Solingen, November 8th 2024

Migrating APL+Win Applications

Morten Kromberg, CTO

A New Wave of Migrants

- Dyalog APL was created by Dyadic Systems Ltd, when the mainframe consulting business started to die (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

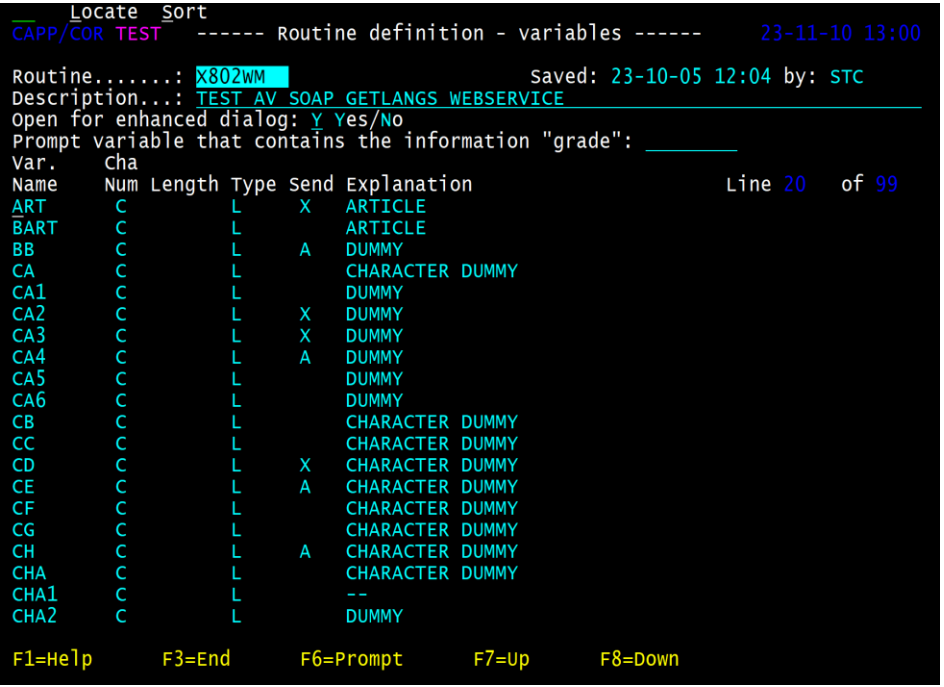
Relatively straightforward

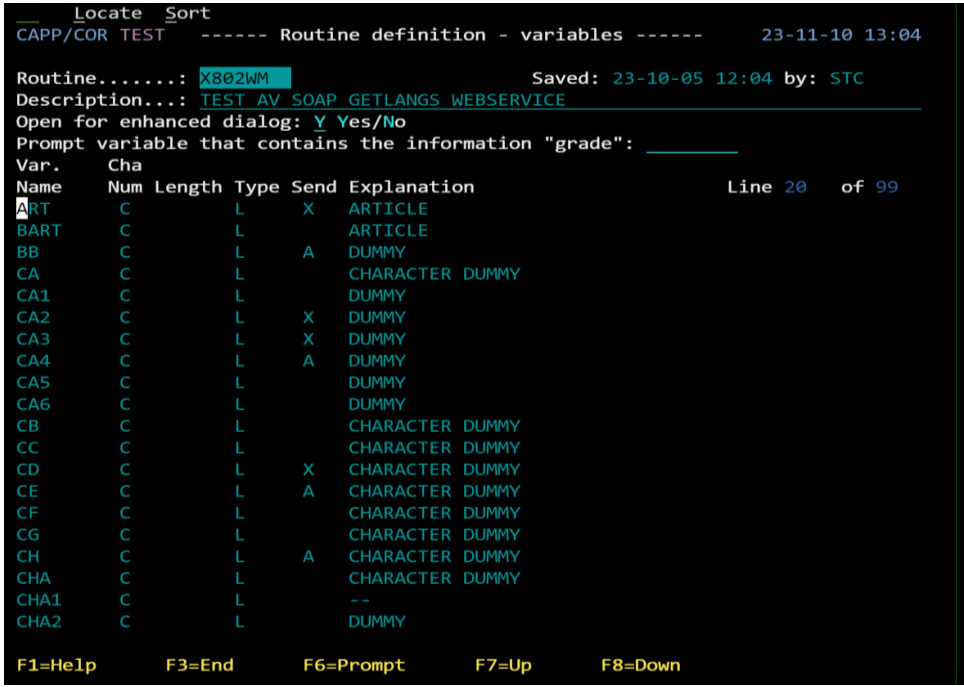
- A few language differences
  - Each (¨) on empty arrays
  - Format by example
- User Interfaces and file I/O are usually already handled by simple cover-functions and can be emulated “easily”
- Linux or Windows apps may be making external DLL calls

