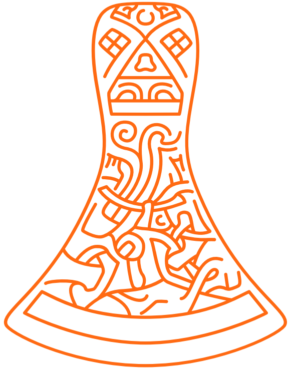
DYALOG

Glasgow 2024

Migrating APL+Win Applications

Karl Holt, Morten Kromberg



A New Wave of Migrants

- Dyalog APL was created by Dyadic Systems Ltd, when the mainframe consulting business faltered (1981)
- Most current users of Dyalog APL migrated from SHARP APL, IBM APL2, APL+Win, or 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)

From APL2

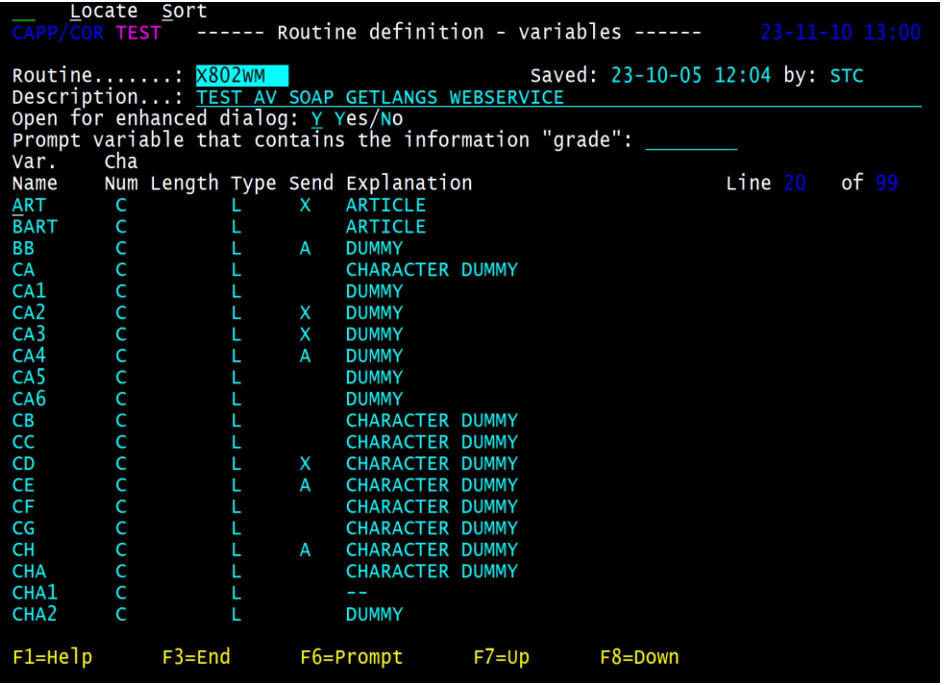
Relatively straightforward

- A few language differences
  - Each on empty arrays
- User Interfaces and file I/O are usually already handled by simple cover-functions and can be emulated “easily”
- Linux or Windows apps may be making external calls which will require "tweaking"
- We are considering implementing "format by example" but so far it has not been necessary

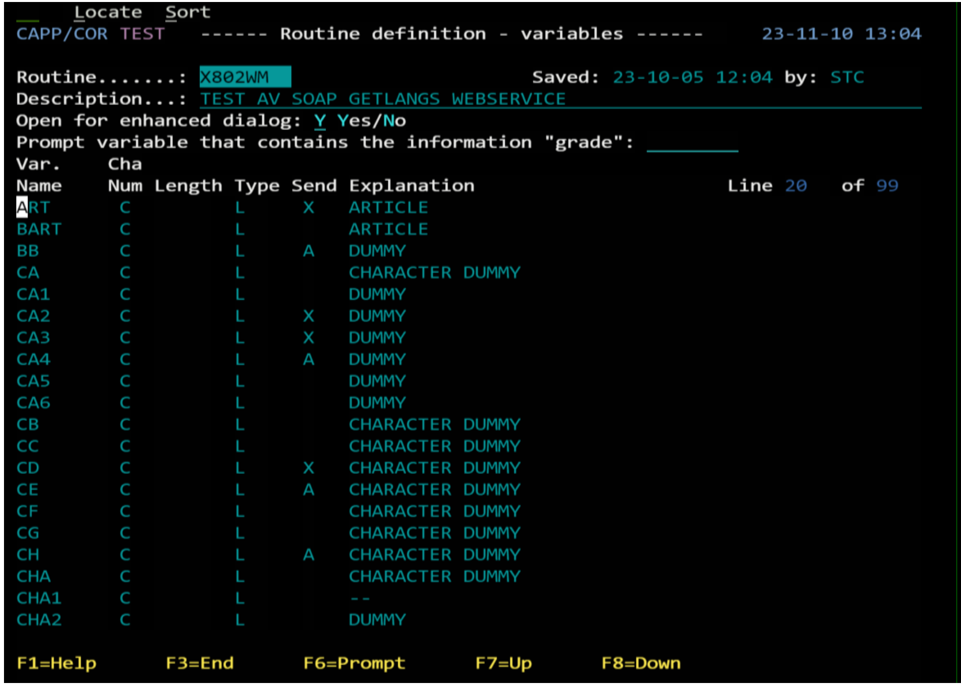
Recent / Active APL2 Migrations

- 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 Mark Wolfson @ BIG "with a little help"

