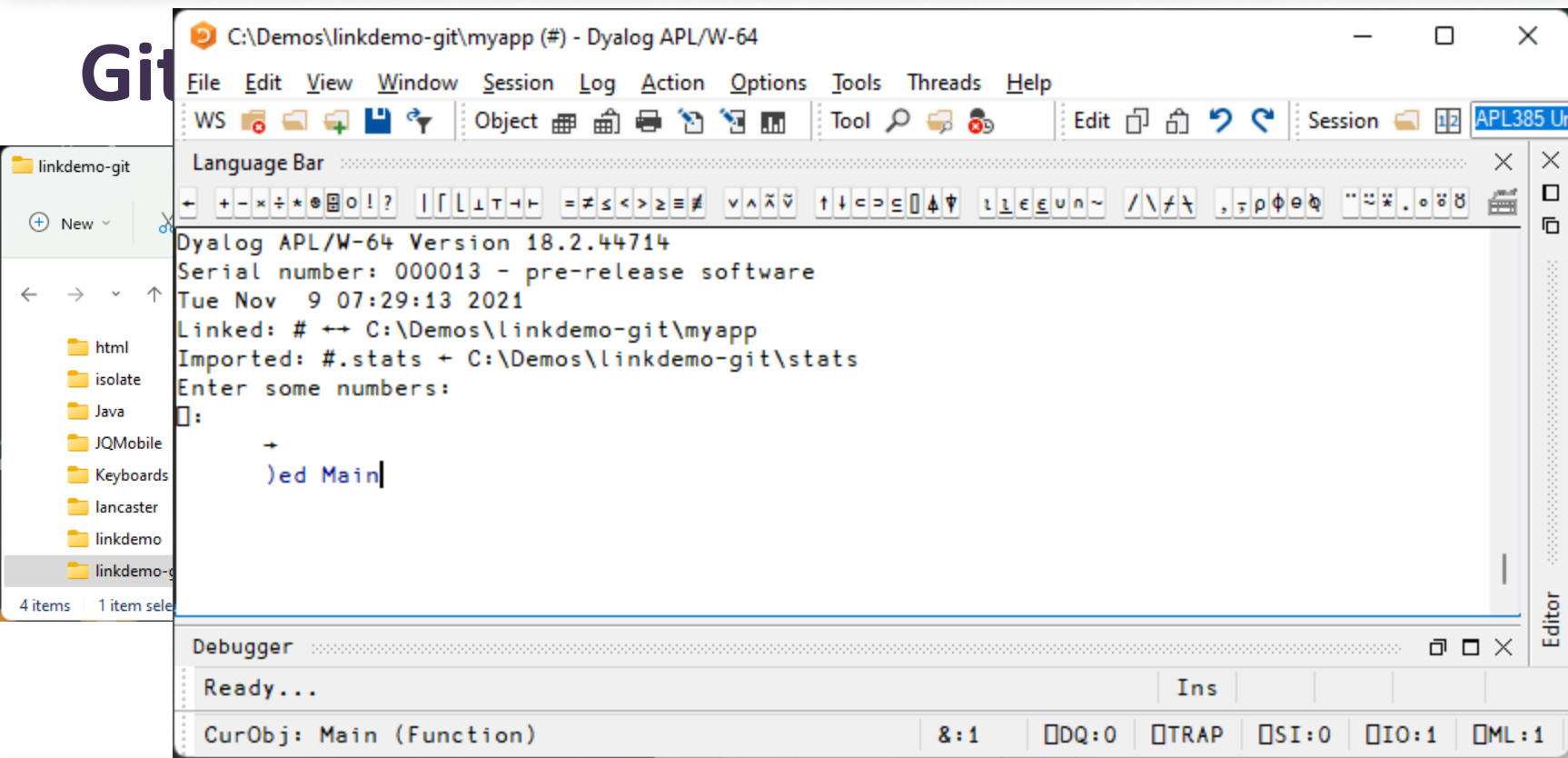
```
$ LINK_RUN="/mnt/c/linkdemo" dyalog
```