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

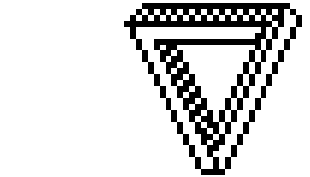


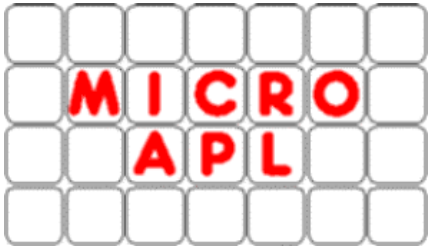


From APL+Win or MicroAPL APLX

Same language differences as APL2, plus:

- System functions & control structures not found in Dyalog APL
- Double quotes (`"Don't do this!"`)
- Graphical User Interfaces





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

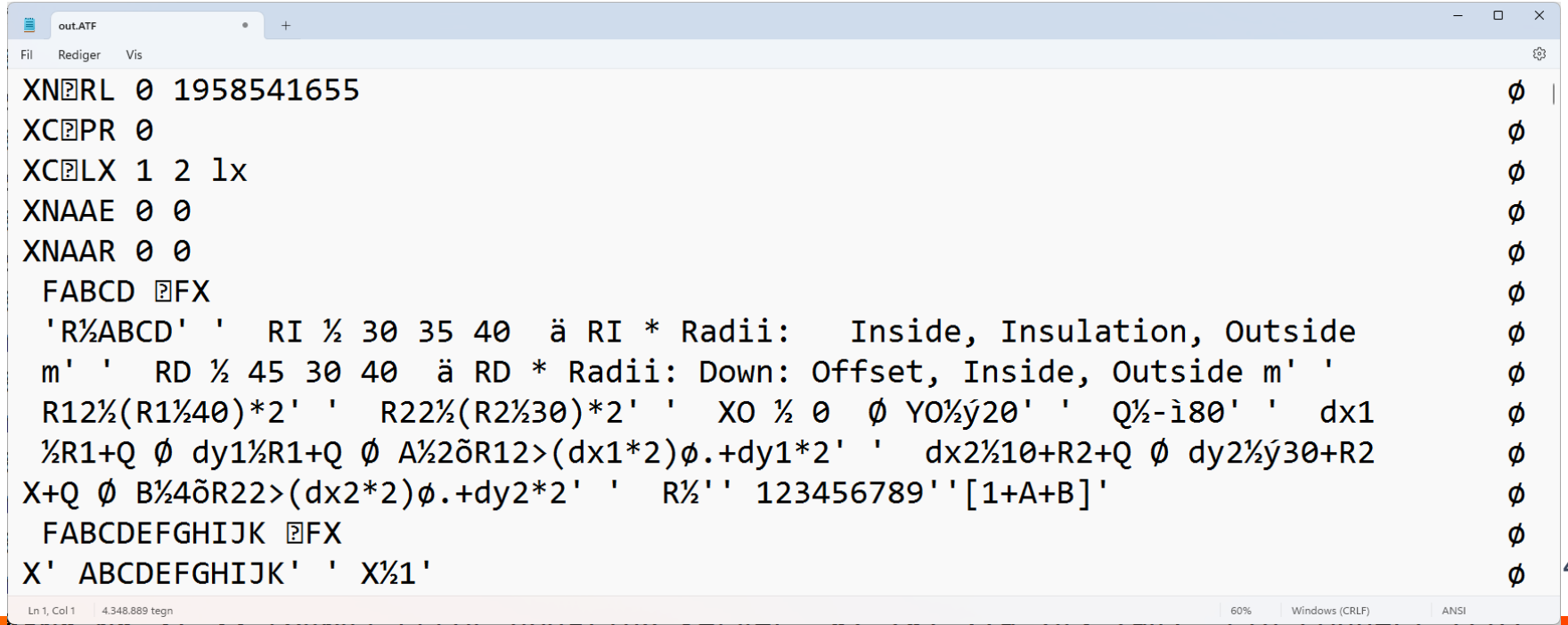
Step 1: Export Source Code

- 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

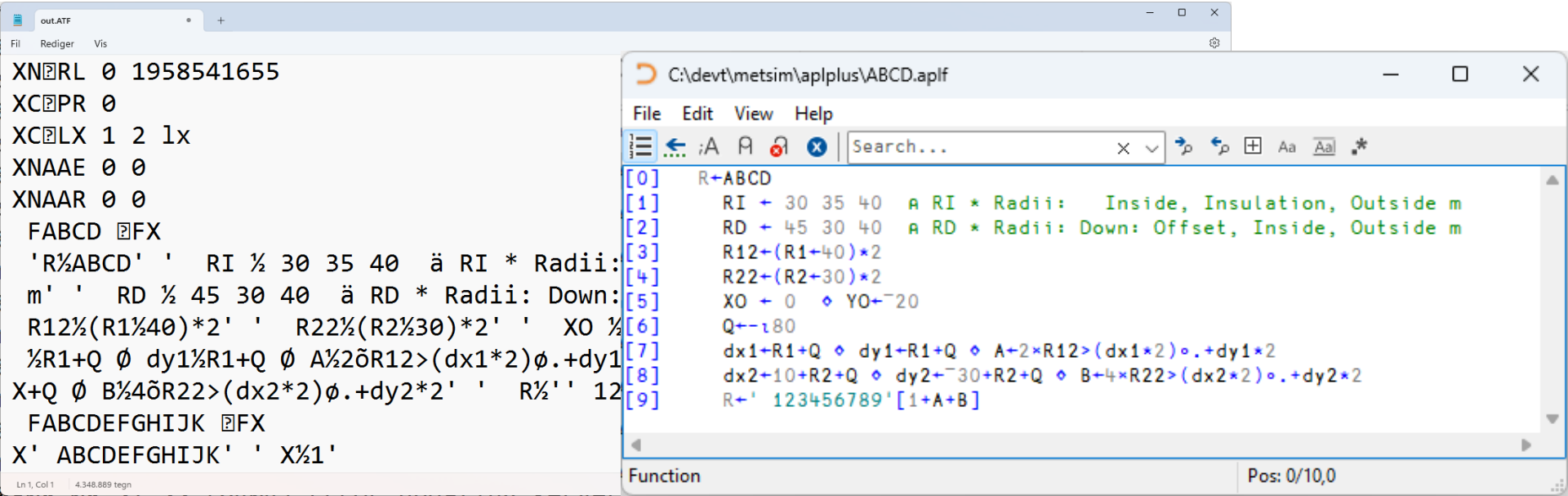
- Result of `]OUT /tmp/out`:



Step 2: Create Text Source

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

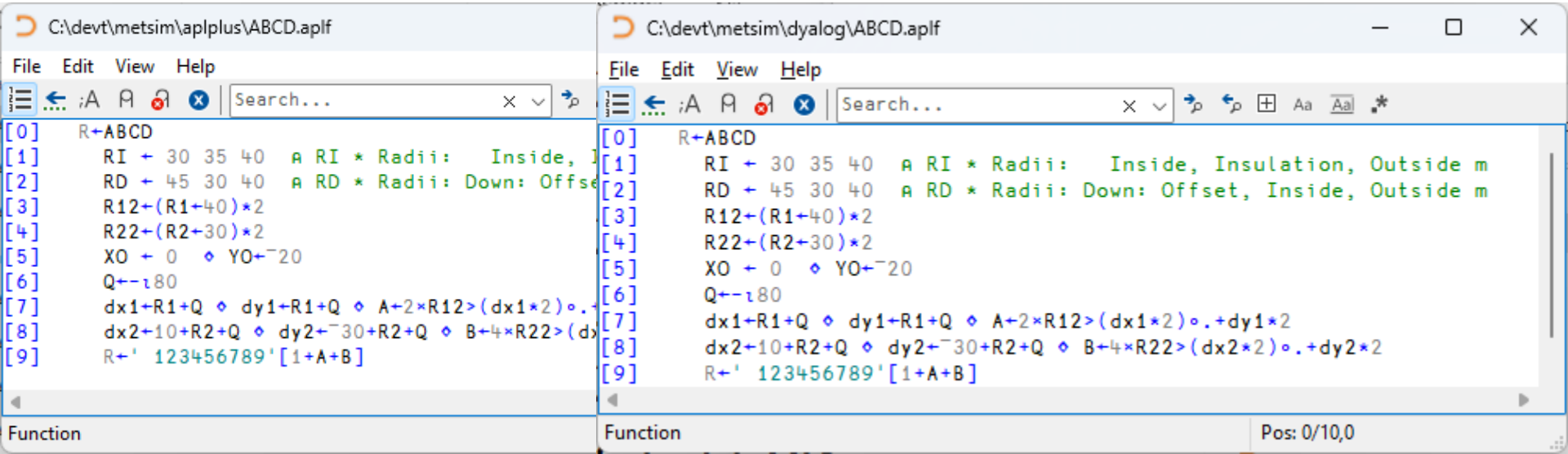
```
]IN /tmp/out.ATF -outdir=/path/aplplus -apl=APLPLUS
```