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
```

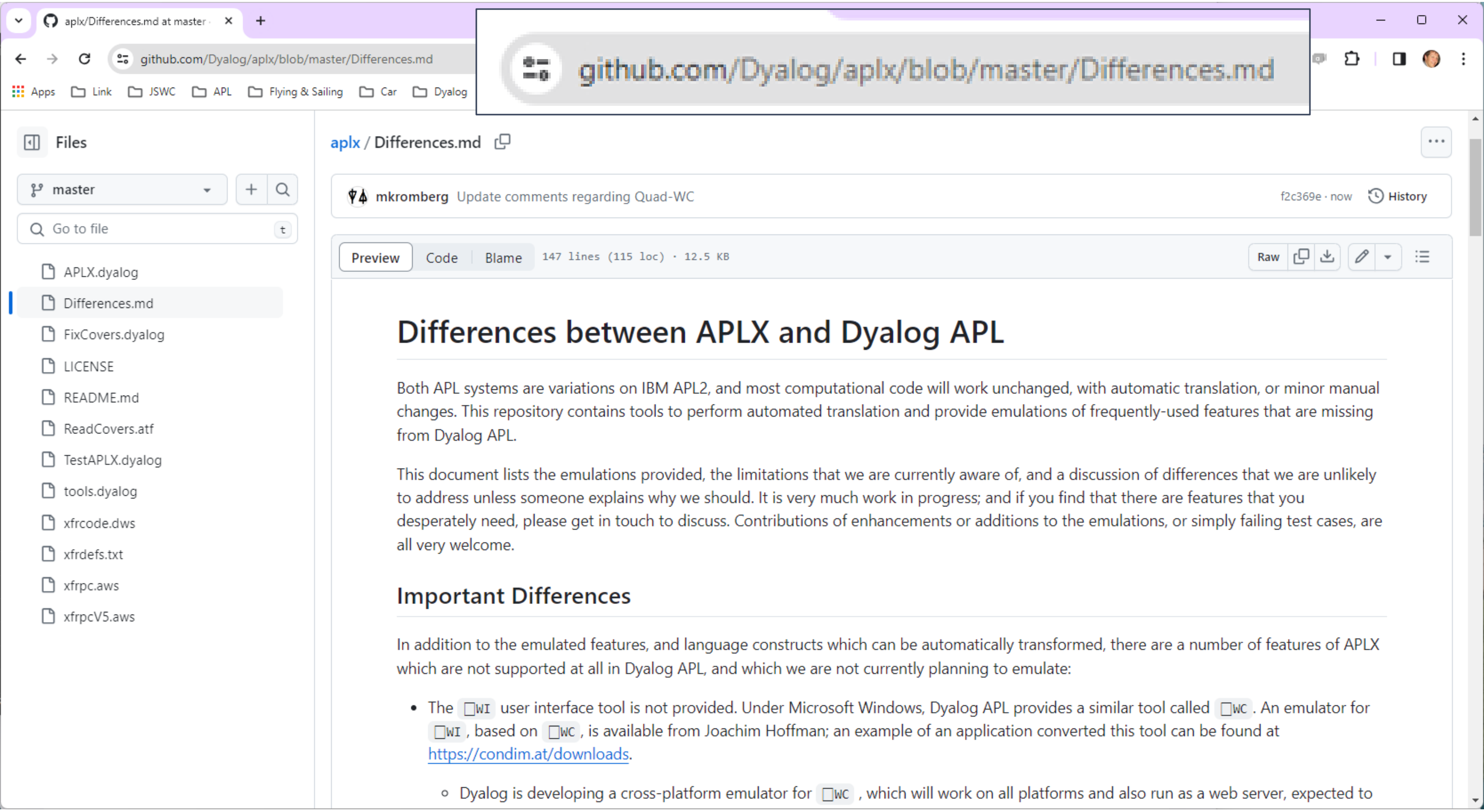
From APL+Win or MicroAPL APLX

Same language differences as APL2, plus:

- Many system functions & control structures not found in Dyalog APL
- Double quotes (`"Don't do this!"`)
- More 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
- A handful of users migrated using these tools



Differences between APLX and Dyalog APL

Both APL systems are variations on IBM APL2, and most computational code will work unchanged, with automatic translation, or minor manual changes. This repository contains tools to perform automated translation and provide emulations of frequently-used features that are missing from Dyalog APL.

This document lists the emulations provided, the limitations that we are currently aware of, and a discussion of differences that we are unlikely to address unless someone explains why we should. It is very much work in progress; and if you find that there are features that you desperately need, please get in touch to discuss. Contributions of enhancements or additions to the emulations, or simply failing test cases, are all very welcome.

Important Differences

In addition to the emulated features, and language constructs which can be automatically transformed, there are a number of features of APLX which are not supported at all in Dyalog APL, and which we are not currently planning to emulate:

- The `⎕WI` user interface tool is not provided. Under Microsoft Windows, Dyalog APL provides a similar tool called `⎕WC`. An emulator for `⎕WI`, based on `⎕WC`, is available from Joachim Hoffman; an example of an application converted this tool can be found at https://condim.at/downloads.
    - Dyalog is developing a cross-platform emulator for `⎕WC`, which will work on all platforms and also run as a web server, expected to

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
  - Migration being handled by Dyalog
  - Will be used to develop tools to automate migration, including the Graphical User Interface
- More under discussion

Migrating APL+Win Applications

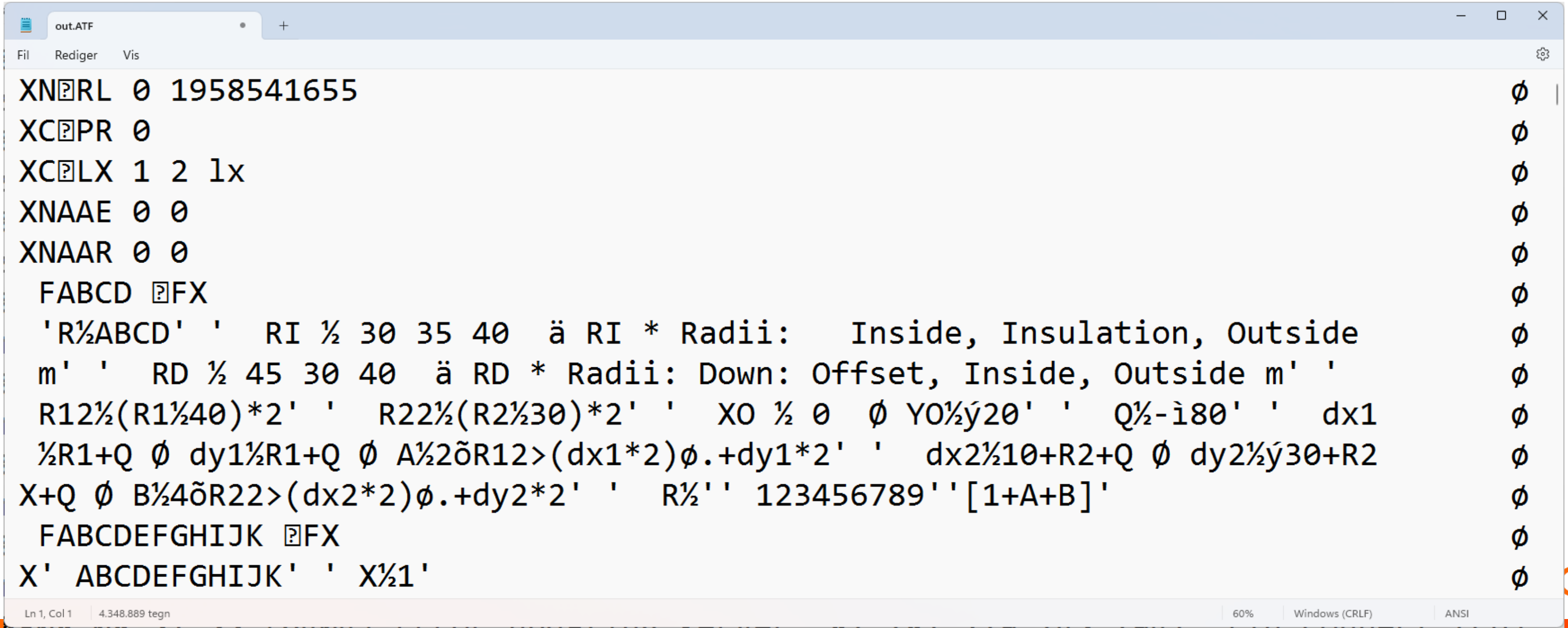
Step 1: Export Source Code

- APL Workspace Transfer format is a standard agreed by APL vendors before 1980
- Most APL systems provide user or system commands IN and OUT to read and write this format
- The APL+Win user command `]OUT` creates a file in Transfer Format:

```
]OUT /tmp/out    ➔    /tmp/out.ATF
```

Step 1 – Export Source Code

- Result of `]OUT /tmp/out`:

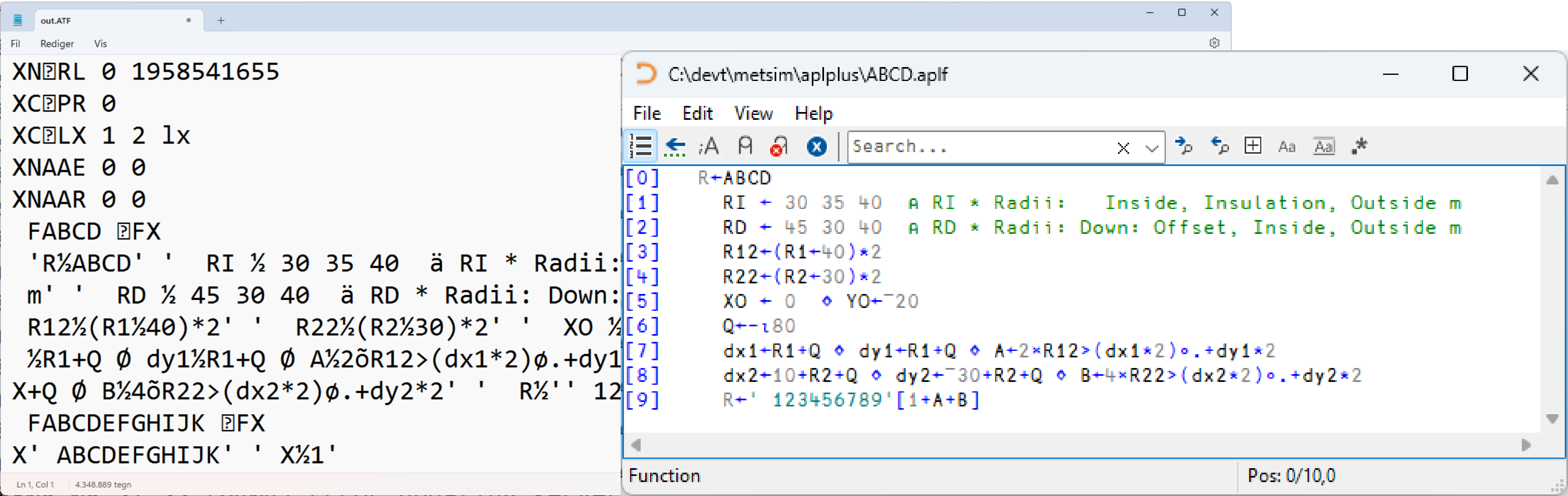


```
XN⍰RL 0 1958541655                                                             ϕ
XC⍰PR 0                                                                        ϕ
XC⍰LX 1 2 lx                                                                   ϕ
XNAAE 0 0                                                                      ϕ
XNAAR 0 0                                                                      ϕ
 FABCD ⍰FX                                                                     ϕ
 'R½ABCD' '  RI ½ 30 35 40  ä RI * Radii:   Inside, Insulation, Outside        ϕ
 m' '  RD ½ 45 30 40  ä RD * Radii: Down: Offset, Inside, Outside m' '         ϕ
 R12½(R1½40)*2' '  R22½(R2½30)*2' '  XO ½ 0  Ø YO½ý20' '  Q½-ì80' '  dx1       ϕ
 ½R1+Q Ø dy1½R1+Q Ø A½2õR12>(dx1*2)ϕ.+dy1*2' '  dx2½10+R2+Q Ø dy2½ý30+R2       ϕ
X+Q Ø B½4õR22>(dx2*2)ϕ.+dy2*2' '  R½'' 123456789''[1+A+B]'                     ϕ
 FABCDEFGHIJK ⍰FX                                                              ϕ
X' ABCDEFGHIJK' ' X½1'                                                         ϕ
```

Step 2: Create Text Source

- Our new `]IN` command can create text source files

```
]IN /tmp/out.ATF -outdir=/path/aplplus -apl=APLPLUS
```



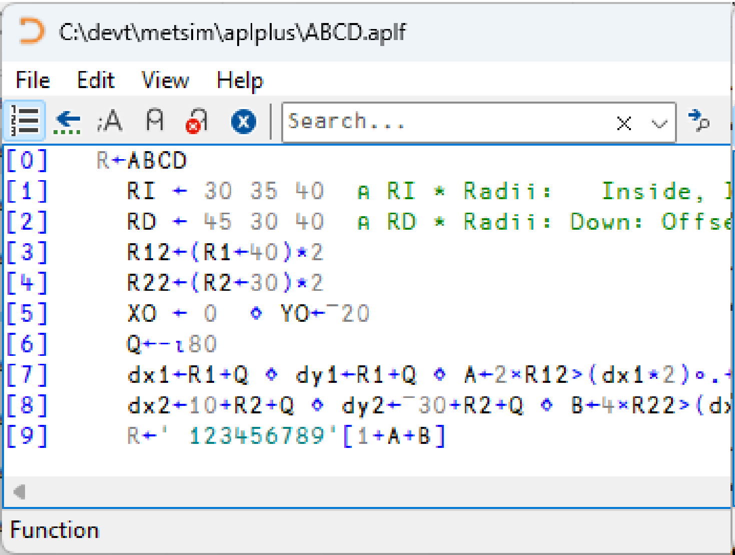
```
XN?RL 0 1958541655
XC?PR 0
XC?LX 1 2 lx
XNAAE 0 0
XNAAR 0 0
 FABCD ?FX
 'R½ABCD' '  RI ½ 30 35 40  ä RI * Radii:
 m' '  RD ½ 45 30 40  ä RD * Radii: Down:
 R12½(R1½40)*2' '  R22½(R2½30)*2' '  XO ½
 ½R1+Q Ø dy1½R1+Q Ø A½2õR12>(dx1*2)ø.+dy1
X+Q Ø B½4õR22>(dx2*2)ø.+dy2*2' '  R½'' 12
 FABCDEFGHIJK ?FX
X' ABCDEFGHIJK' ' X½1'
```

```
    R←ABCD
      RI ← 30 35 40  ⍝ RI * Radii:   Inside, Insulation, Outside m
      RD ← 45 30 40  ⍝ RD * Radii: Down: Offset, Inside, Outside m
      R12←(R1←40)⋆2
      R22←(R2←30)⋆2
      XO ← 0  ⋄ YO←¯20
      Q←-⍳80
      dx1←R1+Q ⋄ dy1←R1+Q ⋄ A←2×R12>(dx1⋆2)∘.+dy1⋆2
      dx2←10+R2+Q ⋄ dy2←¯30+R2+Q ⋄ B←4×R22>(dx2⋆2)∘.+dy2⋆2
      R←' 123456789'[1+A+B]
```

Step 3: Automatic Conversion

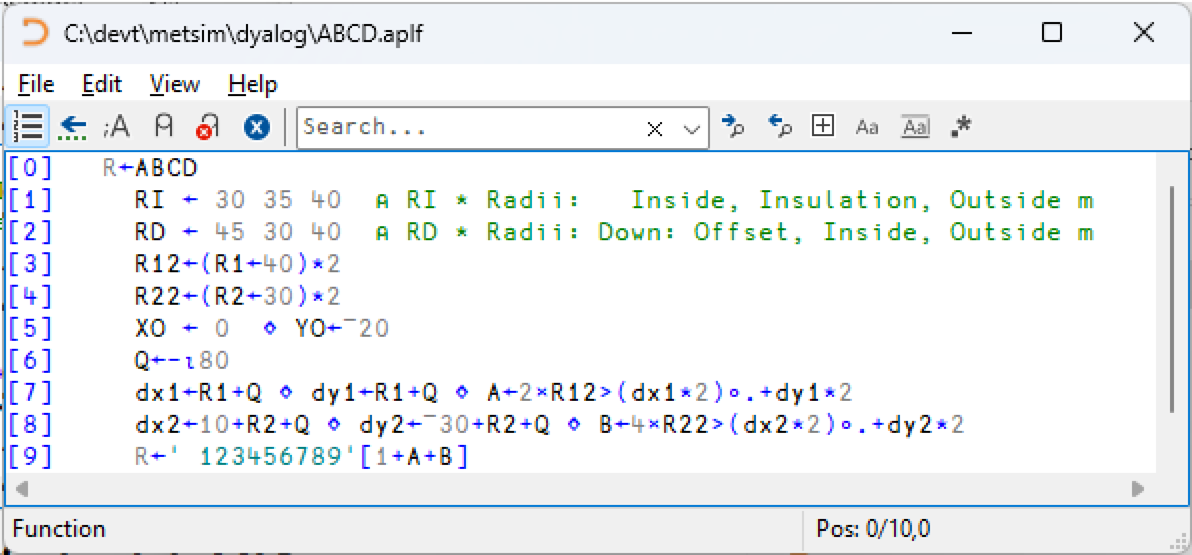
- Apply automated transformations to the original source