A terminal window titled 'mkrom@munin: ~' with standard window controls. The terminal output shows the Dyalog APL startup sequence, including version information, a warning about unregistered use, a license notice enclosed in a dashed box, the current date and time, the linked directory, and the APL version number.

```
mkrom@munin: ~  
Dyalog APL/S-64 Version 18.2.44738  
Serial number: UNREGISTERED - not for commercial use  
+-----+  
| Dyalog is free for non-commercial use but is not free software. |  
| A non-commercial licence can be used for experiments and      |  
| proof of concept until the point in time that it is of value.  |  
| For further information visit                                    |  
| https://www.dyalog.com/prices-and-licences.htm |  
+-----+  
Tue Nov  9 11:02:22 2021  
Linked: # ↔ /mnt/c/linkdemo  
42 42
```

GitHub + Link = True





The screenshot displays the Dyalog Link IDE interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the current file is 'Main.aplf' from the 'linkdemo-git' repository in Visual Studio Code.

The left sidebar contains a 'SOURCE CONTROL' panel with a 'Changes' section showing 'Main.aplf myapp' and a 'COMMITTS' section with a list of recent commits. The main editor area shows two versions of the 'Main.aplf' file side-by-side. The left version is the 'Working Tree' and the right version is a commit from '01:21'.

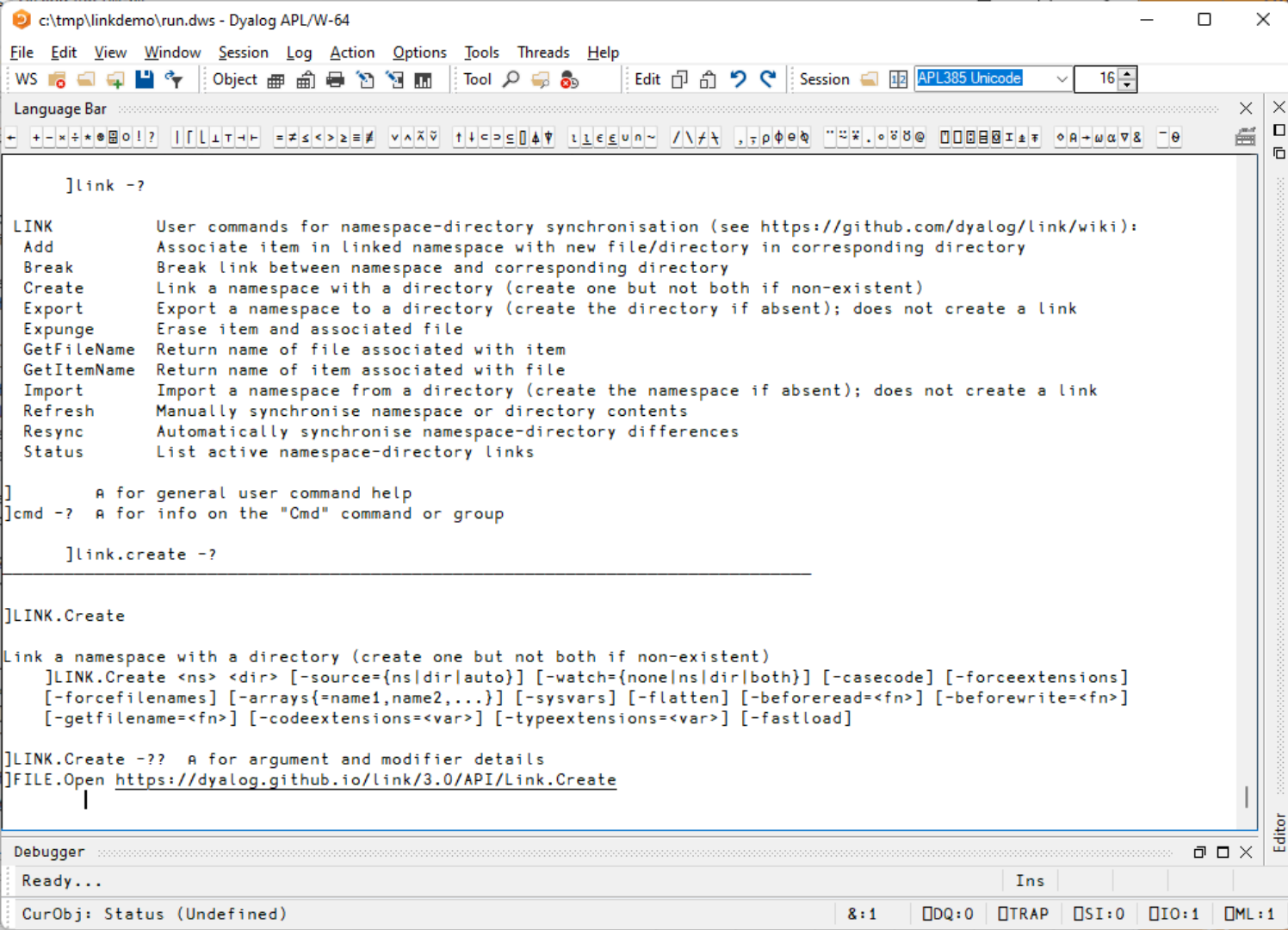
```

1 Main;data
2 A Compute Mean and StdDev until user inputs a
3
4 ST+#.stats
5 :Repeat
6   □←'Enter some numbers:'
7   :If 0≠pdata+□
8     □←'Mean: ',1⌗ST.Mean data
9     □←'StdDev: ',1⌗ST.StdDev data
10  :EndIf
11 :Until 0=≠data
12
13+ □←'Thank you and Goodbye!'
14+