Step 3: Automatic Conversion

- Apply automated transformations to the original source

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



Automatic Substitutions

```
- FRFX C;FN;I;L;N;X;z;Z;⎕elx
- :IF 2=⎕NC 'ΔMOP'
1+ FRFX C;FN;I;L;N;X;z;Z;ΔQELX
2+ :IF 2=#.A2K.ΔNC 'ΔMOP'
3  :If ΔMOP[10]=0
-  ΔFRFX←'NO FILES FOUND' ⋄ ⎕elx←'→Δ90'
-  →(0∊⍴L←(∨/(⍴X)⍴(,X)⎕SS '.CR')⌿X←⎕XLIB ΔMDL,'FNC')/Δ90
4+ ΔFRFX←'NO FILES FOUND' ⋄ ΔQELX←'→Δ90'
5+ →(0∊⍴L←( 1(∊⍤1)(⍴X)⍴(,X)#.A2K.ΔSS '.CR')⌿X←#.A2K.ΔXLIB ΔMDL,'FNC')/Δ90
6  ΔFRFX←'FILES READ'
7  :FOR I :IN ⍳(⍴L)[1] ⋄ FN←ΔMDL,'FNC\',L[I;]
-    ⎕NUNTIE ¯1 ⋄ FN ⎕XNTIE ¯1 ⋄ Z←⎕NREAD ¯1 82 ,⎕NSIZE ¯1
8+   ⎕NUNTIE ¯1 ⋄ FN ⎕NTIE ¯1 ⋄ Z←#.A2K.ΔNREAD ¯1 82 ,⎕NSIZE ¯1
9    Z←FSTM Z ⋄ Z←((Z[;1]=ΔB)^Z[;2]='A')⌽Z←(^/Z[;1 2]=ΔB)⌽Z←(--∨/Z=':')⌽Z,ΔB
10   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.ΔXLIB

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

Manual Steps

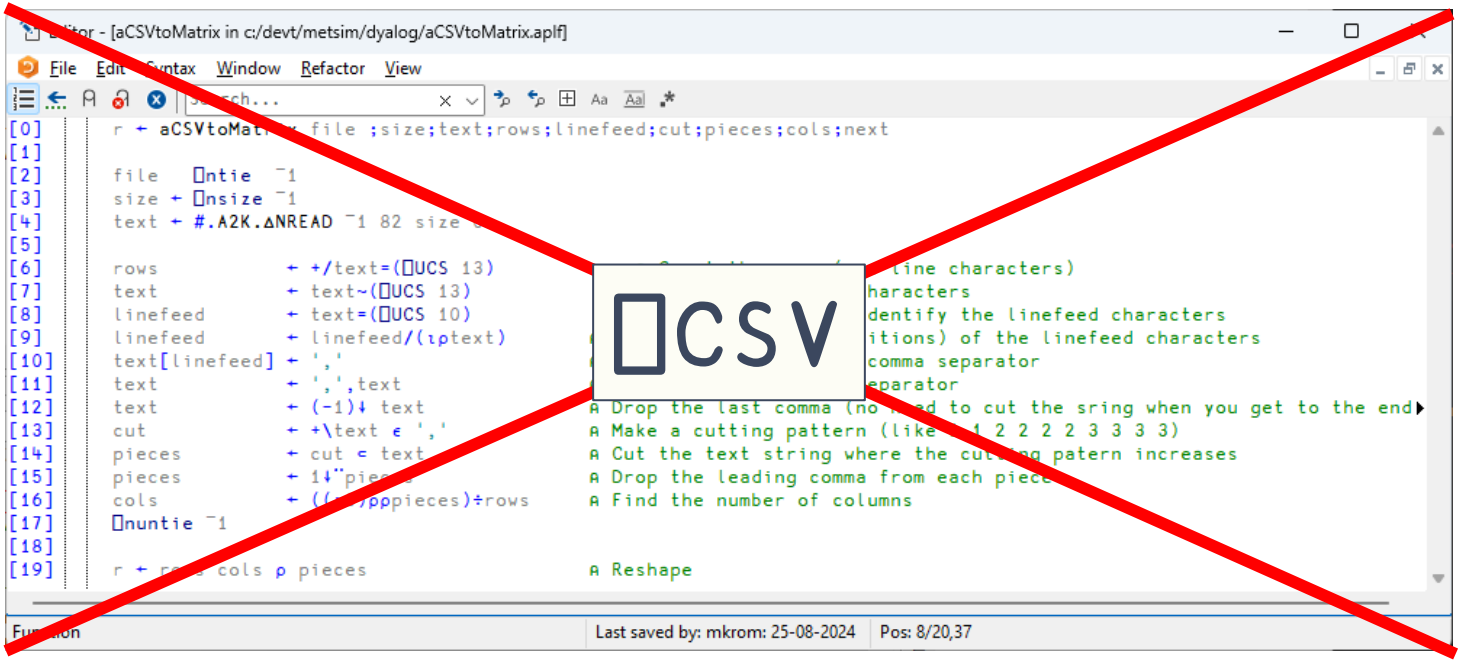
Foreign Function Calls

- Mechanical manual translation

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

- We *may* add automated conversion

CSV files


File Edit Syntax Window Refactor View
r ← aCSVtoMatrix file ;size;text;rows;linefeed;cut;pieces;cols;next
file ⎕ntie ¯1
size ← ⎕nsize ¯1
text ← #.A2K.ΔNREAD ¯1 82 size
rows ← +/text=(⎕UCS 13)
text ← text~(⎕UCS 13)
linefeed ← text=(⎕UCS 10)
linefeed ← linefeed/(⍳⍴text)
text[linefeed] ← ','
text ← ',',text
text ← (¯1)↓ text
cut ← +\text ∈ ','
pieces ← cut ⊂ text
pieces ← 1↓¨pieces
cols ← ((⍴pieces)÷rows
⎕nuntie ¯1
r ← rows cols ⍴ pieces
⎕CSV
Last saved by: mkrom: 25-08-2024
Pos: 8/20,37