```
]todyalog /path/aplplus /path/dyalog A2K
```



C:\devt\metsim\aplplus\ABCD.aplf

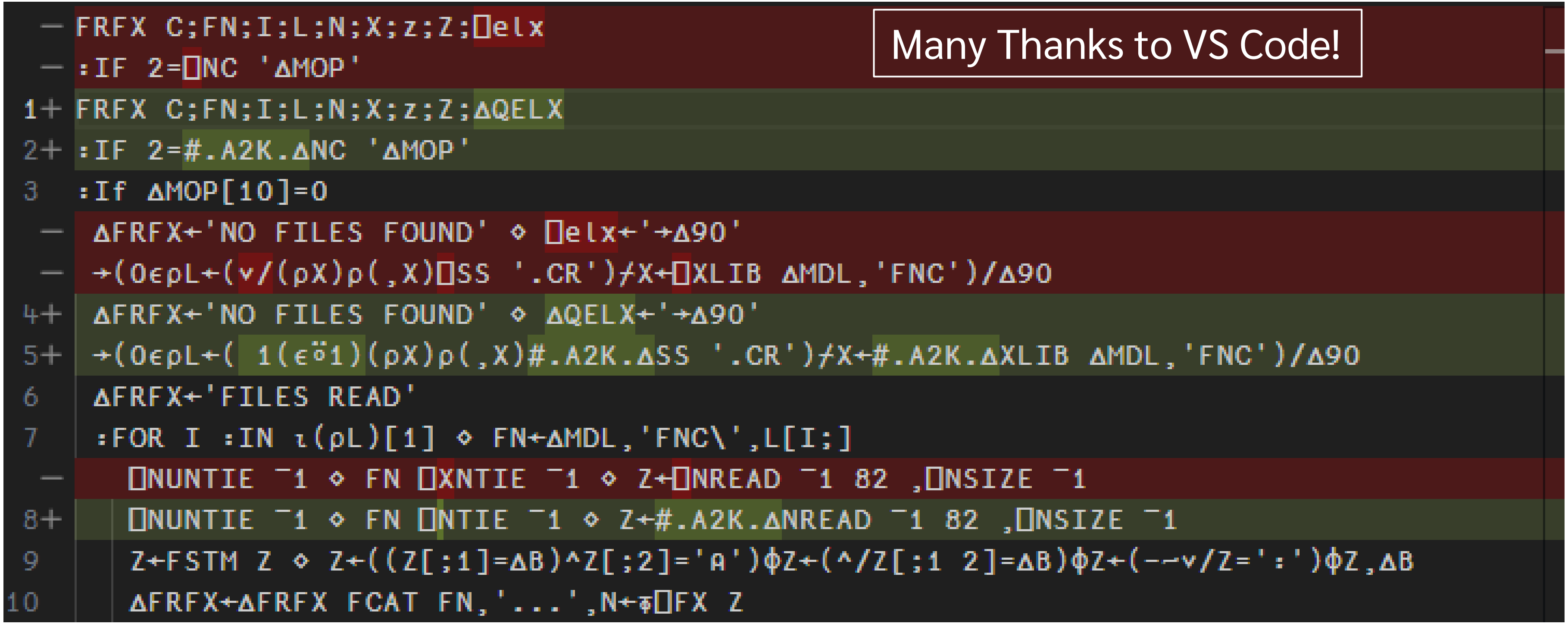
```
[0]    R←ABCD
[1]      RI ← 30 35 40  ⍝ RI * Radii:   Inside, I
[2]      RD ← 45 30 40  ⍝ RD * Radii: Down: Offse
[3]      R12←(R1←40)*2
[4]      R22←(R2←30)*2
[5]      XO ← 0   ⋄ YO←¯20
[6]      Q←-⍳80
[7]      dx1←R1+Q ⋄ dy1←R1+Q ⋄ A←2×R12>(dx1*2)∘.+
[8]      dx2←10+R2+Q ⋄ dy2←¯30+R2+Q ⋄ B←4×R22>(dx
[9]      R←' 123456789'[1+A+B]
```



C:\devt\metsim\dyalog\ABCD.aplf

```
[0]    R←ABCD
[1]      RI ← 30 35 40  ⍝ RI * Radii:   Inside, Insulation, Outside m
[2]      RD ← 45 30 40  ⍝ RD * Radii: Down: Offset, Inside, Outside m
[3]      R12←(R1←40)*2
[4]      R22←(R2←30)*2
[5]      XO ← 0   ⋄ YO←¯20
[6]      Q←-⍳80
[7]      dx1←R1+Q ⋄ dy1←R1+Q ⋄ A←2×R12>(dx1*2)∘.+dy1*2
[8]      dx2←10+R2+Q ⋄ dy2←¯30+R2+Q ⋄ B←4×R22>(dx2*2)∘.+dy2*2
[9]      R←' 123456789'[1+A+B]
```

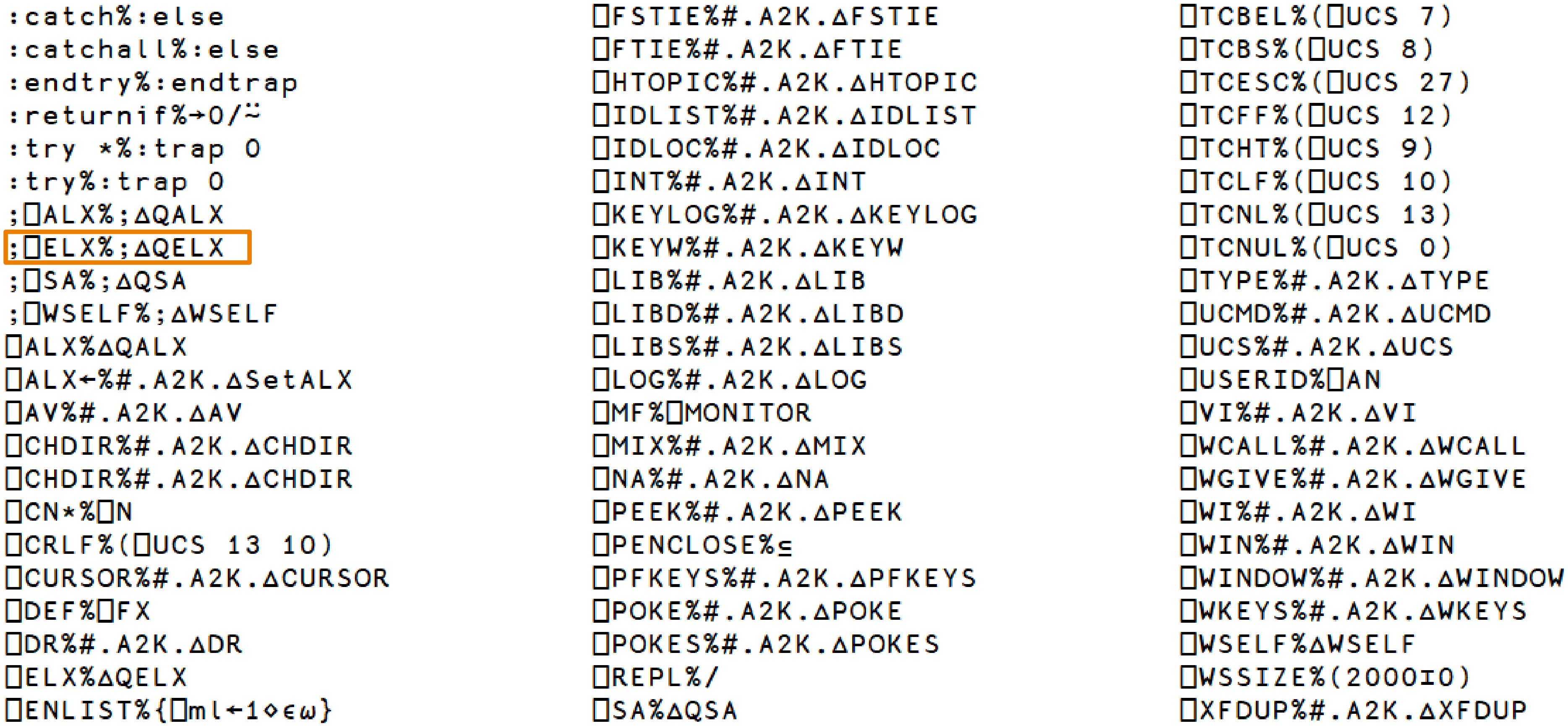
Automatic Substitutions