```

The status bar at the bottom shows the current branch as 'main', the commit hash '01:21', and the file encoding 'UTF-8'.

Documentation



Link.Create - Link User Guide

dialog.github.io/link/3.0/API/Link.Create/

AppsLinkFlying & SailingDyalogCloudExercisesSBOTravelLinuxSportAPLProductivityGit

Reading list

DYALOG

Link User Guide

Search

Dyalog/Link
6 Stars · 8 Forks

Link User Guide

Overview

Introduction

Technical Details and Limitations

Workspaces

History of source files as text in Dyalog

Install and Upgrade

Installation

Upgrading to Link 3.0

Change History

Working with Link

Basic Usage

Setting Up Your Environment

Converting an Existing Workspace to use Link

Migrating from SALT to Link

API & Command Reference

API Overview

Link.Add

Link.Break

Link.CaseCode

Link.Create

Link.Export

Link.Expunge

Link.Create

Syntax

```
]LINK.Create <ns> <dir> [-source={ns|dir|auto}] [-watch={none|ns|dir|both}] [-case: message + {options} ]SE.Link.Create (namespace directory)
```

Arguments

- `namespace` is either a reference to, or a simple character containing the name of a namespace.

In the user command `<ns>` is simply the name of the namespace. If a reference is used, it must refer to a namespace which has a display form which has name class 9 and can be used to locate the namespace (as opposed to an "anonymous" space with a name containing `[namespace]` or similar segments).
- `directory` is a simple character vector containing the path to a file system directory without any trailing slash or backslash.

In the user command, `<dir>` is the path to the file system directory.

Result

- `message` is a simple character vector describing the established link, along with possible failures

Table of contents

Syntax

Arguments

Result

Common options

source

watch

arrays

sysVars

forceExtensions

forceFileNames

Advanced Options

flatten

caseCode

beforeWrite

beforeRead

getFilename

codeExtensions

customExtensions

typeExtensions

fastLoad

Introduction - Link User Guide

← → ↺ dyalog.github.io/link/3.0/

★ 🗨 ⚙ ⌵ 👤 ⋮


📁 Apps 📁 Link 📁 Flying & Sailing 📁 Dyalog 📁 Cloud 📁 Exercises 📁 SBO 📁 Travel 📁 Linux 📁 Sport 📁 APL 📁 Productivity 📁 Git

