Mainz, April 30th 2024

Migrating to Dyalog

Morten Kromberg, CTO

A New Wave of Migrants

- Dyalog APL was created by Dyadic Systems Ltd in 1981
  - To replace mainframe APL systems
- Almost all users of Dyalog APL are migrants
  - from SHARP APL, IBM APL2, APL+Win, APLX, or DEC APLSF, or ...
- Waves of migrants
  - Death of mainframes and minicomputers (1980's)
  - Superior support for Windows GUI (1990's)
  - Now, "the cloud" (& a few more mainframes being shut down)

Why migrate to Dyalog?

- Dyalog re-invests 90-95% of revenues in
  - Enhancing APL core technology
  - Creating tools for APL developers
  - Marketing APL outside the current APL community
- Combined revenues of products and services based on Dyalog APL exceeds $1Bn per year – and is growing

Selected Features added 2006-2024

- Web Server and Web Service Frameworks
- Run APL as a Windows Service
- Public Docker Containers
- Remote IDE for debugging service processes
- Health Monitor for monitoring multiple processes
- Parallel and Asynchronous Execution
- New Data Types:
  - 128-bit Decimal Floating Point
  - Complex Numbers
- Functional Programming (dfns)
- New primitives: Key, Stencil, Where, ...
- Significant steps towards an APL compiler
- Many speed-ups of interpreter algorithms

- Object Orientation
- Microsoft.Net Integration
- HTMLRenderer object embeds Chromium Web Browser engine
- 64-bit: *NO* workspace or component file size limits
- Unicode Support, APL Source in Text Files
- Secure TCP Sockets w/ IPv6 Support
- Encryption Toolkit
- Regular Expressions (PCRE) built-in to APL
- XML and JSON parsers for fast conversion to (and from) APL structures

Most features identical across all platforms

Why Migrate to Dyalog?

Developer tools are free, cross-platform and mostly open source:

Name	Description
SQAPL	ODBC Interface (also ADO and ADO.NET)
Jarvis	HTTP/JSON and REST service framework
HttpCommand	HTTP client
SAWS	SOAP service framework
Conga	TCP and UDP layer
SharpPlot	Business and Technical graphics
⎕XML, ⎕JSON, ⎕CSV	Built in to interpreter

Name	Description
RConnect	Interface to R
MiServer / DUI	Web Application Framework
Docker Containers	Published examples
Link	Interface to source code management
APLProcess	And isolates

Why Migrate to Dyalog?

Emerging Tools

Name	Description
Cider	Project Management
Tatin	Package Manager
NuGet	Interface to .NET Packages
Selenium	Automated GUI testing
Jupyter	Jupyter notebooks containing APL
eWC	JavaScript emulation of Win32 GUI
Arrow & Parquet	Data Science data formats

Separately Licensed Tools

Name	Description
DFS	Dyalog File Server ("SHAREFILE")
Static Analysis	Static Analysis of APL Code (code linting and vulnerability detection) Planned for 2025

Why Migrate to Dyalog?

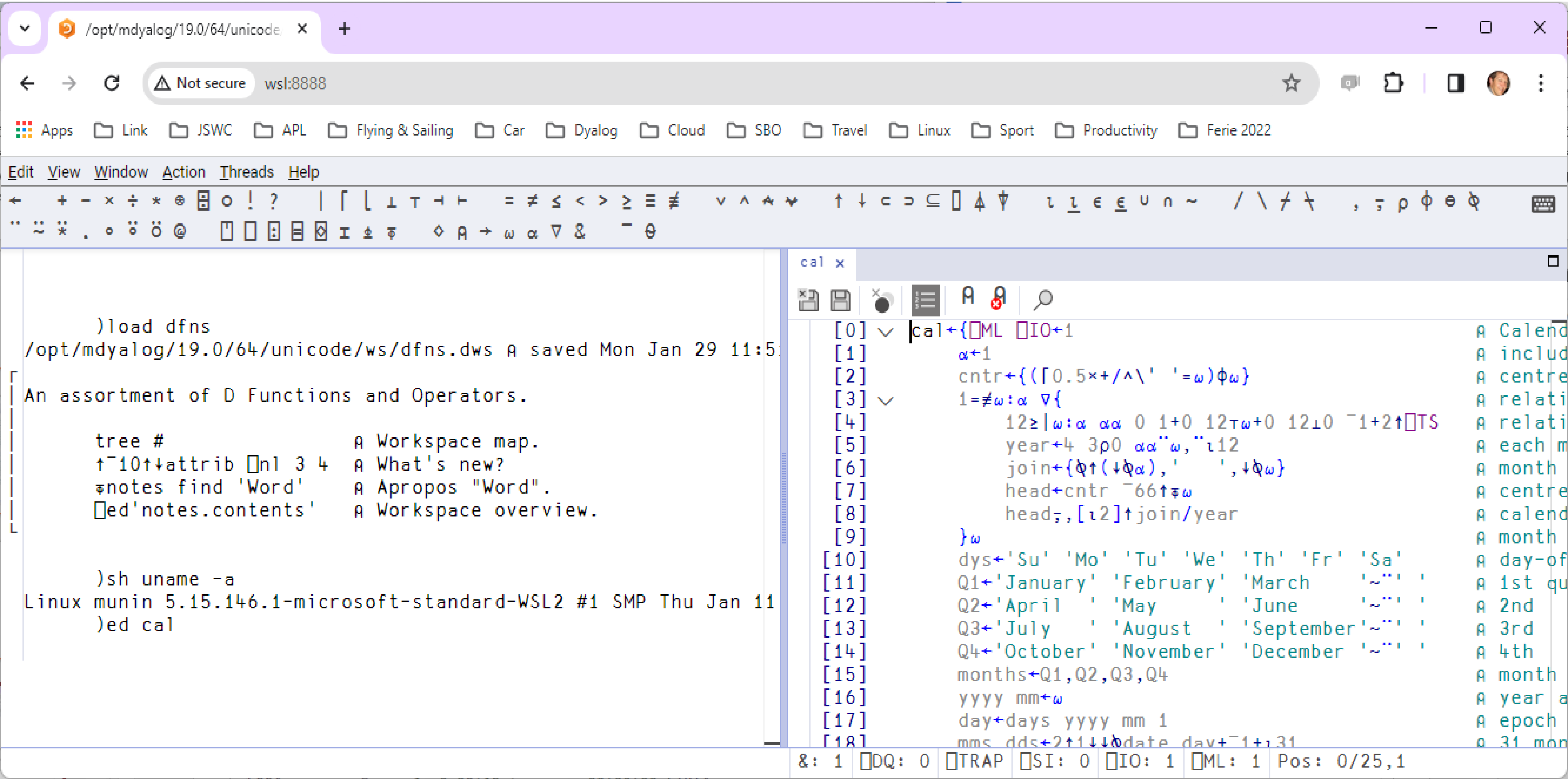
Dyalog is 100% Cross Platform

- Born under UNIX (Solaris, AIX, ...)
  - Ported to DOS, Windows, Linux (ARM, Intel), MacOS (Intel, Mx)
- Single source for all platforms
  - Workspaces and component files compatible across all platforms
- All tools are tested on all platforms
  - Exceptions where O/S does not provide a feature
  - .NET not under AIX, many Windows-only features like DDE, COM/OLE

Remote IDE (RIDE)

- Connect to and debug Dyalog APL running on any platform
- From Windows, Linux or MacOS
- Or indeed a browser running anywhere...
  - Interpreter can serve up RIDE as a web app

RIDE running in a browser



```
      )load dfns
/opt/mdyalog/19.0/64/unicode/ws/dfns.dws ⍝ saved Mon Jan 29 11:5

An assortment of D Functions and Operators.

      tree #                  ⍝ Workspace map.
      ↑¯10↑↓attrib ⎕nl 3 4    ⍝ What's new?
      ⍎notes find 'Word'      ⍝ Apropos "Word".
      ⎕ed'notes.contents'     ⍝ Workspace overview.


      )sh uname -a
Linux munin 5.15.146.1-microsoft-standard-WSL2 #1 SMP Thu Jan 11
      )ed cal
```

```
cal←{⎕ML ⎕IO←1                                                ⍝ Calend
    ⍺←1                                                       ⍝ includ
    cntr←{(⌈0.5×+/∧\' '=⍵)⌽⍵}                                  ⍝ centre
    1=≢⍵:⍺ ∇{                                                 ⍝ relati
        12≥|⍵:⍺ ⍺⍺ 0 1+0 12⊤⍵+0 12⊥0 ¯1+2↑⎕TS                  ⍝ relati
        year←4 3⍴0 ⍺⍺¨⍵,¨⍳12                                  ⍝ each m
        join←{⍉↑(↓⍉⍺),'    ',↓⍉⍵}                              ⍝ month
        head←cntr ¯66↑⍕⍵                                      ⍝ centre
        head⍪,[⍳2]↑join/year                                  ⍝ calend
    }⍵                                                        ⍝ month
    dys←'Su' 'Mo' 'Tu' 'We' 'Th' 'Fr' 'Sa'                    ⍝ day-of
    Q1←'January' 'February' 'March    '~¨' '                  ⍝ 1st qu
    Q2←'April  ' 'May     ' 'June     '~¨' '                  ⍝ 2nd
    Q3←'July   ' 'August  ' 'September'~¨' '                  ⍝ 3rd
    Q4←'October' 'November' 'December '~¨' '                  ⍝ 4th
    months←Q1,Q2,Q3,Q4                                        ⍝ month
    yyyy mm←⍵                                                 ⍝ year a
    day←days yyyy mm 1                                        ⍝ epoch
    mms dds←2↑1↓↓⍉date day+¯1+⍳31                             ⍝ 31 mon
```

```
&: 1  ⎕DQ: 0  ⎕TRAP  ⎕SI: 0  ⎕IO: 1  ⎕ML: 1  Pos: 0/25,1
```

Why Migrate to Dyalog?

Dyalog is "Cloud Ready"

- ARM and Intel Linux versions
- Public Docker containers
- Remote IDE
- Text-based source supports "Continuous Integration"
  - Build & deploy containers on commit or push
- User community starting to gain significant experience
- Working on tools to port Windows GUI to HTML/JS

Why Migrate to Dyalog?

- Dyalog APL is fast!
- Core algorithms regularly updated to take advantage of new hardware and new theory
- Research into a compiler continues

Why Migrate to Dyalog?