```
- FRFX C;FN;I;L;N;X;z;Z;⎕elx
- :IF 2=⎕NC 'ΔMOP'
+ FRFX C;FN;I;L;N;X;z;Z;ΔQELX
+ :IF 2=#.A2K.ΔNC 'ΔMOP'
  :If ΔMOP[10]=0
-  ΔFRFX←'NO FILES FOUND' ⋄ ⎕elx←'→Δ90'
-  →(0∊ρL←(∨/(ρX)ρ(,X)⎕SS '.CR')≠X←⎕XLIB ΔMDL,'FNC')/Δ90
+  ΔFRFX←'NO FILES FOUND' ⋄ ΔQELX←'→Δ90'
+  →(0∊ρL←( 1(∊⍤1)(ρX)ρ(,X)#.A2K.ΔSS '.CR')≠X←#.A2K.ΔXLIB ΔMDL,'FNC')/Δ90
   ΔFRFX←'FILES READ'
   :FOR I :IN ⍳(ρL)[1] ⋄ FN←ΔMDL,'FNC\',L[I;]
-    ⎕NUNTIE ¯1 ⋄ FN ⎕XNTIE ¯1 ⋄ Z←⎕NREAD ¯1 82 ,⎕NSIZE ¯1
+    ⎕NUNTIE ¯1 ⋄ FN ⎕NTIE ¯1 ⋄ Z←#.A2K.ΔNREAD ¯1 82 ,⎕NSIZE ¯1
     Z←FSTM Z ⋄ Z←((Z[;1]=ΔB)^Z[;2]='⍝')⌽Z←(^/Z[;1 2]=ΔB)⌽Z←(--∨/Z=':')⌽Z,ΔB
     ΔFRFX←ΔFRFX FCAT FN,'...',N←⍕⎕FX Z
```

Many Thanks to VS Code!

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
```

System Functions Emulated

⎕XLIB => #.A2K.∆XLIB

```
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

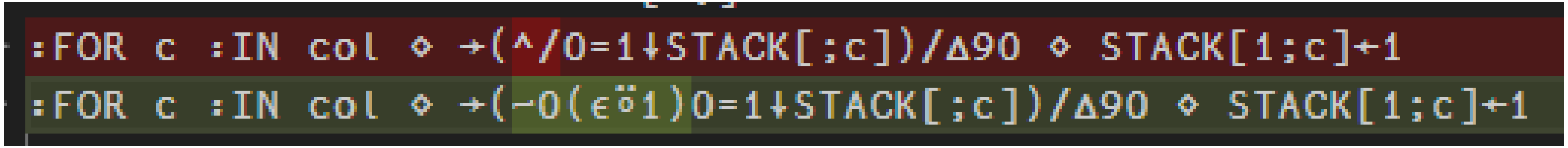
Detect and rewrite

```
:LeaveIf
```

Enhance Interpreter?

Static Analysis could help with some of these

Work in Progress



```
:FOR c :IN col ⋄ →(∧/0=1↓STACK[;c])/Δ90 ⋄ STACK[1;c]←1
:FOR c :IN col ⋄ →(~0(∊⍤1)0=1↓STACK[;c])/Δ90 ⋄ STACK[1;c]←1
```

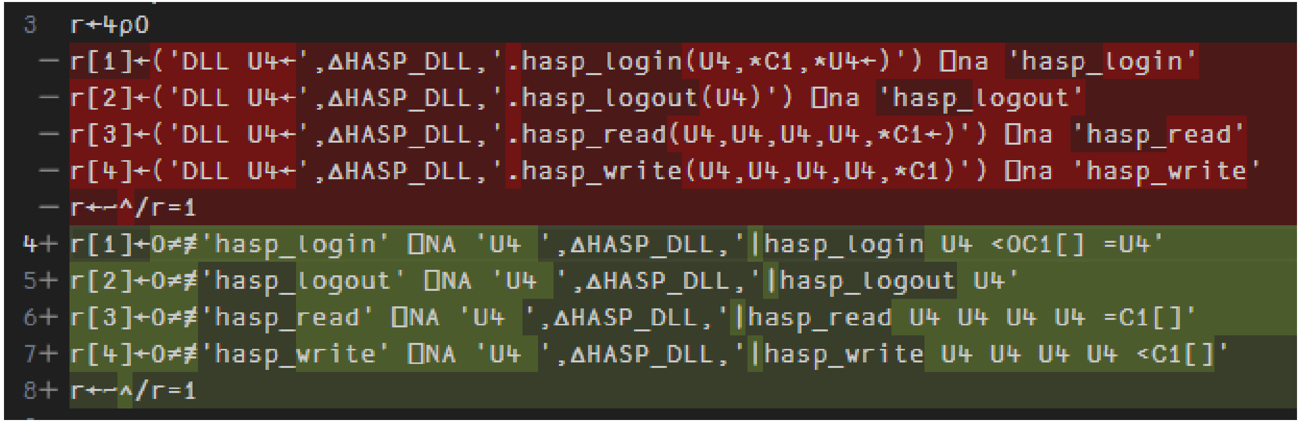
- ∧/ on empty nested arrays can return a result with a different prototype
- The above is a quick hack to allow us to make progress, probably we will map

```
∧/ to All
∨/ to Any
```

Manual Steps

Foreign Function Calls

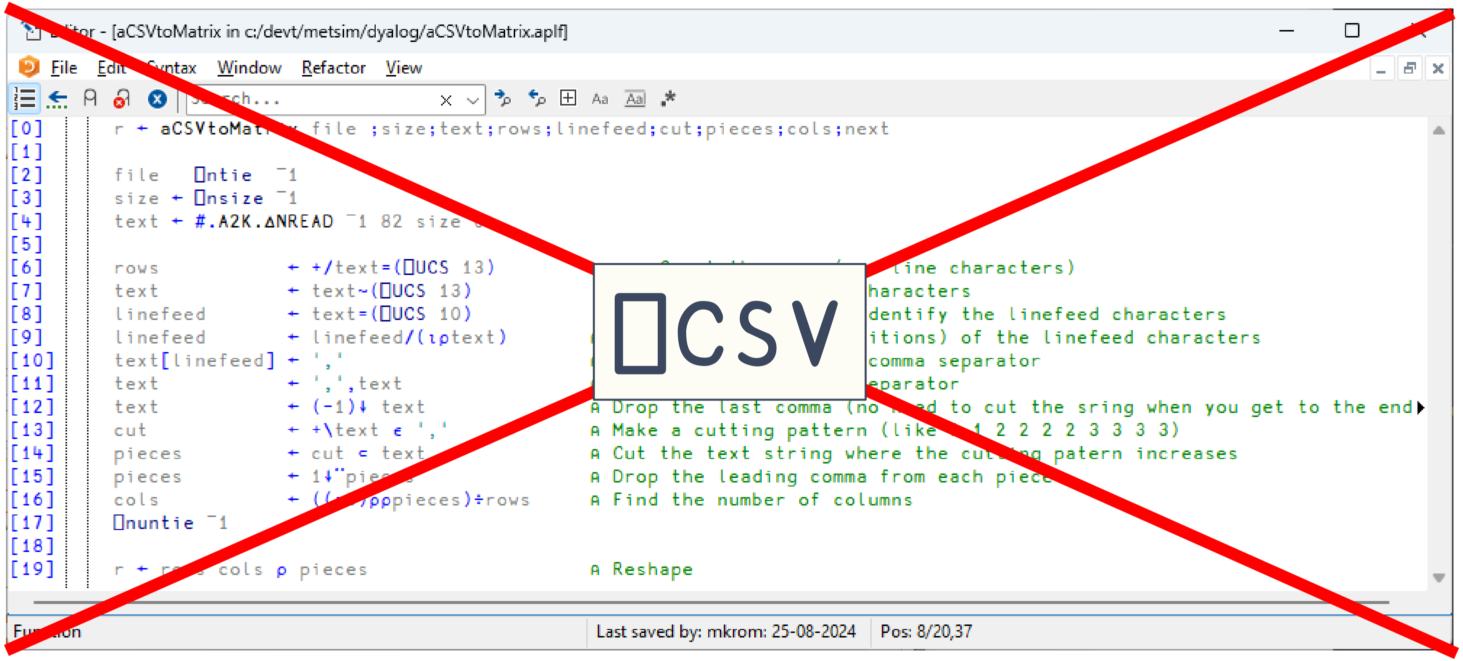
- Mechanical manual translation