📖 Reading list

DYALOG

Link User Guide

🔍 Search

 **Dyalog/Link**
6 Stars - 8 Forks

Link User Guide

Overview

Introduction

Technical Details and Limitations

Workspaces

History of source files as text in Dyalog

Install and Upgrade

Installation

Upgrading to Link 3.0

Change History

Working with Link

Basic Usage

Setting Up Your Environment

Converting an Existing Workspace to use Link

Migrating from SALT to Link

API & Command Reference

API Overview

Link.Add

Link.Break

Link.CaseCode

Link.Create

Link.Export

Link.Expunge

Introduction

Link enables users of Dyalog to store their APL source code in text files. This is the documentation for Link Version 3.0, which will be released in the autumn of 2021 and included with the next release of Dyalog APL. If you have an earlier version of APL or Link, you might want to read one or more of the following pages before continuing:

- Link version 2.0** If you are actually looking for documentation of the version which was distributed with Dyalog APL versions 17.1 and 18.0.
- Migrating to Link 3.0 from Link 2.0:** Dyalog recommends migrating to version 3.0 at your earliest convenience.
- Migrating to Link 3.0 from SALT:** If you have APL source in text files managed by SALT that you want to migrate to Link.
- Installation instructions:** If you want to download and install Link from the GitHub repository rather than use the version installed with APL, for example if you want to use Link 3.0 with Dyalog version 18.0.
- The historical perspective:** Link is a step on a journey which begins more than a decade ago with the introduction of SALT for managing source code in text files, as an alternative to binary workspaces and files, and will hopefully end with the interpreter handling everything itself.

Audience

It is assumed the reader has a reasonable understanding of Dyalog and in particular workspaces, and namespaces.

✎

Table of contents

Audience

What is Link?

Link is NOT...

Link fundamentals

Functions vs. User Commands

User commands

API functions

Further reading

Frequently Asked Questions

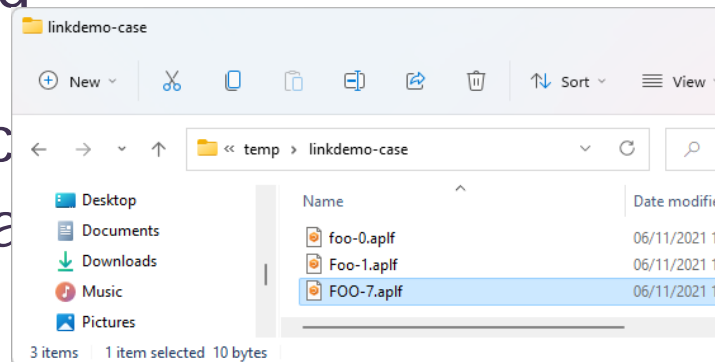
Highlights of Link 3.0

- Works with 18.0, but 18.2 recommended
 - v18.2: LINK_DIR & LINK_RUN
- MUCH more robust and widely tested than Link 2.0
- Handles "Resync" of saved workspaces
- Supports names that differ only in case using "case coding"

18.2

Highlights of Link 3.0

- Works with 18.0, but 18.2 recommended
 - v18.2: LINK_DIR & LINK_RUN
- MUCH more robust and widely tested than Link 2.0
- Handles "Resync" of saved workspace
- Supports names that differ only in case using "case coding"



Highlights of Link 3.0

- Works with 18.0, but 18.2 recommended
 - v18.2: LINK_DIR & LINK_RUN
- MUCH more robust and widely tested than Link 2.0
- Handles "Resync" of saved workspaces
- Supports names that differ only in case using "case coding"
- Too many improvements to list here

18.2

Link 3.0 Milestone

github.com/Dyalog/link/milestone/?closed=1

Apps Link Flying & Sailing Dyalog Cloud Exercises SRO Travel Linux Sport APL Productivity Git

Search or jump to... Pull requests Issues Marketplace Explore

Dyalog/link Public

Unwatch 11 Star 6 Fork 8

<> Code Issues 15 Pull requests Discussions Actions Projects Wiki Security Insights Settings

Labels Milestones Edit milestone New issue

Link 3.0

⚠ Past due by 6 days 87% complete

Companion of next Dyalog release after 18.0.