```
[0]   r←{x} jsonParse2 y;a;q
[1]   q←maskBetwQuotes y
[2]   a←~q^y∊' ',#.A2K.ΔAV[10],#.A2K.ΔCRLF
[3]   y←a/y ⍝ remove non-quoted whitespace (space, tab, cr, lf)
[4]   q←a/q
[5]   :select
[6]     :case '{'=ty
[7]       y←1↓¯1↓y
[8]       q←1↓¯1↓q
[9]       y←(~q^(q maskBetwBrac
[10]      r←newDict ⊂[1]((0.5×
[11]      r.keys←jsonParse2¨r.k
[12]      :if 0=#.A2K.ΔNC 'x'
[13]        r.values←jsonParse2
[14]      :endif
[15]    :case '['=ty
[16]      y←1↓¯1↓y
[17]      q←1↓¯1↓q
[18]      r←(~q^(q maskBetwBrac
[19]      :if 0=#.A2K.ΔNC 'x'
[20]        r←jsonParse2¨r
[21]      :endif
[22]    :case '"'=ty
[23]      r←1↓¯1↓y
[24]    :case (ty)∊'tf'
[25]      ⍝ true/fal
[26]      r←y
[27]    :el
[28]      ⍝ number
[29]      r←⍎#.A2K.ΔFI y
        :endselect
```