```
  r←4⍴0
- r[1]←('DLL U4←',∆HASP_DLL,'.hasp_login(U4,*C1,*U4←)') ⎕na 'hasp_login'
- r[2]←('DLL U4←',∆HASP_DLL,'.hasp_logout(U4)') ⎕na 'hasp_logout'
- r[3]←('DLL U4←',∆HASP_DLL,'.hasp_read(U4,U4,U4,U4,*C1←)') ⎕na 'hasp_read'
- r[4]←('DLL U4←',∆HASP_DLL,'.hasp_write(U4,U4,U4,U4,*C1)') ⎕na 'hasp_write'
- r←~^/r=1
+ r[1]←0≠≢'hasp_login' ⎕NA 'U4 ',∆HASP_DLL,'|hasp_login U4 <0C1[] =U4'
+ r[2]←0≠≢'hasp_logout' ⎕NA 'U4 ',∆HASP_DLL,'|hasp_logout U4'
+ r[3]←0≠≢'hasp_read' ⎕NA 'U4 ',∆HASP_DLL,'|hasp_read U4 U4 U4 U4 =C1[]'
+ r[4]←0≠≢'hasp_write' ⎕NA 'U4 ',∆HASP_DLL,'|hasp_write U4 U4 U4 U4 <C1[]'
+ r←~∧/r=1
```

- We *may* add automated conversion

CSV files

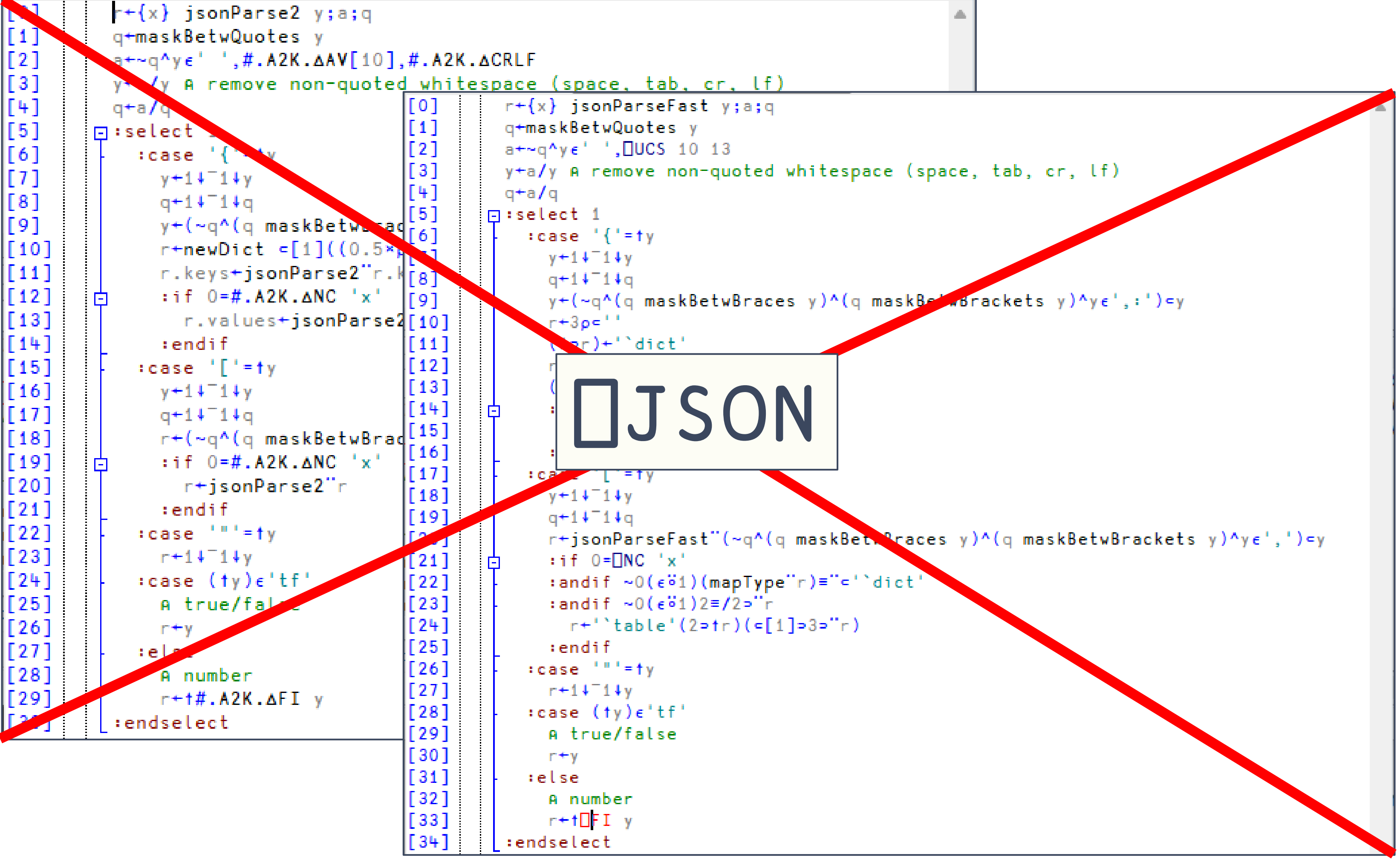


Editor - [aCSVtoMatrix in c:/devt/metsim/dyalog/aCSVtoMatrix.aplf]
File
Edit
Syntax
Window
Refactor
View
Search...