<input type="checkbox"/>	<input checked="" type="radio"/> 4 Open	<input checked="" type="radio"/> 29 Closed		
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Jlink.add should allow system variables enhancement	1
			#381 by mikromberg was closed 19 days ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Link should load system variables before all other files bug	1
			#380 by mikromberg was closed 19 days ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Changing the casing of a name causes trouble in non-case-sensitive file systems bug	1
			#369 by apiteam was closed on Oct 3	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Using Link together with "shared code files" have negative effect on performance. (Dyalog reference 16653) enhancement	1
			#47 by cenghoff was closed on Sep 30	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Link.Break can hang when "Pause Threads on Error" is enabled enhancement	5
			#343 by apiteam was closed on Sep 30	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	JLINK.Create throws spurious warning on MacOS without .NET Core bug interpreter	6
			#166 by xpqz was closed on Sep 30	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	WIBNI: Link.Create informs about newly created directories enhancement	1
			#264 by rikedyt was closed on Sep 30	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	We should issue a warning if a localised name is edited enhancement	1
			#359 by mikromberg was closed on Sep 30	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	"No source for ..." message editing local variable in a thread bug interpreter	1
			#351 by mikromberg was closed on Sep 23	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Array.Deserialise reports "Multi-line input" when there are trailing blank lines arrays bug	1
			#345 by mikromberg was closed on Sep 20	
<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	Following a JLink.Create which failed, unable to jsave because □DM references □SE bug	1
			#328 by mikromberg was closed on Sep 19	



Link 3.0 Milestone

github.com/Dyalog/link/milestone/1

Apps Link Flying & Sailing Dyalog Cloud Exercises SBO Travel Linux Sport APL Productivity Git

Reading list

Search or jump to... Pull requests Issues Marketplace Explore

Dyalog / link Public

Unwatch 11 Star 6 Fork 8

Code Issues 15 Pull requests Discussions Actions Projects Wiki Security Insights Settings

Labels Milestones




Edit milestone New issue

Link 3.0

⚠ Past due by 6 days 87% complete

Companion of next Dyalog release after 18.0.

4 Open 29 Closed

- ☐ ☒ Resolve conflict between versions of WSLoaded documentation refactor  #324 opened on Sep 2 by mikromberg
- ☐ ☒ Link 3.0 docs website can show a blank menu on the left documentation  4 #346 opened on Sep 15 by dyalog
- ☐ ☒ Document that the FSW is inappropriate as a deployment tool documentation #386 opened 15 days ago by mikromberg
- ☐ ☒ Jlink.resync fails when -flatten is set bug  1 #393 opened 21 hours ago by mikromberg

💡 Tip! You can use `shift + j` or `shift + k` to move items with your keyboard.

© 2021 GitHub, Inc. Terms Privacy Security Status Docs

Contact GitHub Pricing API Training Blog About



Dyalog/link: Code for linking a namespace to a directory

github.com/Dyalog/link

Apps Link Flying & Sailing Dyalog Cloud Exercises SBO Travel Linux Sport APL Productivity Git Reading list

Search or jump to... Pull requests Issues Marketplace Explore

Dyalog / link Public

Unwatch 11 Star 6 Fork 8

Code Issues 15 Pull requests Discussions Actions Projects Wiki Security Insights Settings

master 7 branches 1 tag Go to file Add file Code

mkromberg Update README.md bc638ac now 652 commits

.github/ISSUE_TEMPLATE	Update issue templates	2 years ago
SALT/spice	Fix #203: UCMD error when old Link found (#209)	10 months ago
StartupSession	Merge branch 'master' of https://github.com/Dyalog/link	6 days ago
docs	Update installation.md after making first release	last month
help	Update README + v2.0 doc in master to mention 3.0	2 months ago
.gitattributes	Create .gitattributes	3 years ago
.gitmodules	Added GhostRider submodule	16 months ago
LICENSE	Create LICENSE	2 years ago
README.md	Update README.md	now
mkdocs.yml	More doc tweaks	2 months ago
startup.dyalog	Implement LINK_DIR	19 days ago

README.md

Link

Link enables the use of text files as the primary storage mechanism for APL source code, and is expected to replace the use of traditional binary workspaces and component files for many new APL projects.