```
[0]   r←{x} jsonParseFast y;a;q
[1]   q←maskBetwQuotes y
[2]   a←~q^y∊' ',⎕UCS 10 13
[3]   y←a/y ⍝ remove non-quoted whitespace (space, tab, cr, lf)
[4]   q←a/q
[5]   :select 1
[6]     :case '{'=ty
[7]       y←1↓¯1↓y
[8]       q←1↓¯1↓q
[9]       y←(~q^(q maskBetwBraces y)^(q maskBetwBrackets y)^y∊',:')⊂y
[10]      r←3⍴⊂''
[11]      (⊃r)←⊂'dict'
[12]      r
[13]      (
[14]      :
[15]
[16]      :
[17]    :ca   '['=ty
[18]      y←1↓¯1↓y
[19]      q←1↓¯1↓q
[20]      r←jsonParseFast¨(~q^(q maskBetwBraces y)^(q maskBetwBrackets y)^y∊',')⊂y
[21]      :if 0=⎕NC 'x'
[22]      :andif ~0(⊃⍸1)(mapType¨r)≡¨⊂⊂'dict'
[23]      :andif ~0(⊃⍸1)2≡/2↑¨r
[24]        r←⊂'table'(2↑⊃r)(⊂[1]↑3↑¨r)
[25]      :endif
[26]    :case '"'=ty
[27]      r←1↓¯1↓y
[28]    :case (ty)∊'tf'
[29]      ⍝ true/false
[30]      r←y
[31]    :else
[32]      ⍝ number
[33]      r←⍎⎕FI y
[34]  :endselect
```

⎕JSON

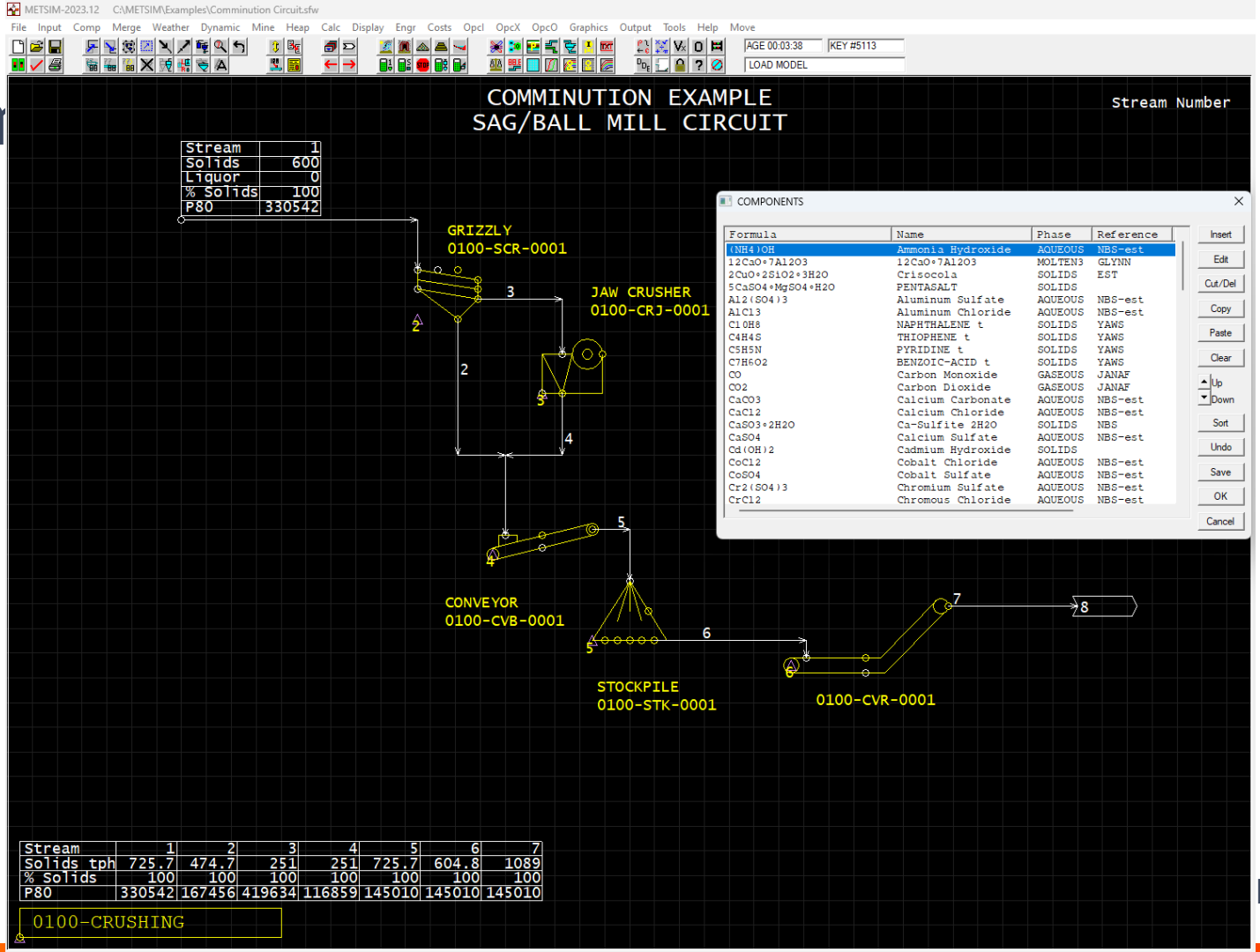
Component Files

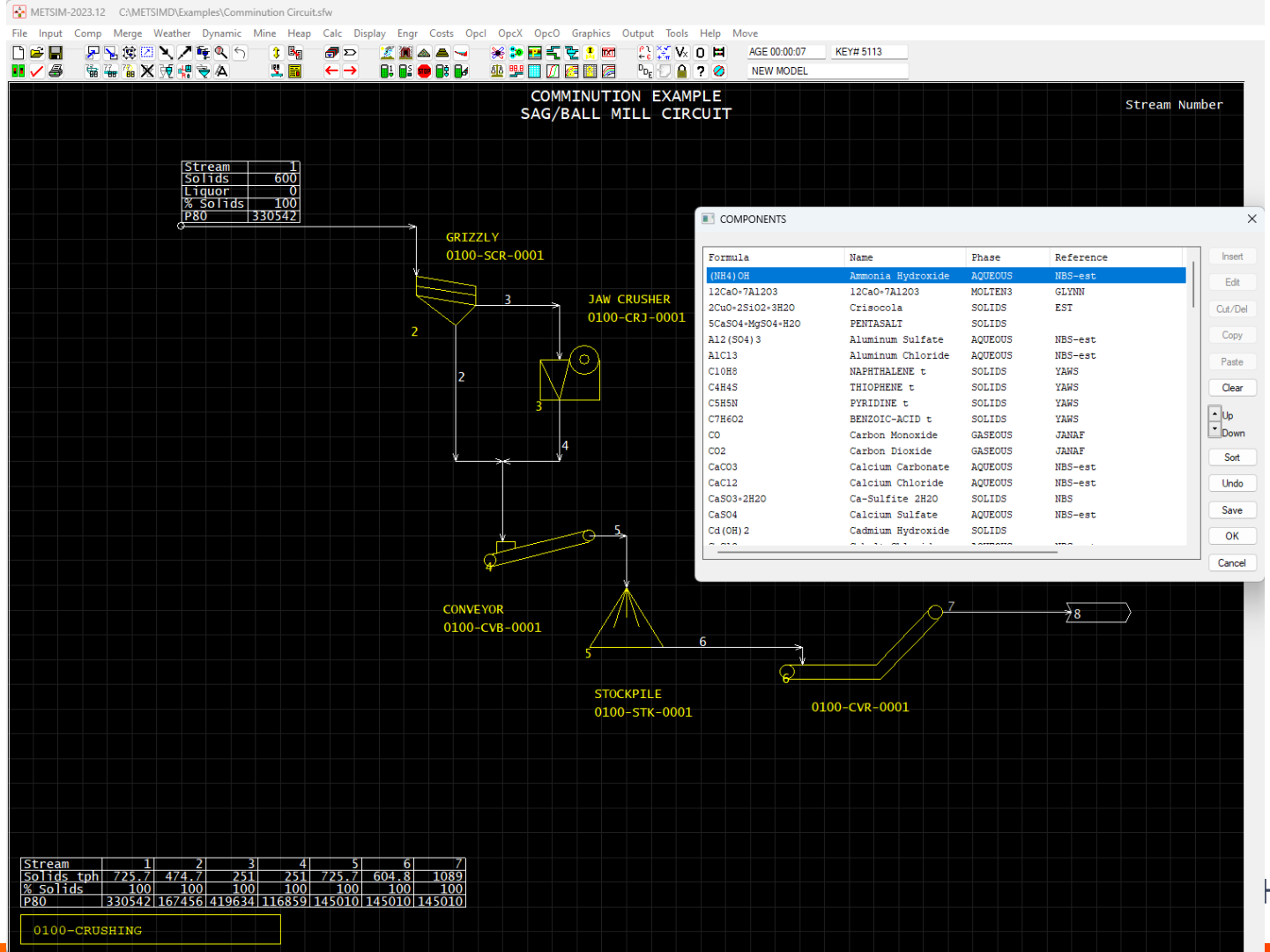
- Emulations of ⎕F * system functions which can read ***and*** write APL+Win component files directly
  - Uses an APL+Win runtime application and binary SCAR format via TCP sockets
- Allows parallel operation of old + new versions of the application code
- Component files can be migrated gradually

The Hard Part

- Graphical User Interfaces

Dem


METSIM-2023.12 C:\METSIM\Examples\Comminution Circuit.sfw