```
 r ← aCSVtoMatrix file ;size;text;rows;linefeed;cut;pieces;cols;next

 file   ⎕ntie ¯1
 size ← ⎕nsize ¯1
 text ← #.A2K.∆NREAD ¯1 82 size

 rows              ← +/text=(⎕UCS 13)                       line characters)
 text              ← text~(⎕UCS 13)                        haracters
 linefeed          ← text=(⎕UCS 10)                        dentify the linefeed characters
 linefeed          ← linefeed/(⍳⍴text)                     itions) of the linefeed characters
 text[linefeed] ← ','                                      comma separator
 text              ← ',',text                              eparator
 text              ← (-1)↓ text                 ⍝ Drop the last comma (no need to cut the sring when you get to the end
 cut               ← +\text ∊ ','               ⍝ Make a cutting pattern (like 1 1 2 2 2 2 3 3 3 3)
 pieces            ← cut ⊂ text                 ⍝ Cut the text string where the cutting patern increases
 pieces            ← 1↓¨pieces                  ⍝ Drop the leading comma from each piece
 cols              ← ((+/ρρpieces)÷rows         ⍝ Find the number of columns
 ⎕nuntie ¯1

 r ← rows cols ρ pieces                         ⍝ Reshape
```

Function | Last saved by: mkrom: 25-08-2024 | Pos: 8/20,37

⎕CSV



```
r←{x} jsonParse2 y;a;q
q←maskBetwQuotes y
a←~q^yε' ',#.A2K.∆AV[10],#.A2K.∆CRLF
y←a/y ⍝ remove non-quoted whitespace (space, tab, cr, lf)
q←a/q
:select 1
  :case '{'=↑y
    y←1↓¯1↓y
    q←1↓¯1↓q
    y←(~q^(q maskBetwBrac
    r←newDict ⊂[1]((0.5×ρ
    r.keys←jsonParse2¨r.k
    :if 0=#.A2K.∆NC 'x'
      r.values←jsonParse2
    :endif
  :case '['=↑y
    y←1↓¯1↓y
    q←1↓¯1↓q
    r←(~q^(q maskBetwBrac
    :if 0=#.A2K.∆NC 'x'
      r←jsonParse2¨r
    :endif
  :case '"'=↑y
    r←1↓¯1↓y
  :case (↑y)ε'tf'
    ⍝ true/false
    r←y
  :else
    ⍝ number
    r←↑#.A2K.∆FI y
:endselect
```

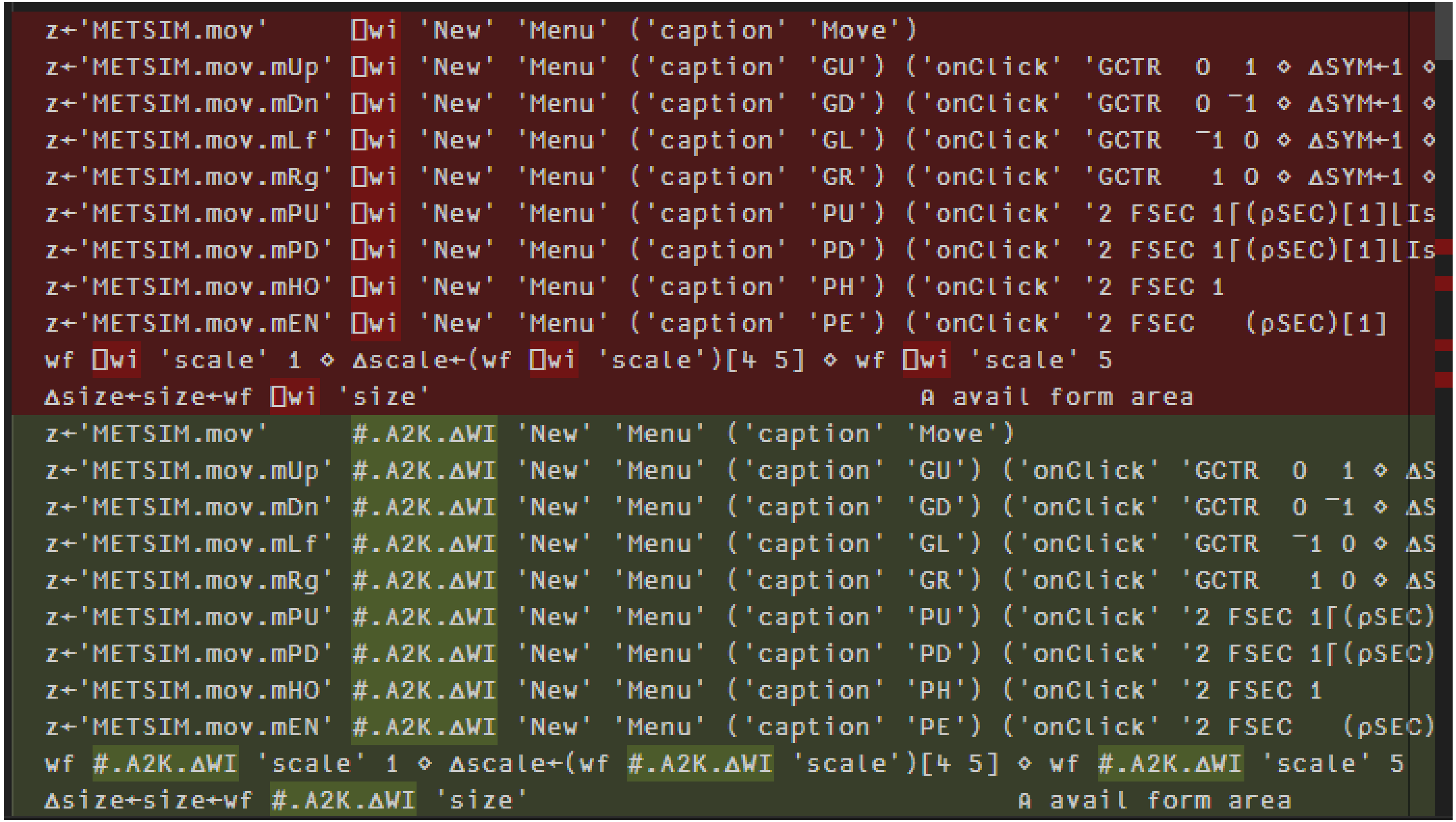
```
r←{x} jsonParseFast y;a;q
q←maskBetwQuotes y
a←~q^yε' ',⎕UCS 10 13
y←a/y ⍝ remove non-quoted whitespace (space, tab, cr, lf)
q←a/q
:select 1
  :case '{'=↑y
    y←1↓¯1↓y
    q←1↓¯1↓q
    y←(~q^(q maskBetwBraces y)^(q maskBetwBrackets y)^yε',:')⊂y
    r←3ρ⊂''
    (⊃r)←'`dict'
  :case '['=↑y
    y←1↓¯1↓y
    q←1↓¯1↓q
    r←jsonParseFast¨(~q^(q maskBetwBraces y)^(q maskBetwBrackets y)^yε',')⊂y
    :if 0=⎕NC 'x'
    :andif ~0(ε⍤1)(mapType¨r)≡¨⊂'`dict'
    :andif ~0(ε⍤1)2≡/2⊃¨r
      r←'`table'(2⊃↑r)(⊂[1]⊃3⊃¨r)
    :endif
  :case '"'=↑y
    r←1↓¯1↓y
  :case (↑y)ε'tf'
    ⍝ true/false
    r←y
  :else
    ⍝ number
    r←↑⎕FI y
:endselect
```

⎕JSON

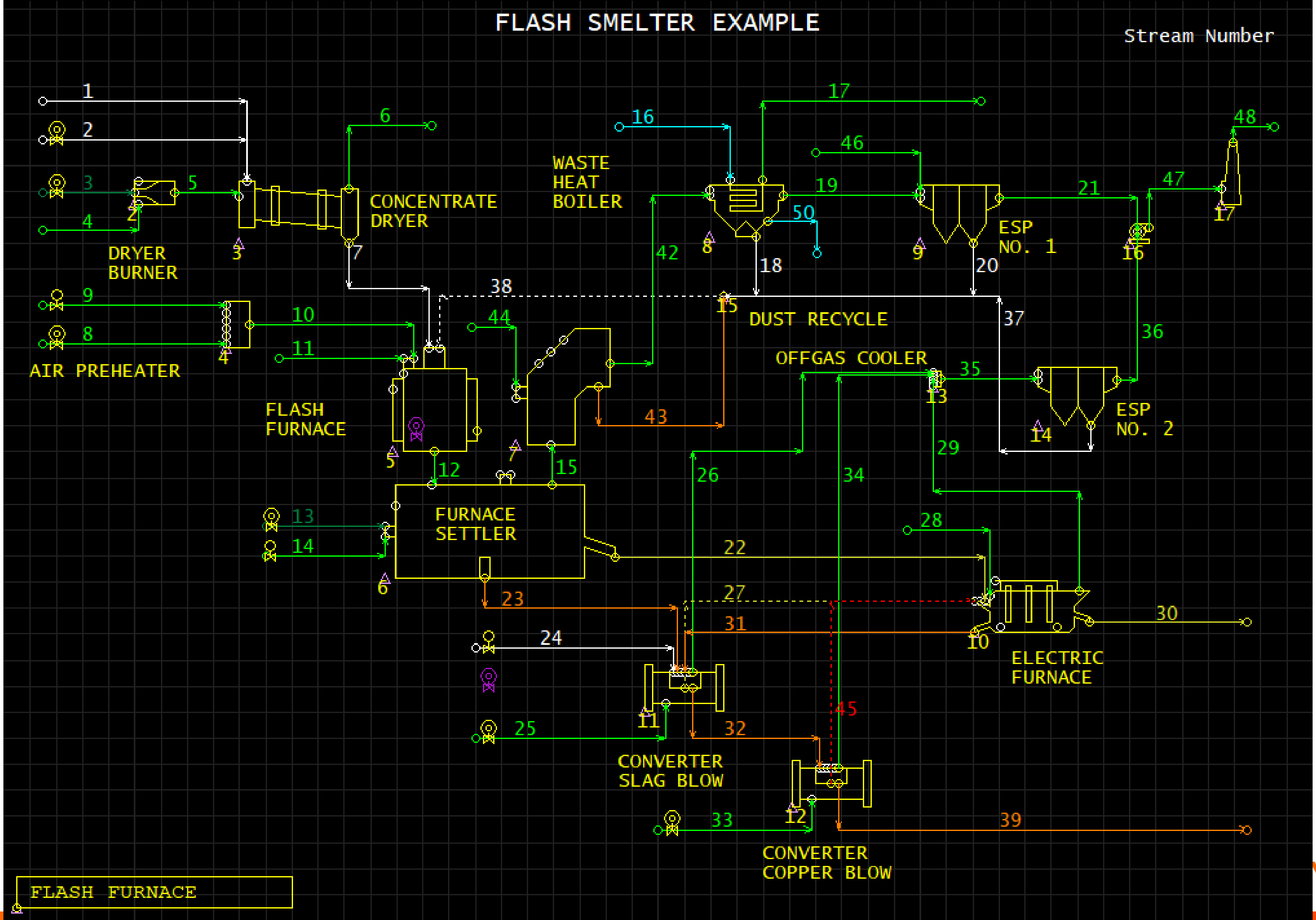
Component Files

- Emulations of ⎕F* system functions which can read ***and*** write APL+Win component files directly
  - Uses an APL+Win runtime application and binary SCAR format via TCP sockets
- Allows parallel operation of old + new versions of the application code
- Component files can be migrated gradually

The Hard Part