Dyalog APL is carefully designed to last. For example:

- Dyalog APL is tightly integrated with .NET
  - ... and still supports the old .NET Framework
  - However, Dyalog APL does not and WILL NOT depend on .NET
  - It also runs under IBM AIX, where .NET does not exist

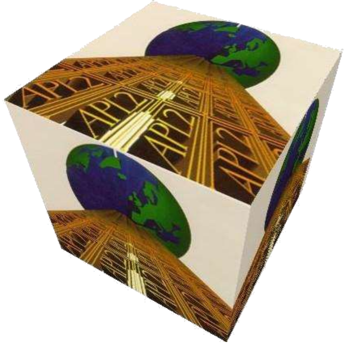
- Dyalog *will* remain very portable and independent of "temporary" frameworks

The Real Reason to Pick Dyalog APL

HOW to Migrate to Dyalog APL

1. From IBM / Logon APL2
2. From APL+Win or MicroAPL APLX

From APL2



Relatively straightforward

- A few language differences
- User Interfaces and file I/O are usually handled by simple cover-functions and possible to emulate automatically
- Linux or Windows apps may be making external calls which will require "tweaking"
- Considering implementing "format by example" but so far it has not been necessary

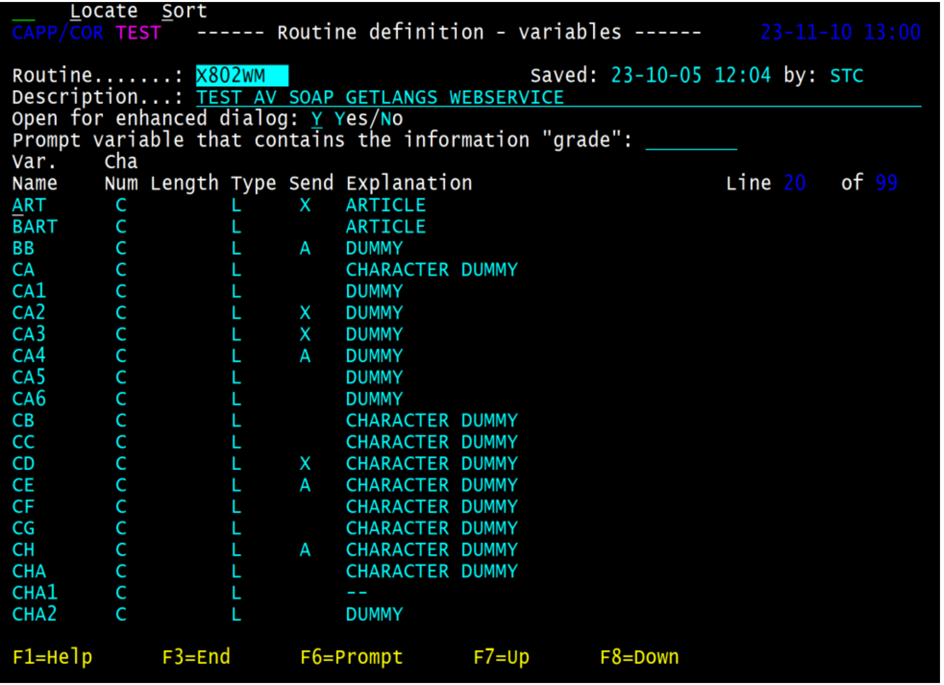
```
'555-5555'⍕nums
```

  - Easy to model in APL if necessary

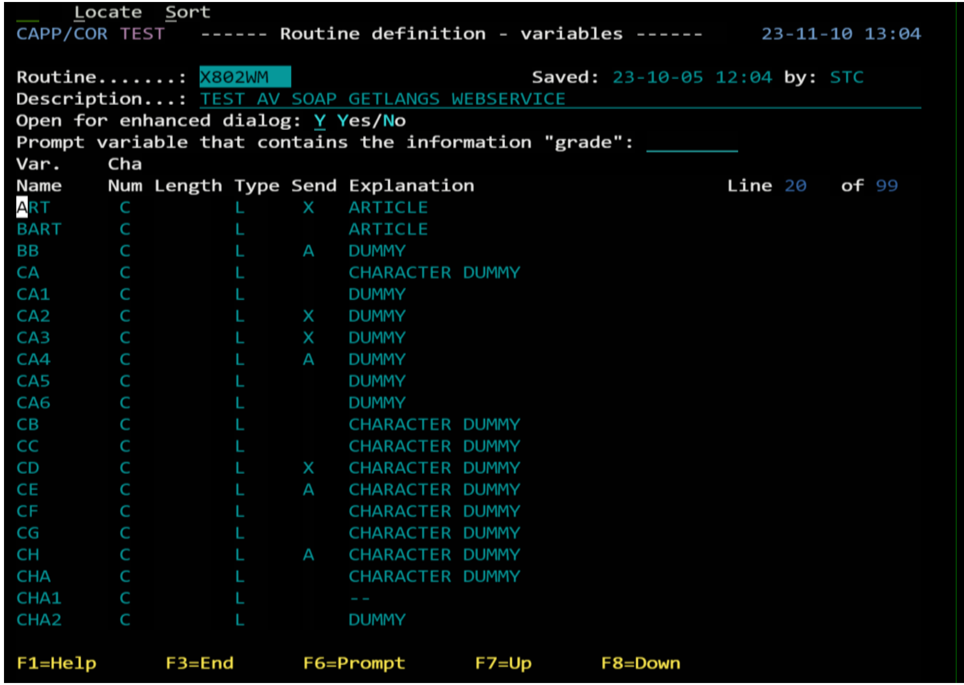
Recent / Active APL2 Migrations

- Insurance company
  - No UI, manipulates text and Excel files
  - Handled by European Consulting Partner
- Sandvik (Sweden) – in progress: Mainframe APL2 direct to Docker Containers and HTML/svg
  - Handled by Tiamatica in Malmö (Gilgamesh Athoraya)
- BIG Jewellers: Windows
  - Handled by customer "with a little help"
- Two more under discussion
  - (Germany, Canada)

Migrated APL2 Mainframe UI

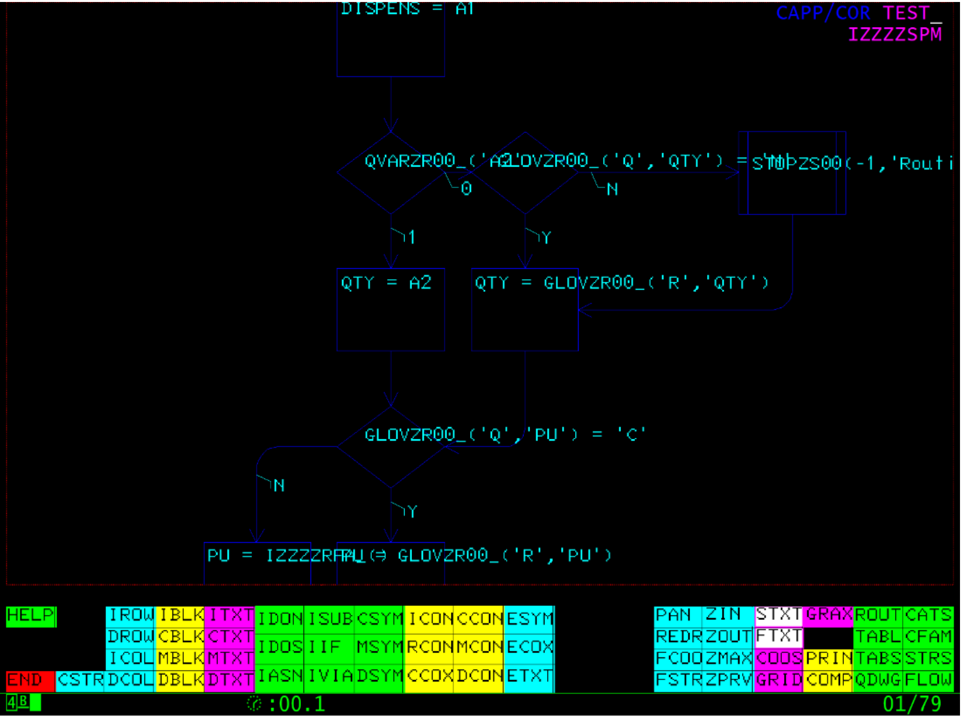


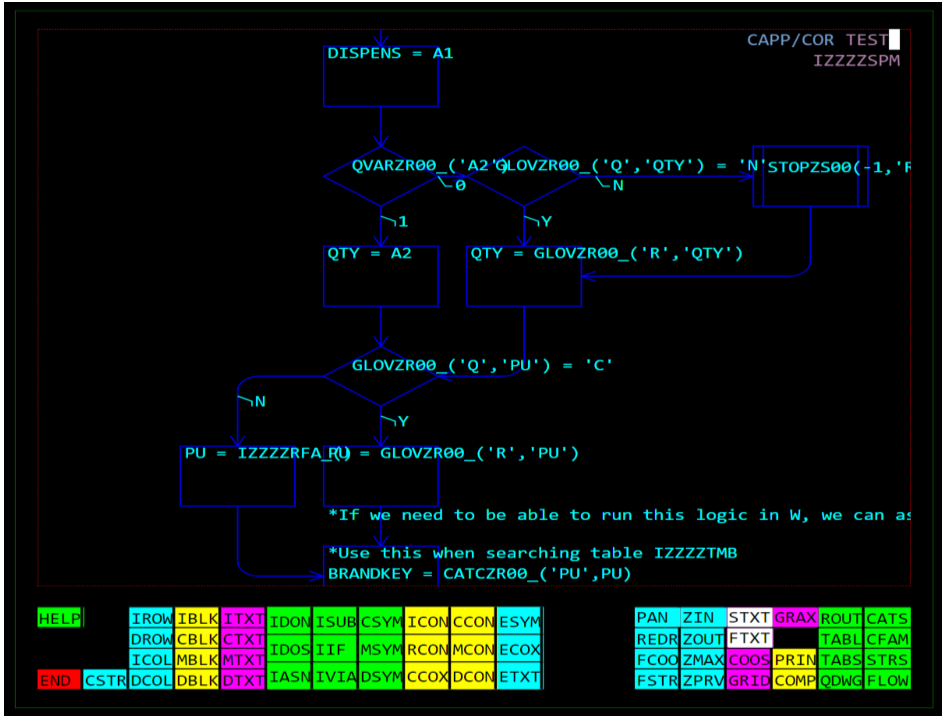
```
___    Locate  Sort
CAPP/COR TEST   ------ Routine definition - variables ------      23-11-10 13:00

Routine.......: X802WM                          Saved: 23-10-05 12:04 by: STC
Description...: TEST AV SOAP GETLANGS WEBSERVICE
Open for enhanced dialog: Y Yes/No
Prompt variable that contains the information "grade": ________
Var.     Cha
Name     Num Length Type Send Explanation                        Line 20    of 99
ART       C          L    X   ARTICLE
BART      C          L        ARTICLE
BB        C          L    A   DUMMY
CA        C          L        CHARACTER DUMMY
CA1       C          L        DUMMY
CA2       C          L    X   DUMMY
CA3       C          L    X   DUMMY
CA4       C          L    A   DUMMY
CA5       C          L        DUMMY
CA6       C          L        DUMMY
CB        C          L        CHARACTER DUMMY
CC        C          L        CHARACTER DUMMY
CD        C          L    X   CHARACTER DUMMY
CE        C          L    A   CHARACTER DUMMY
CF        C          L        CHARACTER DUMMY
CG        C          L        CHARACTER DUMMY
CH        C          L    A   CHARACTER DUMMY
CHA       C          L        CHARACTER DUMMY
CHA1      C          L        --
CHA2      C          L        DUMMY

F1=Help      F3=End      F6=Prompt      F7=Up      F8=Down
```



```
___    Locate  Sort
CAPP/COR TEST   ------ Routine definition - variables ------      23-11-10 13:04

Routine.......: X802WM                          Saved: 23-10-05 12:04 by: STC
Description...: TEST AV SOAP GETLANGS WEBSERVICE
Open for enhanced dialog: Y Yes/No
Prompt variable that contains the information "grade": ________
Var.     Cha
Name     Num Length Type Send Explanation                        Line 20    of 99
ART       C          L    X   ARTICLE
BART      C          L        ARTICLE
BB        C          L    A   DUMMY
CA        C          L        CHARACTER DUMMY
CA1       C          L        DUMMY
CA2       C          L    X   DUMMY
CA3       C          L    X   DUMMY
CA4       C          L    A   DUMMY
CA5       C          L        DUMMY
CA6       C          L        DUMMY
CB        C          L        CHARACTER DUMMY
CC        C          L        CHARACTER DUMMY
CD        C          L    X   CHARACTER DUMMY
CE        C          L    A   CHARACTER DUMMY
CF        C          L        CHARACTER DUMMY
CG        C          L        CHARACTER DUMMY
CH        C          L    A   CHARACTER DUMMY
CHA       C          L        CHARACTER DUMMY
CHA1      C          L        --
CHA2      C          L        DUMMY

F1=Help      F3=End      F6=Prompt      F7=Up      F8=Down
```