Two significantly different versions of Link are currently available:

Version 3.0

Version 3.0 is scheduled for distribution with the next release of Dyalog APL, and works with version 18.0. It is recommended to anyone getting started with Link and Dyalog APL. The [online documentation](#) is maintained using markdown files which can be found in `docs` folder and deployed using `mkdocs` and `mike`, hosted with GitHub Pages.

This version is actively developed in the master branch of this repository and supported via GitHub issues and via normal Dyalog support channels. Some early versions of this code still identify themselves as version 2.1.

About

Code for linking a namespace to a directory

github.com/dyalog/link/wiki

synchronization sourcecode source-code apl scm dyalog dyalog-library dyalog-user-command dyalog-apl

Readme MIT License

Releases

1 tags

Create a new release

Packages

No packages published

Publish your first package

Contributors

10

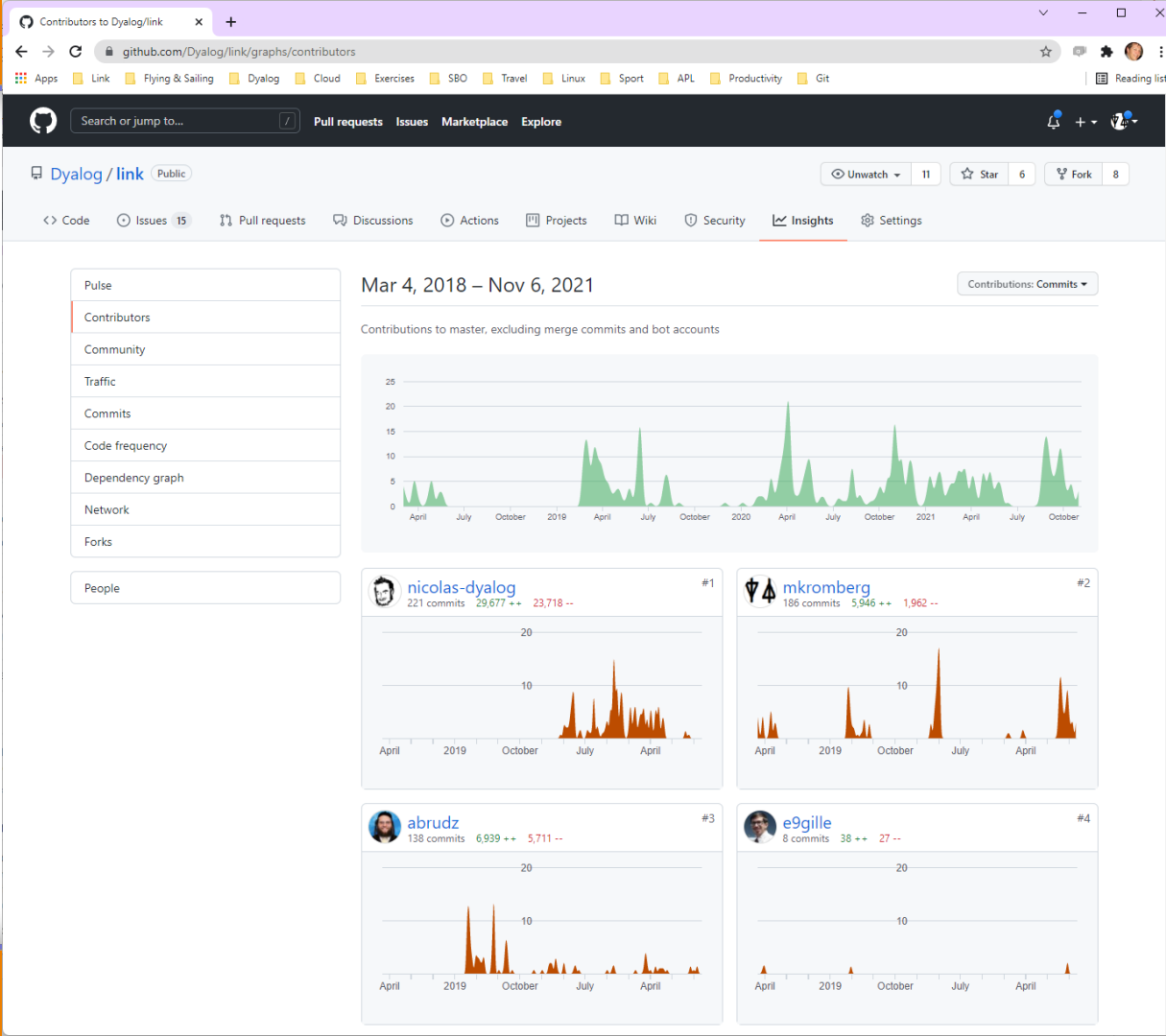
Environments

github-pages Active

Languages

APL 100.0%





Plans for Link 3.1

- Note that, since Link is on GitHub, features may be released independently of Dyalog APL
 - Link 3.1 will be included with Dyalog 19.0 but probably released earlier
- Optional "Crawler" detecting differences between workspace and directory
 - (running `Link.Resync` in the background)
 - Will detect use of `)COPY` or definitions using `⌈FX` and assignment (`dup←{ω ω}`)
- Think again about cross-platform file system watcher and Linux/macOS desktop integration



Link 3.1 Milestone

github.com/Dyalog/link/milestone/2

Apps Link Flying & Sailing Dyalog Cloud Exercises SBO Travel Linux Sport APL Productivity Git

Search or jump to... Pull requests Issues Marketplace Explore

Dyalog / link Public

Unwatch 11 Star 6 Fork 8

Code Issues 15 Pull requests Discussions Actions Projects Wiki Security Insights Settings

Labels Milestones Edit milestone New issue

Link 3.1

Due by March 31, 2022 0% complete

Spring 2022 release

☐ 7 Open ☒ 0 Closed

- ☐ ☒ Create a more regular API with clear return codes and error messages enhancement