File Input Comp Merge Weather Dynamic Mine Heap Calc Display Engr Costs Opcl OpcX OpcO Graphics Output Tools Help Move
AGE 00:03:38
KEY #5113
LOAD MODEL
COMMINUTION EXAMPLE
SAG/BALL MILL CIRCUIT
Stream Number
Stream 1
Solids 600
Liquor 0
% Solids 100
P80 330542
GRIZZLY
0100-SCR-0001
JAW CRUSHER
0100-CRJ-0001
CONVEYOR
0100-CVB-0001
STOCKPILE
0100-STK-0001
0100-CVR-0001
COMPONENTS
Formula Name Phase Reference
(NH4)OH Ammonia Hydroxide AQUEOUS NBS-est
12CaO•7Al2O3 12CaO•7Al2O3 MOLTEN3 GLYNN
2CuO•2SiO2•3H2O Crisocola SOLIDS EST
5CaSO4•MgSO4•H2O PENTASALT SOLIDS
Al2(SO4)3 Aluminum Sulfate AQUEOUS NBS-est
AlCl3 Aluminum Chloride AQUEOUS NBS-est
C10H8 NAPHTHALENE t SOLIDS YAWS
C4H4S THIOPHENE t SOLIDS YAWS
C5H5N PYRIDINE t SOLIDS YAWS
C7H6O2 BENZOIC-ACID t SOLIDS YAWS
CO Carbon Monoxide GASEOUS JANAF
CO2 Carbon Dioxide GASEOUS JANAF
CaCO3 Calcium Carbonate AQUEOUS NBS-est
CaCl2 Calcium Chloride AQUEOUS NBS-est
CaSO3•2H2O Ca-Sulfite 2H2O SOLIDS NBS
CaSO4 Calcium Sulfate AQUEOUS NBS-est
Cd(OH)2 Cadmium Hydroxide SOLIDS
CoCl2 Cobalt Chloride AQUEOUS NBS-est
CoSO4 Cobalt Sulfate AQUEOUS NBS-est
Cr2(SO4)3 Chromium Sulfate AQUEOUS NBS-est
CrCl2 Chromous Chloride AQUEOUS NBS-est
Insert
Edit
Cut/Del
Copy
Paste
Clear
Up
Down
Sort
Undo
Save
OK
Cancel
Stream 1 2 3 4 5 6 7
Solids tph 725.7 474.7 251 251 725.7 604.8 1089
% Solids 100 100 100 100 100 100 100