Migrated APL2 Mainframe UI


DISPENS = A1
CAPP/COR TEST_
IZZZZSPM
QVARZR00_('A2'GLOVZR00_('Q','QTY') = 'N' STOPZS00(-1,'Routi
0
N
1
Y
QTY = A2
QTY = GLOVZR00_('R','QTY')
GLOVZR00_('Q','PU') = 'C'
N
Y
PU = IZZZZRFA_PU = GLOVZR00_('R','PU')
HELP
END
IROW IBLK ITXT IDON ISUB CSYM ICON CCON ESYM
DROW CBLK CTXT IDOS IIF MSYM RCON MCON ECOX
ICOL MBLK MTXT
CSTR DCOL DBLK DTXT IASN IVIA DSYM CCOX DCON ETXT
PAN ZIN STXT GRAX ROUT CATS
REDR ZOUT FTXT TABL CFAM
FCOO ZMAX COOS PRIN TABS STRS
FSTR ZPRV GRID COMP QDWG FLOW
4B
:00.1
01/79


CAPP/COR TEST
IZZZZSPM
DISPENS = A1
QVARZR00_('A2'GLOVZR00_('Q','QTY') = 'N'STOPZS00(-1,'R
0
N
1
Y
QTY = A2
QTY = GLOVZR00_('R','QTY')
GLOVZR00_('Q','PU') = 'C'
N
Y
PU = IZZZZRFA_PU) = GLOVZR00_('R','PU')
BRANDKEY = CATCZR00_('PU',PU)
HELP
END
IROW IBLK ITXT IDON ISUB CSYM ICON CCON ESYM
DROW CBLK CTXT IDOS IIF MSYM RCON MCON ECOX
ICOL MBLK MTXT
CSTR DCOL DBLK DTXT IASN IVIA DSYM CCOX DCON ETXT
PAN ZIN STXT GRAX ROUT CATS
REDR ZOUT FTXT TABL CFAM
FCOO ZMAX COOS PRIN TABS STRS
FSTR ZPRV GRID COMP QDWG FLOW

```
*If we need to be able to run this logic in W, we can as
*Use this when searching table IZZZZTMB
```

From APL+Win or MicroAPL APLX

Same language differences as APL2, plus:

- Many system functions & control structures not found in Dyalog APL
- Double quotes (`"Don't do this!"`)
- Advanced Graphical User Interfaces
- Calls to external libraries

APLX Migrations

- MicroAPL stopped developing APLX in 2016
  - Dyalog hosts a download of the last free version
- Dyalog developed migration tools in 2016

Recent / Active APL+Win Migrations

- Two European Insurance companies
  - One with GUI, completely rewritten in Dyalog APL, the other a pure service converted to Jarvis in Linux containers
  - Handled by a European consulting partner
- METSIM® - in progress
  - Dyalog engaged to perform this migration
  - Will be used to develop tools to automate migration, including the Graphical User Interface
- Met one more potential migrant last week

Differences which are "easy"

/ is sometimes a function in Dyalog APL

```
1 0 1/¨'ABC' 'DEF' 'GHI'
```

```
'AC' 'DF' 'GI'        vs
'ABC' 'GHI'
```

Translate /¨ to ∘/¨

```
←Y
```

Not supported in Dyalog

Translate ← to {}

Other "Easy" Differences

⎕XLIB

System function not in Dyalog

```
R←∆XLIB X
X,←'*'↓⍨≢X
:If 0∊ρR←↑⊃⎕NINFO⍠1⊢X
    :If ∨/'?*'∊X
        R←0 0ρ' '
    :Else
        'XFHOST ERROR FindFirstFile 1 0 3 The system cannot find the path specified.'
                                                                              ⎕SIGNAL 22
    :EndIf
:Else
    R←R[⍋R;]
:EndIf
```

Tricky Differences

```
A B[I]
```

```
A (B[I])  or
(A B)[I]  ?
```

`f.g` when `f` or `g`
are not scalar functions

Detect and rewrite

Fortunately, these are very rare in practice

```
:LeaveIf
```

Enhance Interpreter

The Hard Parts

- Component Files
- User Interfaces (especially Graphical)
- Other I/O (e.g. SQL Databases)
- External Library Calls

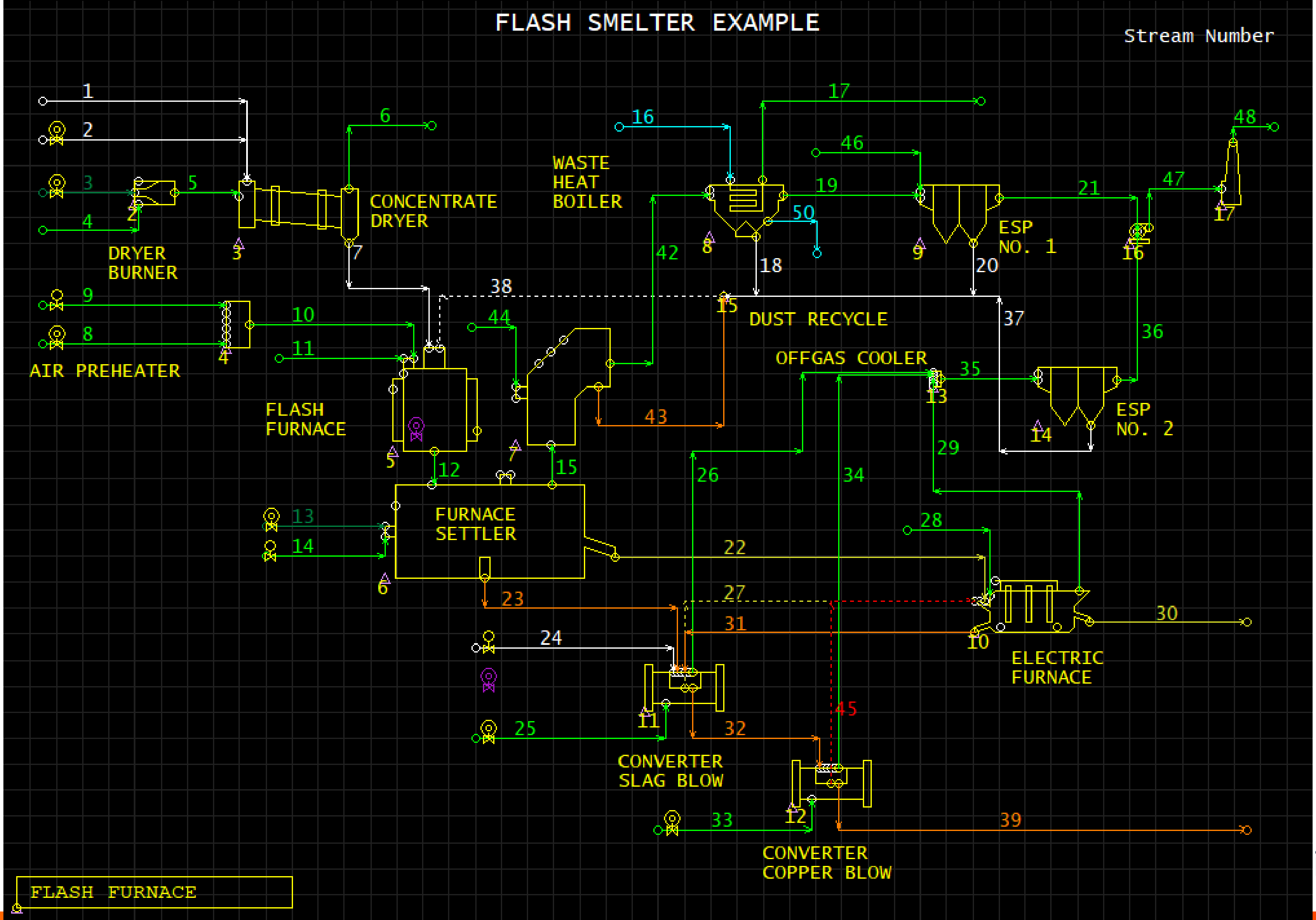
Component Files

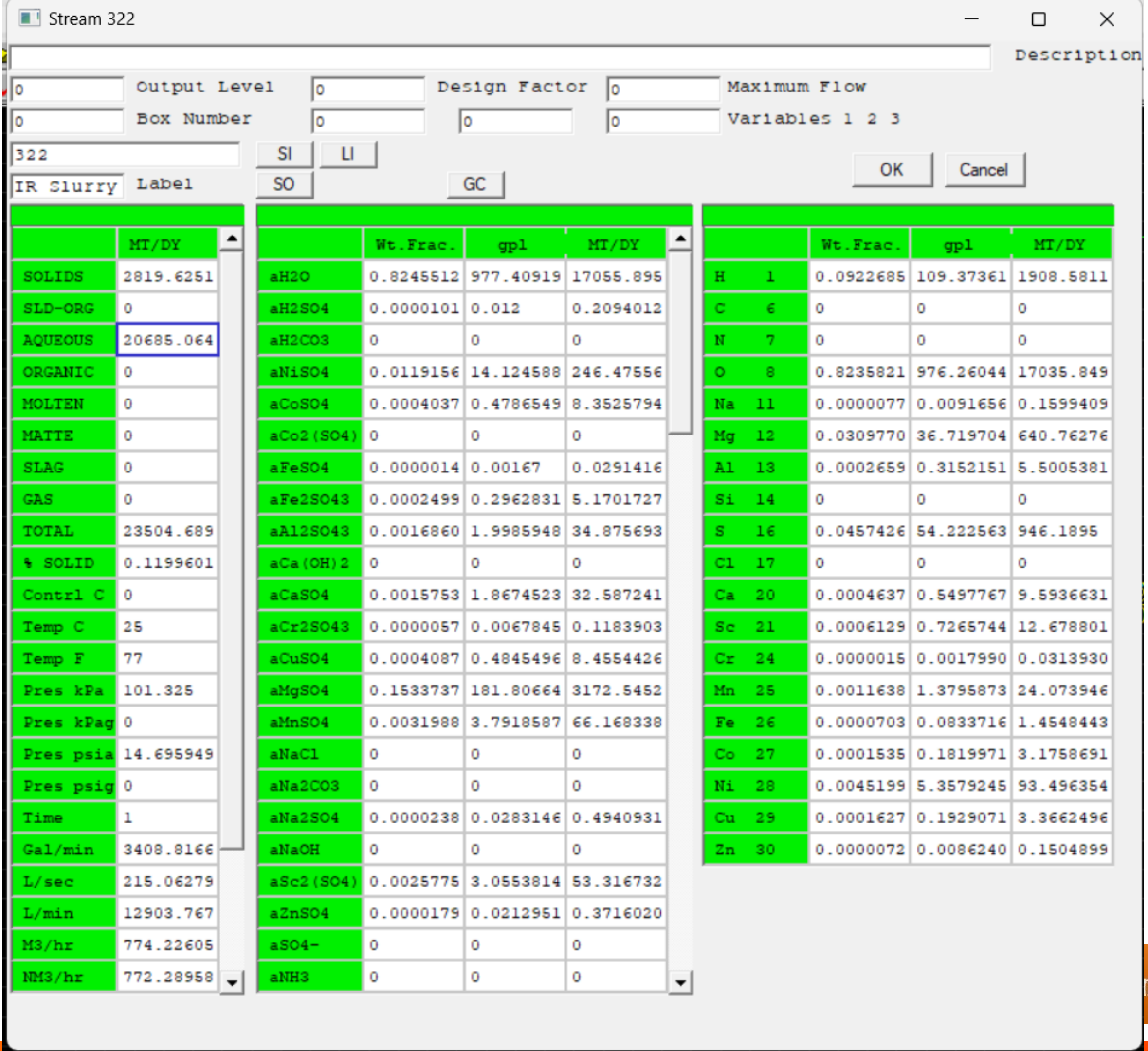
- We will develop an APL+Win COM server to read/write APL+Win component files directly from Dyalog APL
- Avoid the need for "big bang" data migrations
  - Component files can be migrated over time

The Elephant in the Room: ⎕WI ?