```
z←'METSIM.mov'     ⎕wi 'New' 'Menu' ('caption' 'Move')
z←'METSIM.mov.mUp' ⎕wi 'New' 'Menu' ('caption' 'GU') ('onClick' 'GCTR  0  1 ⋄ ∆SYM←1 ⋄
z←'METSIM.mov.mDn' ⎕wi 'New' 'Menu' ('caption' 'GD') ('onClick' 'GCTR  0 ¯1 ⋄ ∆SYM←1 ⋄
z←'METSIM.mov.mLf' ⎕wi 'New' 'Menu' ('caption' 'GL') ('onClick' 'GCTR ¯1 0 ⋄ ∆SYM←1 ⋄
z←'METSIM.mov.mRg' ⎕wi 'New' 'Menu' ('caption' 'GR') ('onClick' 'GCTR   1 0 ⋄ ∆SYM←1 ⋄
z←'METSIM.mov.mPU' ⎕wi 'New' 'Menu' ('caption' 'PU') ('onClick' '2 FSEC 1⌈(ρSEC)[1]⌊Is
z←'METSIM.mov.mPD' ⎕wi 'New' 'Menu' ('caption' 'PD') ('onClick' '2 FSEC 1⌈(ρSEC)[1]⌊Is
z←'METSIM.mov.mHO' ⎕wi 'New' 'Menu' ('caption' 'PH') ('onClick' '2 FSEC 1
z←'METSIM.mov.mEN' ⎕wi 'New' 'Menu' ('caption' 'PE') ('onClick' '2 FSEC   (ρSEC)[1]
wf ⎕wi 'scale' 1 ⋄ ∆scale←(wf ⎕wi 'scale')[4 5] ⋄ wf ⎕wi 'scale' 5
∆size←size←wf ⎕wi 'size'                              ⍝ avail form area
z←'METSIM.mov'     #.A2K.∆WI 'New' 'Menu' ('caption' 'Move')
z←'METSIM.mov.mUp' #.A2K.∆WI 'New' 'Menu' ('caption' 'GU') ('onClick' 'GCTR  0  1 ⋄ ∆S
z←'METSIM.mov.mDn' #.A2K.∆WI 'New' 'Menu' ('caption' 'GD') ('onClick' 'GCTR  0 ¯1 ⋄ ∆S
z←'METSIM.mov.mLf' #.A2K.∆WI 'New' 'Menu' ('caption' 'GL') ('onClick' 'GCTR ¯1 0 ⋄ ∆S
z←'METSIM.mov.mRg' #.A2K.∆WI 'New' 'Menu' ('caption' 'GR') ('onClick' 'GCTR   1 0 ⋄ ∆S
z←'METSIM.mov.mPU' #.A2K.∆WI 'New' 'Menu' ('caption' 'PU') ('onClick' '2 FSEC 1⌈(ρSEC)
z←'METSIM.mov.mPD' #.A2K.∆WI 'New' 'Menu' ('caption' 'PD') ('onClick' '2 FSEC 1⌈(ρSEC)
z←'METSIM.mov.mHO' #.A2K.∆WI 'New' 'Menu' ('caption' 'PH') ('onClick' '2 FSEC 1
z←'METSIM.mov.mEN' #.A2K.∆WI 'New' 'Menu' ('caption' 'PE') ('onClick' '2 FSEC   (ρSEC)
wf #.A2K.∆WI 'scale' 1 ⋄ ∆scale←(wf #.A2K.∆WI 'scale')[4 5] ⋄ wf #.A2K.∆WI 'scale' 5
∆size←size←wf #.A2K.∆WI 'size'                              ⍝ avail form area
```


FLASH SMELTER EXAMPLE
Stream Number
DRYER BURNER
CONCENTRATE DRYER
WASTE HEAT BOILER
ESP NO. 1
ESP NO. 2
AIR PREHEATER
FLASH FURNACE
DUST RECYCLE
OFFGAS COOLER
FURNACE SETTLER
ELECTRIC FURNACE
CONVERTER SLAG BLOW
CONVERTER COPPER BLOW
FLASH FURNACE
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

Demo

∆WI Status September 2024

Some Support

button
check
combo
edit
form
frame
imagelist
label
list
listview
menu
page
picture
richedit
scroll
selector
spinner
timer

No Support

activecontrol
activeobject
commandbar
commandbutton
datetime
grid
listview
mdiform
media
option
printer
progress
status
toolbox
trackbar
tree

Declaration of Intent

- We have hired two new APL developers in 2024
- METSIM® migration complete expected around end of 2024
- Our partners in Germany, USA and Sweden are gaining experience of migrations

Migration Status

- `∆WI` is close to supporting all of METSIM®
  - Additional features will be added to support future migrants
- We will produce a document enumerating differences and documenting emulation functions

- **All migration tools and documentation will be free and open source**

- We *may* also decide to add new features to v20
  - For example `:LeaveIf`