P80 330542 167456 419634 116859 145010 145010 145010
0100-CRUSHING


METSIM-2023.12 C:\METSIMD\Examples\Comminution Circuit.sfw
File Input Comp Merge Weather Dynamic Mine Heap Calc Display Engr Costs Opcl OpcX OpcO Graphics Output Tools Help Move
AGE 00:00:07
KEY# 5113
NEW MODEL
COMMINUTION EXAMPLE
SAG/BALL MILL CIRCUIT
Stream Number
Stream 1
Solids 600
Liquor 0
% Solids 100
P80 330542
GRIZZLY
0100-SCR-0001
JAW CRUSHER
0100-CRJ-0001
CONVEYOR
0100-CVB-0001
STOCKPILE
0100-STK-0001
0100-CVR-0001
COMPONENTS
Formula Name Phase Reference
(NH4)OH Ammonia Hydroxide AQUEOUS NBS-est
12CaO·7Al2O3 12CaO·7Al2O3 MOLTEN3 GLYNN
2CuO·2SiO2·3H2O Crisocola SOLIDS EST
5CaSO4·MgSO4·H2O PENTASALT SOLIDS
Al2(SO4)3 Aluminum Sulfate AQUEOUS NBS-est
AlCl3 Aluminum Chloride AQUEOUS NBS-est
C10H8 NAPHTHALENE t SOLIDS YAWS
C4H4S THIOPHENE t SOLIDS YAWS
C5H5N PYRIDINE t SOLIDS YAWS
C7H6O2 BENZOIC-ACID t SOLIDS YAWS
CO Carbon Monoxide GASEOUS JANAF
CO2 Carbon Dioxide GASEOUS JANAF
CaCO3 Calcium Carbonate AQUEOUS NBS-est
CaCl2 Calcium Chloride AQUEOUS NBS-est
CaSO3·2H2O Ca-Sulfite 2H2O SOLIDS NBS
CaSO4 Calcium Sulfate AQUEOUS NBS-est
Cd(OH)2 Cadmium Hydroxide SOLIDS
Insert
Edit
Cut/Del
Copy
Paste
Clear
Up
Down
Sort
Undo
Save
OK
Cancel
Stream 1 2 3 4 5 6 7
Solids tph 725.7 474.7 251 251 725.7 604.8 1089
% Solids 100 100 100 100 100 100 100
P80 330542 167456 419634 116859 145010 145010 145010
0100-CRUSHING