Dyalog is building an emulator to support the METSIM® migration

- Our goal is that no significant changes to application code will be required
- METSIM® screen shots follow
  - Many thanks to Alex Holtzapple, CEO of MSI


FLASH SMELTER EXAMPLE
Stream Number
DRYER BURNER
CONCENTRATE DRYER
WASTE HEAT BOILER
ESP NO. 1
DUST RECYCLE
AIR PREHEATER
FLASH FURNACE
OFFGAS COOLER
ESP NO. 2
FURNACE SETTLER
ELECTRIC FURNACE
CONVERTER SLAG BLOW
CONVERTER COPPER BLOW
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
42
43
44
45
46
47
48
50
FLASH FURNACE



Stream 322

Description

0 Output Level 0 Design Factor 0 Maximum Flow

0 Box Number 0 0 0 Variables 1 2 3

322

IR Slurry Label

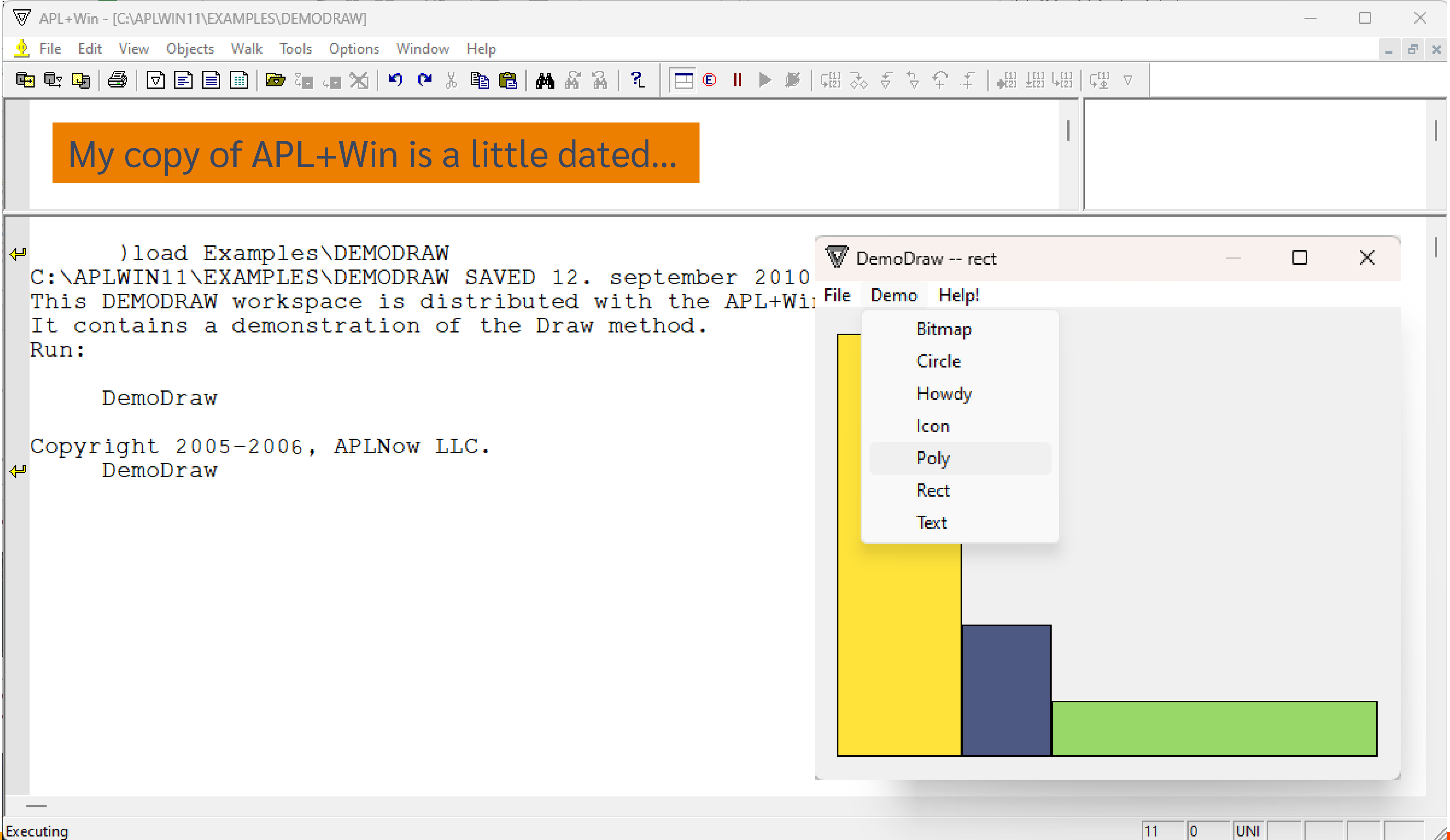
	MT/DY
SOLIDS	2819.6251
SLD-ORG	0
AQUEOUS	20685.064
ORGANIC	0
MOLTEN	0
MATTE	0
SLAG	0
GAS	0
TOTAL	23504.689
% SOLID	0.1199601
Contrl C	0
Temp C	25
Temp F	77
Pres kPa	101.325
Pres kPag	0
Pres psia	14.695949
Pres psig	0
Time	1
Gal/min	3408.8166
L/sec	215.06279
L/min	12903.767
M3/hr	774.22605
NM3/hr	772.28958

	Wt.Frac.	gpl	MT/DY
aH2O	0.8245512	977.40919	17055.895
aH2SO4	0.0000101	0.012	0.2094012
aH2CO3	0	0	0
aNiSO4	0.0119156	14.124588	246.47556
aCoSO4	0.0004037	0.4786549	8.3525794
aCo2(SO4)	0	0	0
aFeSO4	0.0000014	0.00167	0.0291416
aFe2SO43	0.0002499	0.2962831	5.1701727
aAl2SO43	0.0016860	1.9985948	34.875693
aCa(OH)2	0	0	0
aCaSO4	0.0015753	1.8674523	32.587241
aCr2SO43	0.0000057	0.0067845	0.1183903
aCuSO4	0.0004087	0.4845496	8.4554426
aMgSO4	0.1533737	181.80664	3172.5452
aMnSO4	0.0031988	3.7918587	66.168338
aNaCl	0	0	0
aNa2CO3	0	0	0
aNa2SO4	0.0000238	0.0283146	0.4940931
aNaOH	0	0	0
aSc2(SO4)	0.0025775	3.0553814	53.316732
aZnSO4	0.0000179	0.0212951	0.3716020
aSO4-	0	0	0
aNH3	0	0	0

	Wt.Frac.	gpl	MT/DY
H 1	0.0922685	109.37361	1908.5811
C 6	0	0	0
N 7	0	0	0
O 8	0.8235821	976.26044	17035.849
Na 11	0.0000077	0.0091656	0.1599409
Mg 12	0.0309770	36.719704	640.76276
Al 13	0.0002659	0.3152151	5.5005381
Si 14	0	0	0
S 16	0.0457426	54.222563	946.1895
Cl 17	0	0	0
Ca 20	0.0004637	0.5497767	9.5936631
Sc 21	0.0006129	0.7265744	12.678801
Cr 24	0.0000015	0.0017990	0.0313930
Mn 25	0.0011638	1.3795873	24.073946
Fe 26	0.0000703	0.0833716	1.4548443
Co 27	0.0001535	0.1819971	3.1758691
Ni 28	0.0045199	5.3579245	93.496354
Cu 29	0.0001627	0.1929071	3.3662496
Zn 30	0.0000072	0.0086240	0.1504899

Practicing

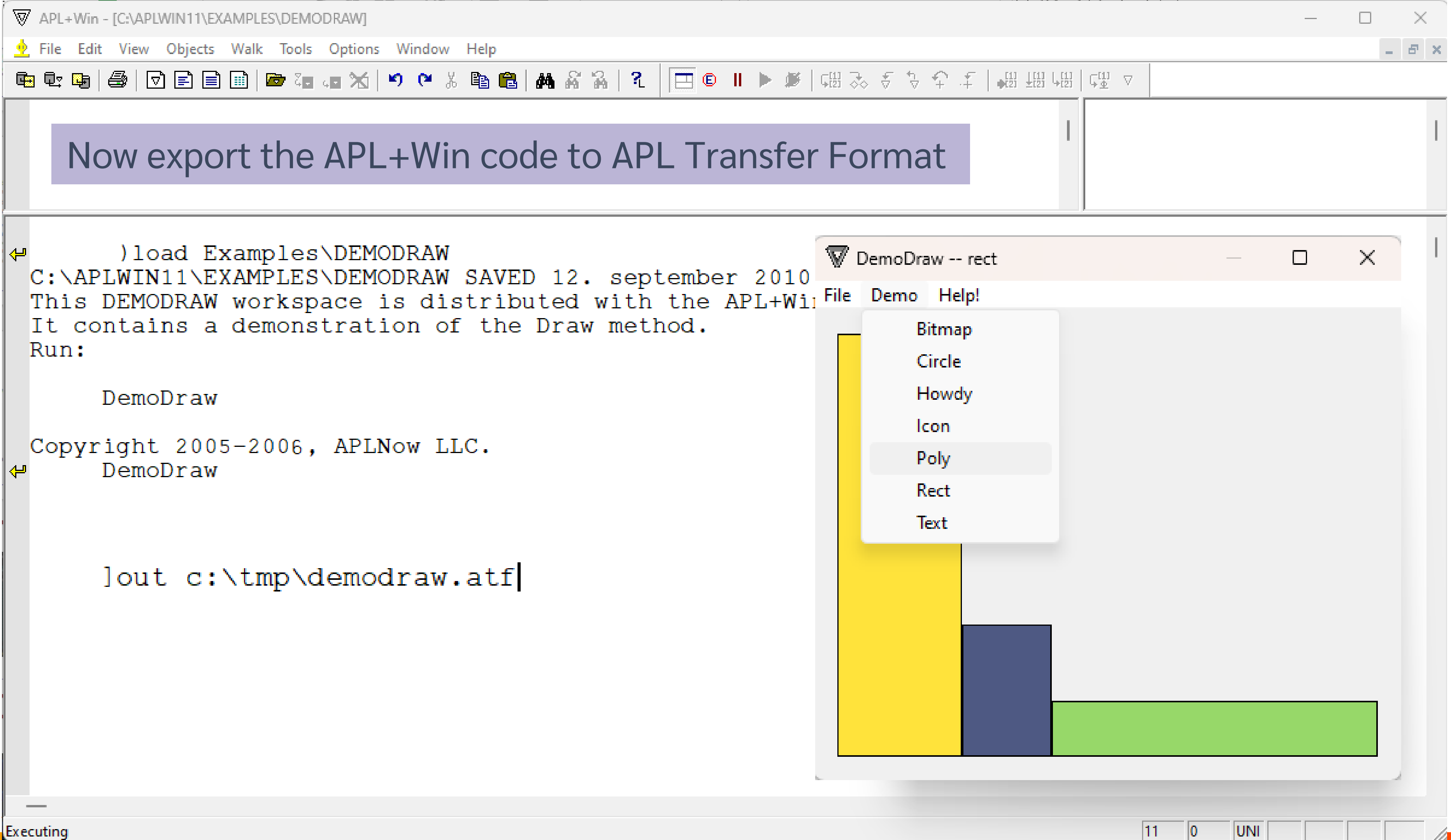
- APL+WIN comes with a handful of GUI demonstration applications



My copy of APL+Win is a little dated...

```
      )load Examples\DEMODRAW
C:\APLWIN11\EXAMPLES\DEMODRAW SAVED 12. september 2010
This DEMODRAW workspace is distributed with the APL+Wi
It contains a demonstration of the Draw method.
Run:

     DemoDraw

Copyright 2005-2006, APLNow LLC.
     DemoDraw
```