#125 opened on May 14, 2020 by nicolas-dyalog
- ☐ ☒ Would like to be able to use link to connect a scripted namespace to a file in the next major Link revamp enhancement 1
#245 opened on Mar 18 by dyavc
- ☐ ☒ Link should support storing multi-line character data in "normal" text files enhancement
#210 opened on Jan 22 by mkromberg
- ☐ ☒ Add a file crawler in background as backup for FileSystemWatcher enhancement 3
#88 opened on Jul 30, 2019 by mkromberg
- ☐ ☒ Add a namespace crawler background process to check for APL names that have been changed by APL primitives enhancement 3
#51 opened on Apr 30, 2019 by stamper
- ☐ ☒ Link.Test should use new extensions rather than '.dyalog' refactor
#370 opened on Oct 3 by mkromberg
- ☐ ☒ Use Link to load other things into ☐SE enhancement
#377 opened 29 days ago by abrudz



Link 3.1

Due by March 31, 2022 0% complete

Spring 2022 release

☐ 7 Open ✓ 0 Closed

- ☐ Create a more regular API with clear return codes and error messages enhancement
#125 opened on May 14, 2020 by nicolas-dyalog
- ☐ Would like to be able to use link to connect a scripted namespace to a file in the next major Link revamp enhancement 1
#245 opened on Mar 18 by dyavc
- ☐ Link should support storing multi-line character data in "normal" text files enhancement
#210 opened on Jan 22 by mkromberg
- ☐ Add a file crawler in background as backup for FileSystemWatcher enhancement 3
#88 opened on Jul 30, 2019 by mkromberg
- ☐ Add a namespace crawler background process to check for APL names that have been changed by APL primitives enhancement 3
#51 opened on Apr 30, 2019 by stamper
- ☐ Link.Test should use new extensions rather than '.dyalog' refactor
#370 opened on Oct 3 by mkromberg
- ☐ Use Link to load other things into SE enhancement
#377 opened 29 days ago by abrudz

Next Steps

Act Now!

- Watch Josh David on Dado
 - <https://github.com/the-carlisle-group/dado>
- Watch Brian Becker on Tatin
 - <https://github.com/aplteam/tatin>
- Take a look at
 - Acre <https://github.com/the-carlisle-group/Acre-Desktop>
 - Cider <https://github.com/aplteam/cider>
- Read the docs & start playing