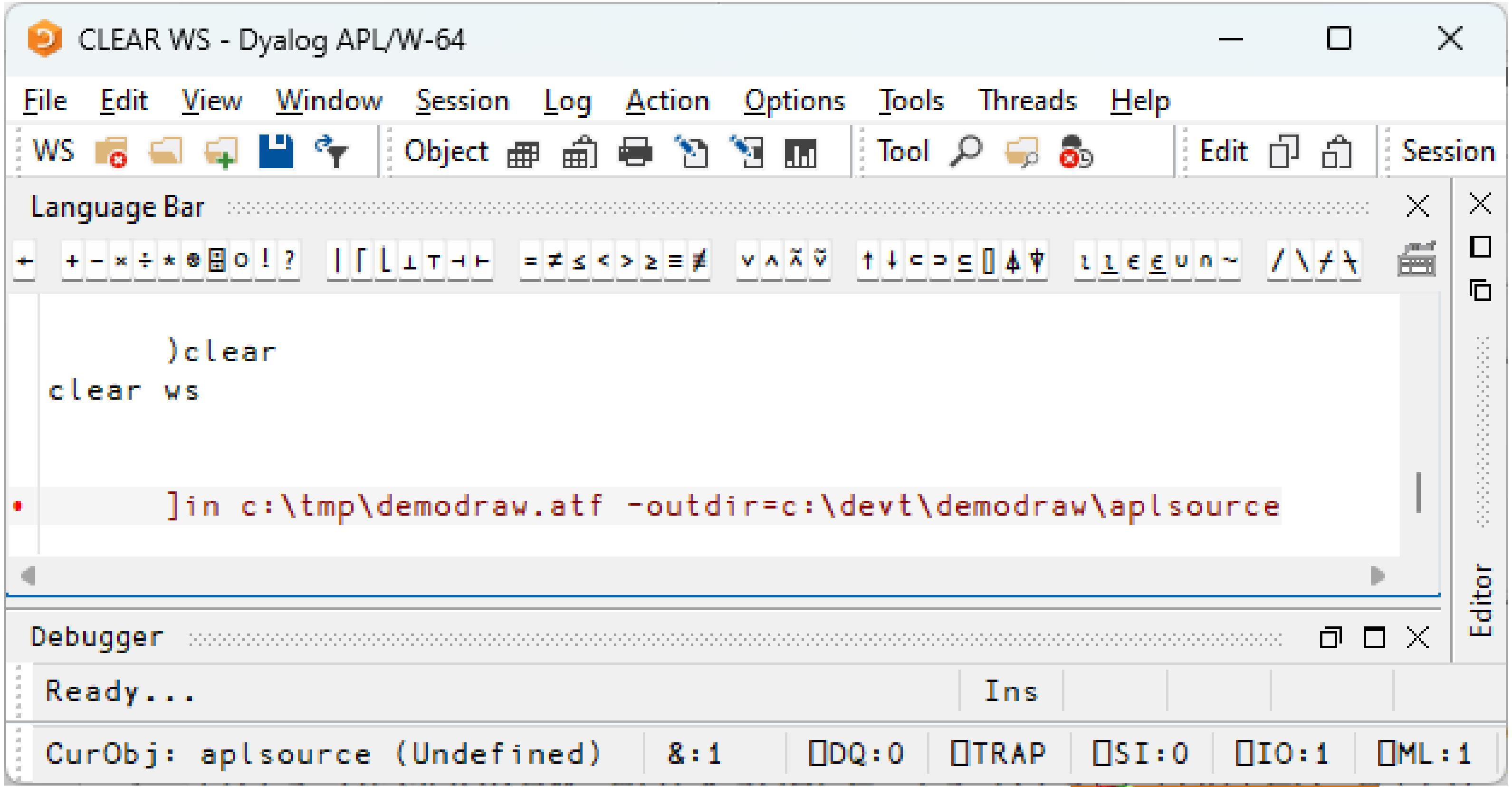
APL+Win - [C:\APLWIN11\EXAMPLES\DEMODRAW]
File Edit View Objects Walk Tools Options Window Help
DemoDraw -- rect
File Demo Help!
Bitmap
Circle
Howdy
Icon
Poly
Rect
Text
Executing
11
0
UNI



Now export the APL+Win code to APL Transfer Format

```
      )load Examples\DEMODRAW
C:\APLWIN11\EXAMPLES\DEMODRAW SAVED 12. september 2010
This DEMODRAW workspace is distributed with the APL+Wi
It contains a demonstration of the Draw method.
Run:

     DemoDraw

Copyright 2005-2006, APLNow LLC.
     DemoDraw



     ]out c:\tmp\demodraw.atf
```

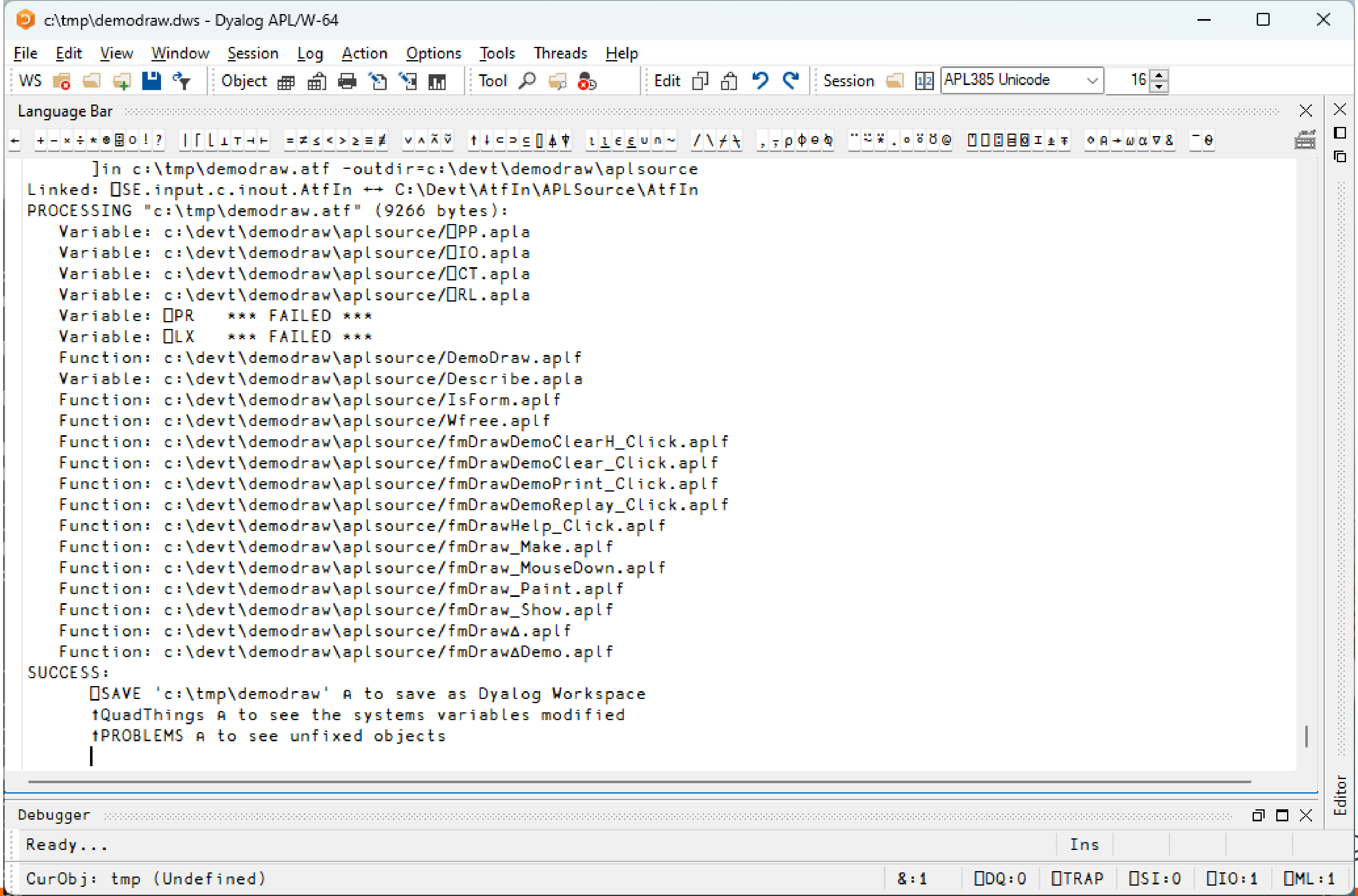
Convert the ATF file to Dyalog Source files



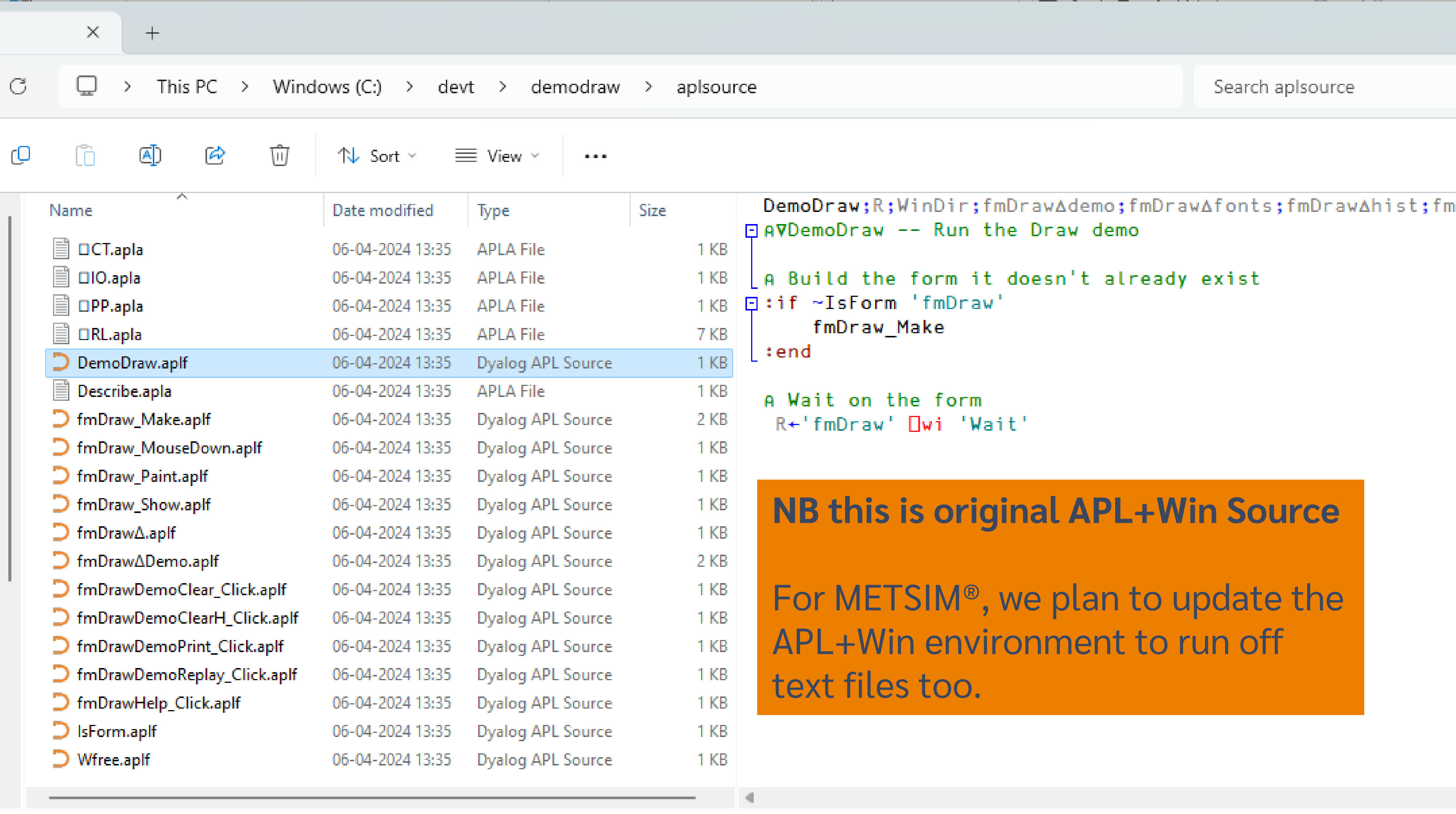
CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
Language Bar
Debugger
Editor

```
      )clear
clear ws

      ]in c:\tmp\demodraw.atf -outdir=c:\devt\demodraw\aplsource
```

```
Ready...                                  Ins
CurObj: aplsource (Undefined)   &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```


c:\tmp\demodraw.dws - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL385 Unicode 16
Language Bar
Debugger
Editor
Ready...
Ins
CurObj: tmp (Undefined)
&:1
⎕DQ:0
⎕TRAP
⎕SI:0
⎕IO:1
⎕ML:1

```
      ]in c:\tmp\demodraw.atf -outdir=c:\devt\demodraw\aplsource
Linked: ⎕SE.input.c.inout.AtfIn ←→ C:\Devt\AtfIn\APLSource\AtfIn
PROCESSING "c:\tmp\demodraw.atf" (9266 bytes):
   Variable: c:\devt\demodraw\aplsource/⎕PP.apla
   Variable: c:\devt\demodraw\aplsource/⎕IO.apla
   Variable: c:\devt\demodraw\aplsource/⎕CT.apla
   Variable: c:\devt\demodraw\aplsource/⎕RL.apla
   Variable: ⎕PR    *** FAILED ***
   Variable: ⎕LX    *** FAILED ***
   Function: c:\devt\demodraw\aplsource/DemoDraw.aplf
   Variable: c:\devt\demodraw\aplsource/Describe.apla
   Function: c:\devt\demodraw\aplsource/IsForm.aplf
   Function: c:\devt\demodraw\aplsource/Wfree.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawDemoClearH_Click.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawDemoClear_Click.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawDemoPrint_Click.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawDemoReplay_Click.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawHelp_Click.aplf
   Function: c:\devt\demodraw\aplsource/fmDraw_Make.aplf
   Function: c:\devt\demodraw\aplsource/fmDraw_MouseDown.aplf
   Function: c:\devt\demodraw\aplsource/fmDraw_Paint.aplf
   Function: c:\devt\demodraw\aplsource/fmDraw_Show.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawΔ.aplf
   Function: c:\devt\demodraw\aplsource/fmDrawΔDemo.aplf
SUCCESS:
      ⎕SAVE 'c:\tmp\demodraw' ⍝ to save as Dyalog Workspace
      ↑QuadThings ⍝ to see the systems variables modified
      ↑PROBLEMS ⍝ to see unfixed objects
```



Name	Date modified	Type	Size
⎕CT.apla	06-04-2024 13:35	APLA File	1 KB
⎕IO.apla	06-04-2024 13:35	APLA File	1 KB
⎕PP.apla	06-04-2024 13:35	APLA File	1 KB
⎕RL.apla	06-04-2024 13:35	APLA File	7 KB
DemoDraw.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
Describe.apla	06-04-2024 13:35	APLA File	1 KB
fmDraw_Make.aplf	06-04-2024 13:35	Dyalog APL Source	2 KB
fmDraw_MouseDown.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDraw_Paint.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDraw_Show.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDrawΔ.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDrawΔDemo.aplf	06-04-2024 13:35	Dyalog APL Source	2 KB
fmDrawDemoClear_Click.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDrawDemoClearH_Click.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDrawDemoPrint_Click.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDrawDemoReplay_Click.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
fmDrawHelp_Click.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
IsForm.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB
Wfree.aplf	06-04-2024 13:35	Dyalog APL Source	1 KB

```
DemoDraw;R;WinDir;fmDrawΔdemo;fmDrawΔfonts;fmDrawΔhist;fm
⍝∇DemoDraw -- Run the Draw demo

⍝ Build the form it doesn't already exist
:if ~IsForm 'fmDraw'
    fmDraw_Make
:end

⍝ Wait on the form
 R←'fmDraw' ⎕wi 'Wait'
```

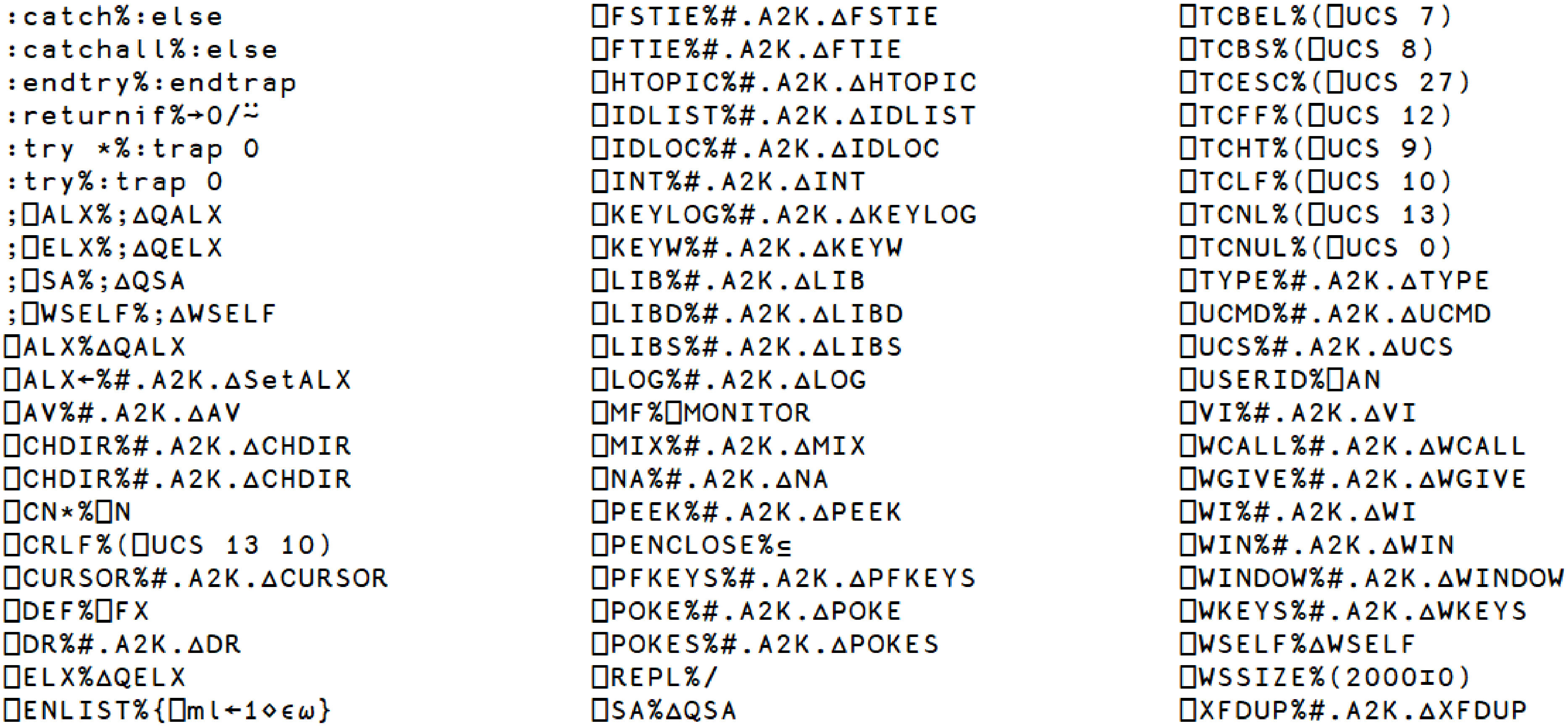
NB this is original APL+Win Source

For METSIM®, we plan to update the APL+Win environment to run off text files too.

Next, map code from APL+Win to Dyalog

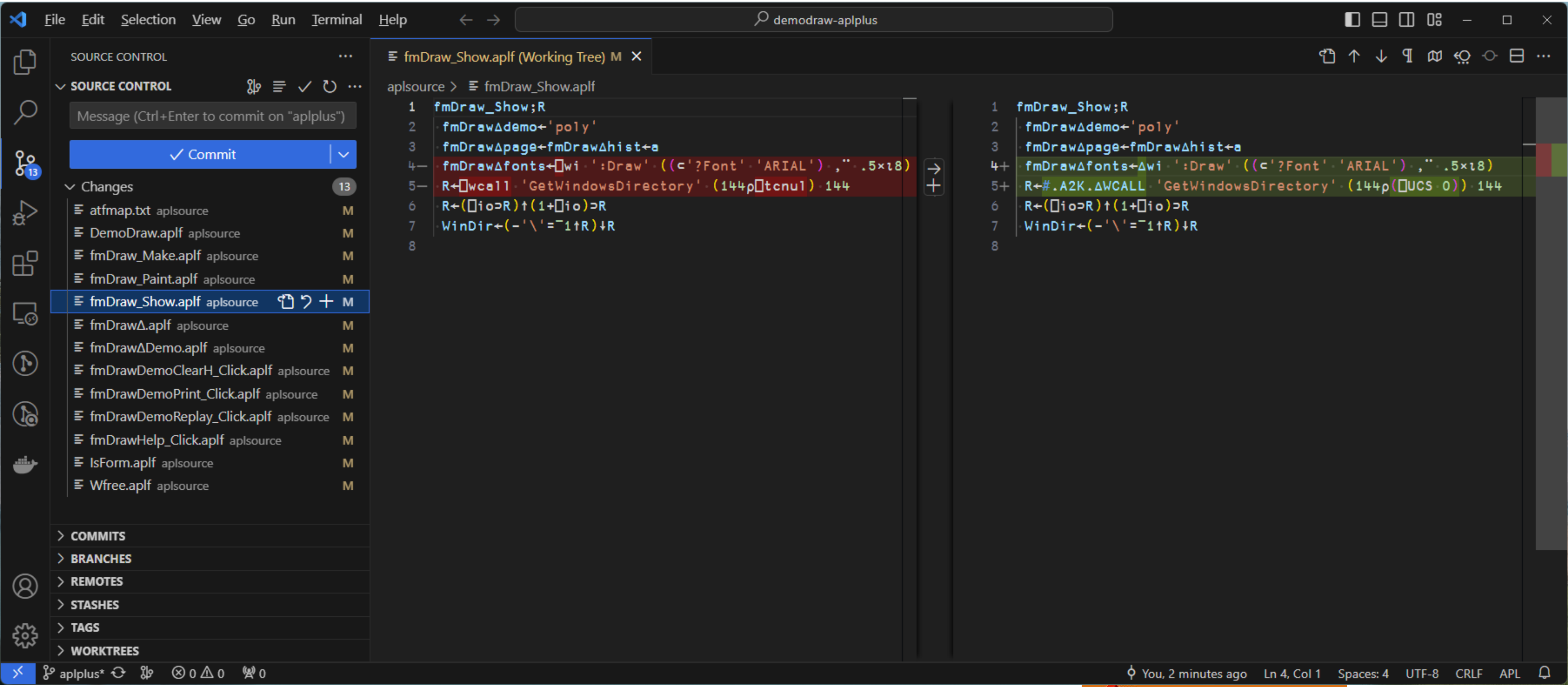
```
      ]todyalog aplsource c:\devt\demodraw\dyalog a2k
Using c:\devt\demodraw\aplsource\atfmap.txt
20 files processed
```

atfmap.txt



```
:catch%:else
:catchall%:else
:endtry%:endtrap
:returnif%→0/⍨
:try *%:trap 0
:try%:trap 0
;⎕ALX%;∆QALX
;⎕ELX%;∆QELX
;⎕SA%;∆QSA
;⎕WSELF%;∆WSELF
⎕ALX%∆QALX
⎕ALX←%#.A2K.∆SetALX
⎕AV%#.A2K.∆AV
⎕CHDIR%#.A2K.∆CHDIR
⎕CHDIR%#.A2K.∆CHDIR
⎕CN*%⎕N
⎕CRLF%(⎕UCS 13 10)
⎕CURSOR%#.A2K.∆CURSOR
⎕DEF%⎕FX
⎕DR%#.A2K.∆DR
⎕ELX%∆QELX
⎕ENLIST%{⎕ml←1◇∊⍵}
⎕FSTIE%#.A2K.∆FSTIE
⎕FTIE%#.A2K.∆FTIE
⎕HTOPIC%#.A2K.∆HTOPIC
⎕IDLIST%#.A2K.∆IDLIST
⎕IDLOC%#.A2K.∆IDLOC
⎕INT%#.A2K.∆INT
⎕KEYLOG%#.A2K.∆KEYLOG
⎕KEYW%#.A2K.∆KEYW
⎕LIB%#.A2K.∆LIB
⎕LIBD%#.A2K.∆LIBD
⎕LIBS%#.A2K.∆LIBS
⎕LOG%#.A2K.∆LOG
⎕MF%⎕MONITOR
⎕MIX%#.A2K.∆MIX
⎕NA%#.A2K.∆NA
⎕PEEK%#.A2K.∆PEEK
⎕PENCLOSE%⊆
⎕PFKEYS%#.A2K.∆PFKEYS
⎕POKE%#.A2K.∆POKE
⎕POKES%#.A2K.∆POKES
⎕REPL%/
⎕SA%∆QSA
⎕TCBEL%(⎕UCS 7)
⎕TCBS%(⎕UCS 8)
⎕TCESC%(⎕UCS 27)
⎕TCFF%(⎕UCS 12)
⎕TCHT%(⎕UCS 9)
⎕TCLF%(⎕UCS 10)
⎕TCNL%(⎕UCS 13)
⎕TCNUL%(⎕UCS 0)
⎕TYPE%#.A2K.∆TYPE
⎕UCMD%#.A2K.∆UCMD
⎕UCS%#.A2K.∆UCS
⎕USERID%⎕AN
⎕VI%#.A2K.∆VI
⎕WCALL%#.A2K.∆WCALL
⎕WGIVE%#.A2K.∆WGIVE
⎕WI%#.A2K.∆WI
⎕WIN%#.A2K.∆WIN
⎕WINDOW%#.A2K.∆WINDOW
⎕WKEYS%#.A2K.∆WKEYS
⎕WSELF%∆WSELF
⎕WSSIZE%(2000⌶0)
⎕XFDUP%#.A2K.∆XFDUP
```

Take advantage of Git and VS Code


File Edit Selection View Go Run Terminal Help
demodraw-aplplus
SOURCE CONTROL
SOURCE CONTROL
Message (Ctrl+Enter to commit on "aplplus")
✓ Commit
Changes 13

- atfmap.txt aplsource M
- DemoDraw.aplf aplsource M
- fmDraw_Make.aplf aplsource M
- fmDraw_Paint.aplf aplsource M
- fmDraw_Show.aplf aplsource M
- fmDrawΔ.aplf aplsource M
- fmDrawΔDemo.aplf aplsource M
- fmDrawDemoClearH_Click.aplf aplsource M
- fmDrawDemoPrint_Click.aplf aplsource M
- fmDrawDemoReplay_Click.aplf aplsource M
- fmDrawHelp_Click.aplf aplsource M
- IsForm.aplf aplsource M
- Wfree.aplf aplsource M

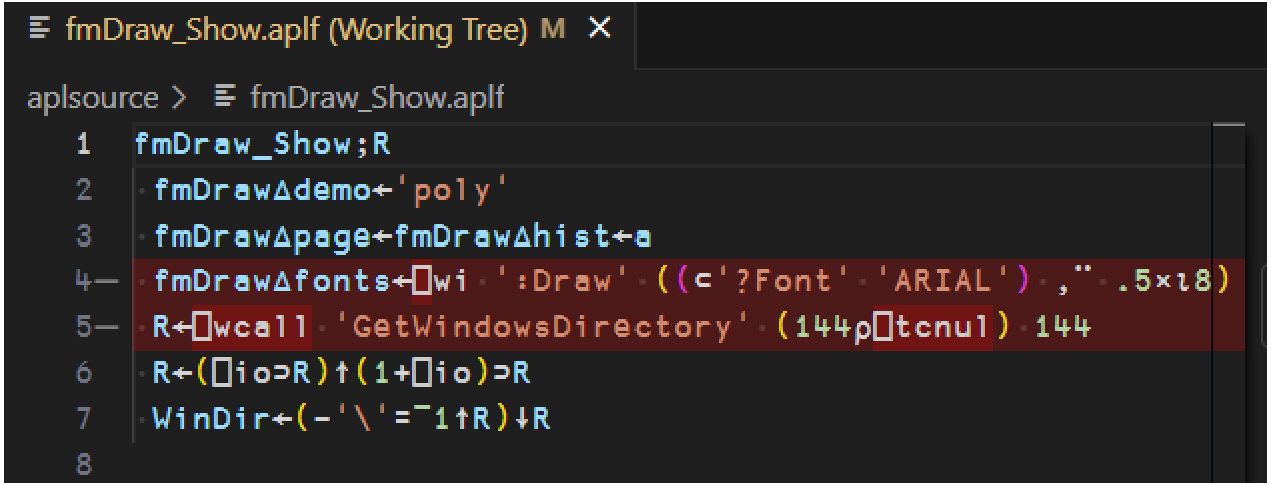
fmDraw_Show.aplf (Working Tree) M

aplsource > fmDraw_Show.aplf

```
 fmDraw_Show;R
  fmDrawΔdemo←'poly'
  fmDrawΔpage←fmDrawΔhist←a
- fmDrawΔfonts←⎕wi ':Draw' ((⊂'?Font' 'ARIAL') ,¨ .5×⍳8)
- R←⎕wcall 'GetWindowsDirectory' (144⍴⎕tcnul) 144
  R←(⎕io⊃R)↑(1+⎕io)⊃R
  WinDir←(-'\'=¯1↑R)↓R
```

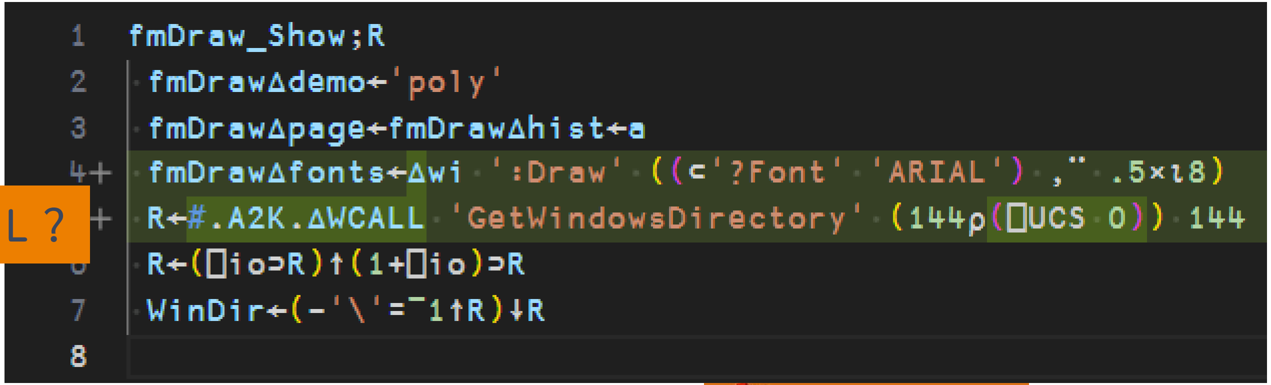
```
 fmDraw_Show;R
  fmDrawΔdemo←'poly'
  fmDrawΔpage←fmDrawΔhist←a
+ fmDrawΔfonts←Δwi ':Draw' ((⊂'?Font' 'ARIAL') ,¨ .5×⍳8)
+ R←#.A2K.ΔWCALL 'GetWindowsDirectory' (144⍴(⎕UCS 0)) 144
  R←(⎕io⊃R)↑(1+⎕io)⊃R
  WinDir←(-'\'=¯1↑R)↓R
```

Original
APL+Win code


fmDraw_Show.aplf (Working Tree) M
aplsource > fmDraw_Show.aplf

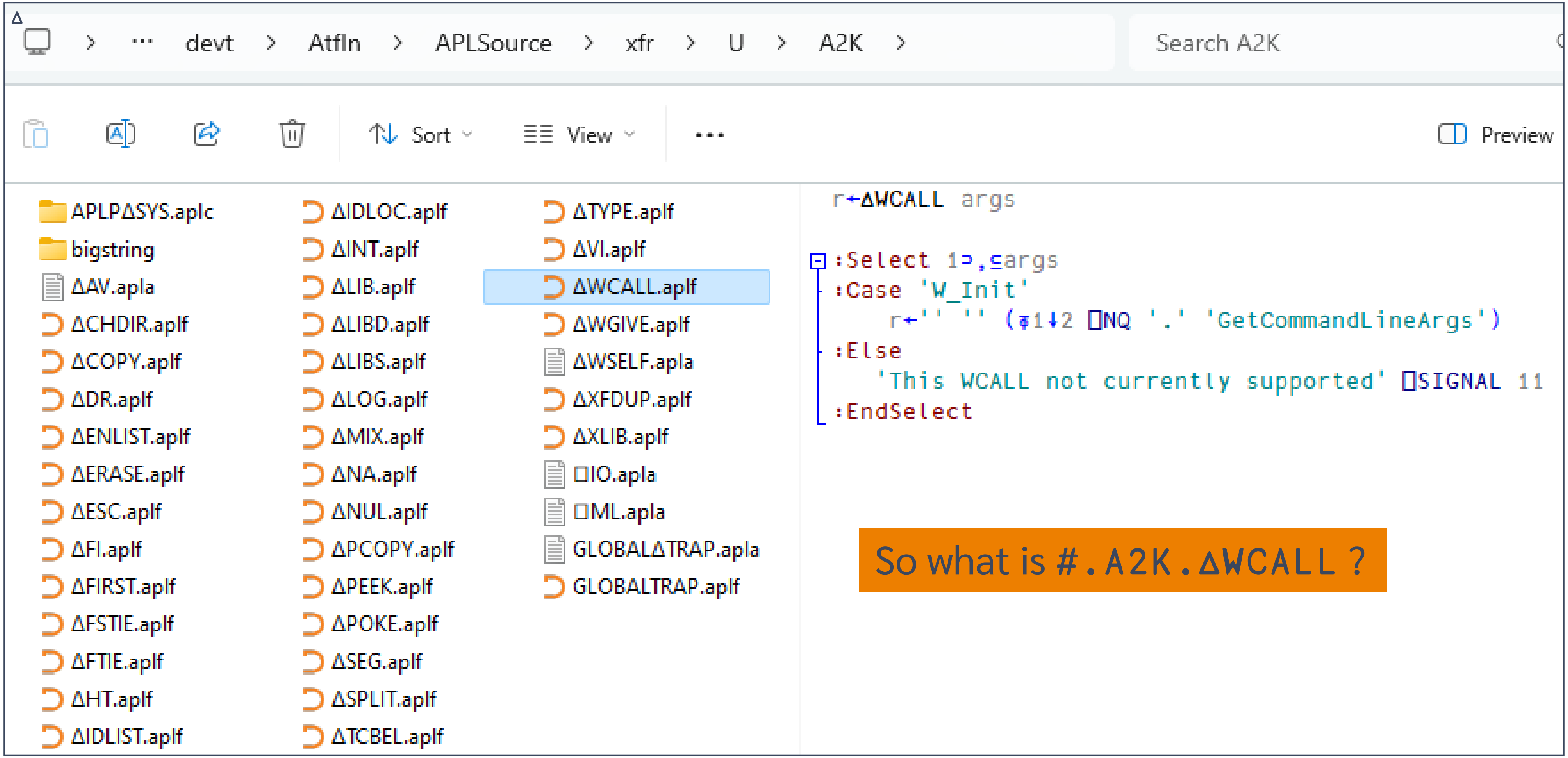
```
fmDraw_Show;R
 fmDrawΔdemo←'poly'
 fmDrawΔpage←fmDrawΔhist←a
 fmDrawΔfonts←⎕wi ':Draw' ((⊂'?Font' 'ARIAL') ,'' .5×⍳8)
 R←⎕wcall 'GetWindowsDirectory' (144⍴⎕tcnul) 144
 R←(⎕io⊃R)↑(1+⎕io)⊃R
 WinDir←(-'\'=¯1↑R)↓R
```

Converted to
Dyalog



```
fmDraw_Show;R
 fmDrawΔdemo←'poly'
 fmDrawΔpage←fmDrawΔhist←a
 fmDrawΔfonts←Δwi ':Draw' ((⊂'?Font' 'ARIAL') ,'' .5×⍳8)
 R←#.A2K.ΔWCALL 'GetWindowsDirectory' (144⍴(⎕UCS 0)) 144
 R←(⎕io⊃R)↑(1+⎕io)⊃R
 WinDir←(-'\'=¯1↑R)↓R
```

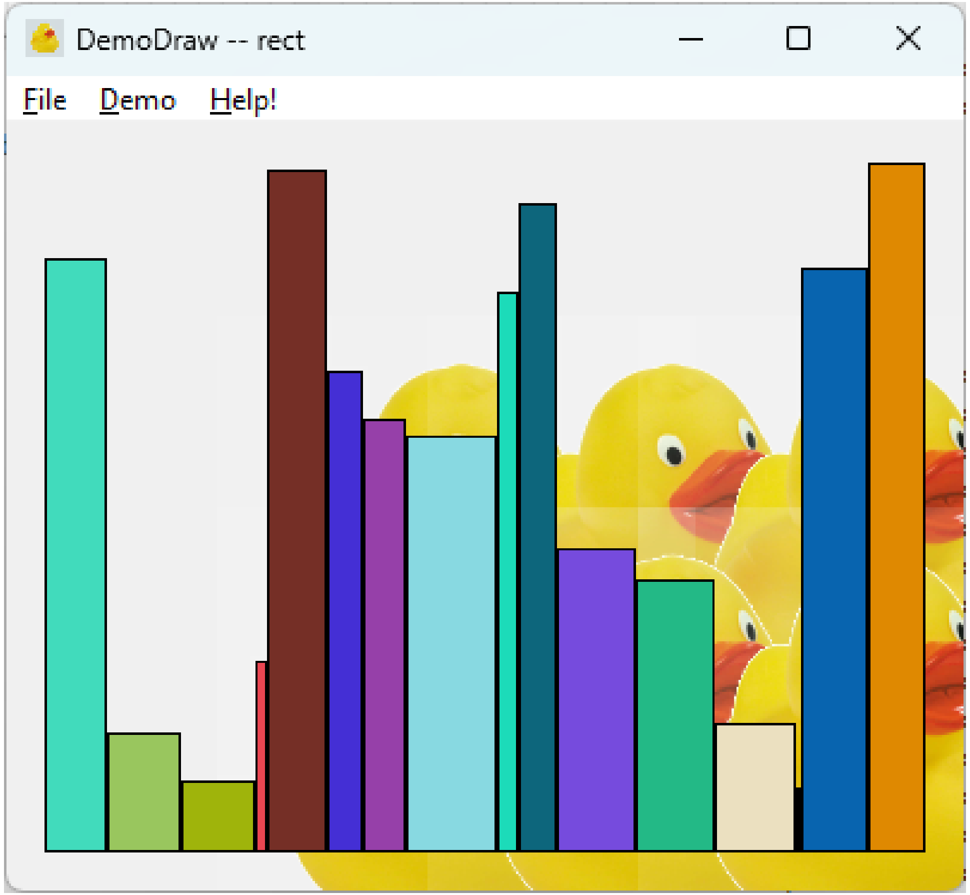
So what is #.A2K.ΔWCALL ?

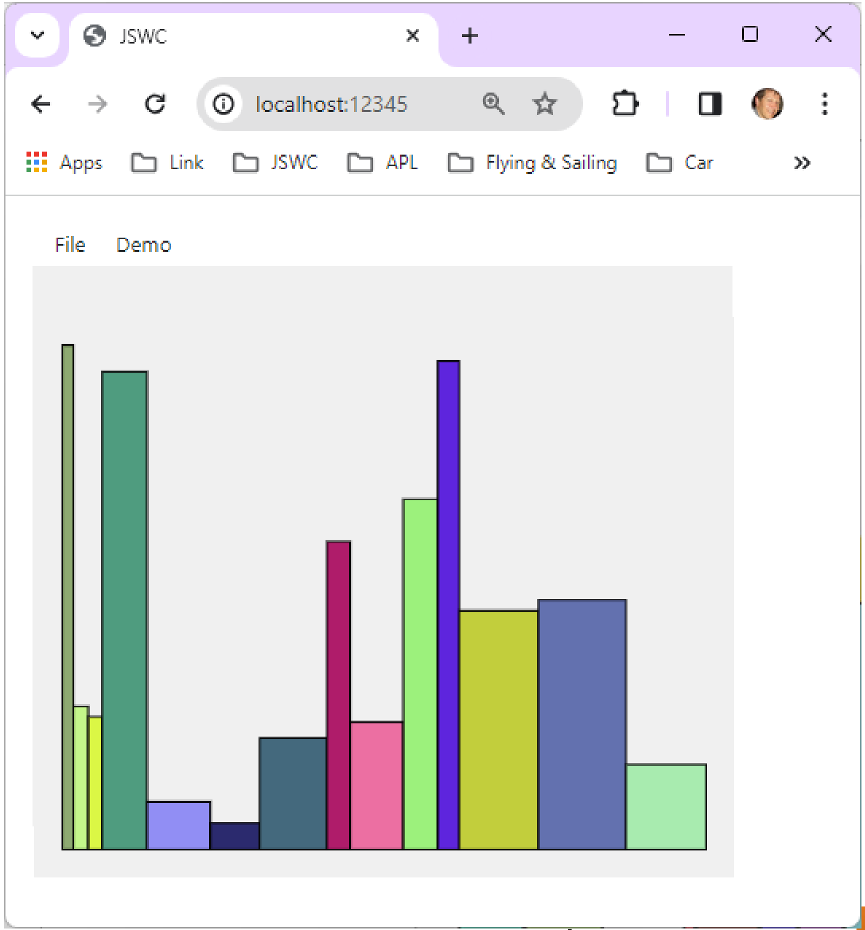


APLPΔSYS.aplc
bigstring
ΔAV.apla
ΔCHDIR.aplf
ΔCOPY.aplf
ΔDR.aplf
ΔENLIST.aplf
ΔERASE.aplf
ΔESC.aplf
ΔFI.aplf
ΔFIRST.aplf
ΔFSTIE.aplf
ΔFTIE.aplf
ΔHT.aplf
ΔIDLIST.aplf
ΔIDLOC.aplf
ΔINT.aplf
ΔLIB.aplf
ΔLIBD.aplf
ΔLIBS.aplf
ΔLOG.aplf
ΔMIX.aplf
ΔNA.aplf
ΔNUL.aplf
ΔPCOPY.aplf
ΔPEEK.aplf
ΔPOKE.aplf
ΔSEG.aplf
ΔSPLIT.aplf
ΔTCBEL.aplf
ΔTYPE.aplf
ΔVI.aplf
ΔWCALL.aplf
ΔWGIVE.aplf
ΔWSELF.apla
ΔXFDUP.aplf
ΔXLIB.aplf
⎕IO.apla
⎕ML.apla
GLOBALΔTRAP.apla
GLOBALTRAP.aplf

```
 r←ΔWCALL args

:Select 1⊃,⊆args
:Case 'W_Init'
    r←'' '' (⍕1↓2 ⎕NQ '.' 'GetCommandLineArgs')
:Else
   'This WCALL not currently supported' ⎕SIGNAL 11
:EndSelect
```

So what is #.A2K.ΔWCALL ?


DemoDraw -- rect
File
Demo
Help!



JSWC
localhost:12345
Apps
Link
JSWC
APL
Flying & Sailing
Car
File
Demo

Status

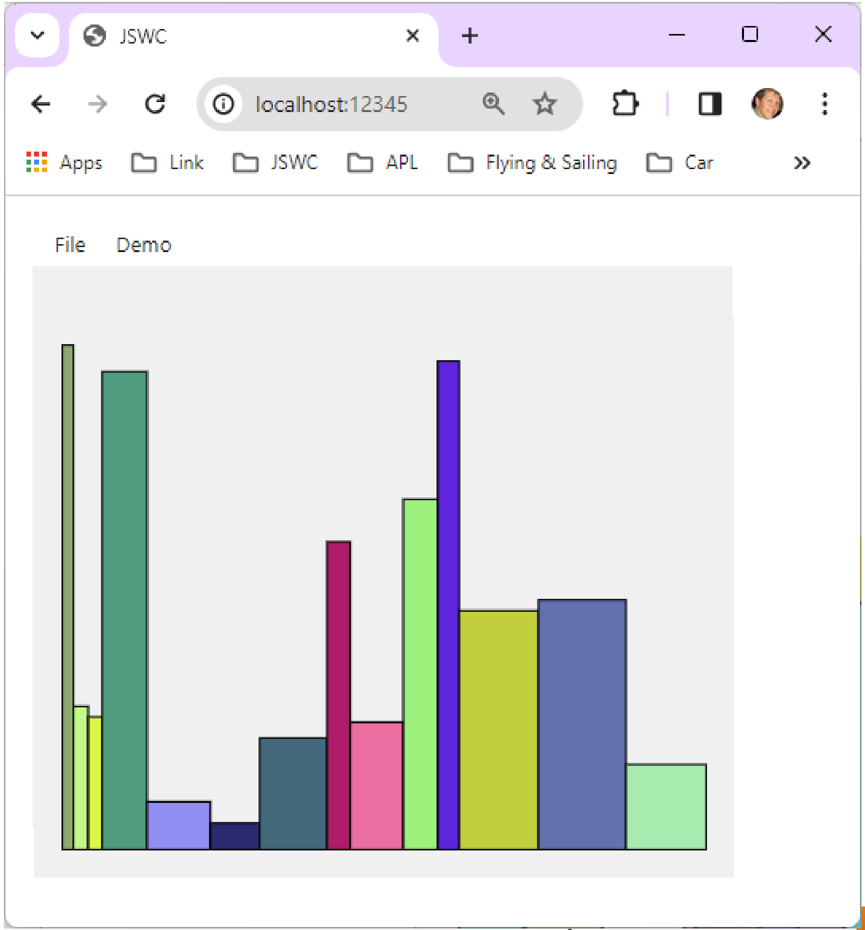
- We are reworking and enhancing the tools developed for APLX migrations
- Will enumerate differences between APL+Win and Dyalog
- Will create emulation functions as required

- We *may* also decide to add new features to Dyalog v20 (which should start user testing in late 2024)
  - For example `:LeaveIf`

Status

- Dyalog has been contracted to port the METSIM® application
- Hired one new APL developer in January, aiming for 1 more
- We will have ~1.5-2 full time equivalent resources working on migration tools until further notice

- **All the resulting tools and documentation will be free and open source**


JSWC
localhost:12345
Apps
Link
JSWC
APL
Flying & Sailing
Car
File
Demo