```
[0]     {name} WLVIEWC x;cap;clip;col;csel;cw;d;datd;datn;datp;datx;hed;i;wflv;wz;z;⎕PP
[1]      ΔLVIEW←1 ⋄ (datn col hed cap datp datx)←x       ⍝ Generic Listview Control
[2]      datd←⍴datn ⋄ i←0 ⋄ cw←⍳0 ⋄ csel←3
[3]     ΔL10:i←i+1 ⋄ cw←cw,2+(2↑⍴⍎datn[;col[i]])[2] ⋄ →(i<⍴col)/ΔL10
[4]      :if 2≠#.A2K.ΔNC 'dats' ⋄ dats←,1 ⋄ :end ⋄ clip←(0,⍴datp)⍴datp
[5]      wflv←'LVIEW' ⋄ 2 0 WFORM wflv cap 16 ('extent' 24.5 80)
[6]      wflv ⎕WI 'onShow' ('wflv←''',wflv,''' ⋄ WLVIEWH 0 ⋄ WLVIEWH 1')
[7]      ΔWSELF←wflv ⎕WI '.lv1.New' 'Listview' ⋄ ⎕WI ΔFONTD
[8]      ⎕WI 'where' 1 1 22 70                  ⋄ ⎕WI 'style' 1
[9]      ⎕WI 'onColClick' 'WLVIEWH 10'          ⋄ ⎕WI 'onKeyDown'  'WLVIEWH 11'
[10]     ⎕WI 'onSelect'   'WLVIEWH  7'          ⋄ ⎕WI 'onDblClick' 'WLVIEWH  3'
[11]     z←name datn dats datp datx clip col hed cap ⋄ wflv ⎕WI 'data' z  ⍝ store data
[12]     z←0 4⍴ΔB ⋄ i←0                                                  ⍝ set column headings
[13]    Δ30:i←i+1 ⋄ z←z,[1] (⍎hed[i]) (5⌈cw[i]⌈3+⍴⍎hed[i]) 'left' (i) ⋄ →(i<⍴col)/Δ30
[14]     ⎕WI 'columndisplay' z
[15]     WLVIEWH 1 ⋄ wz←1 72 1.35 7
[16]     (wz←wz+0    0 0 0) WBUTN  2 'Insert'  'WLVIEWH  2' ('enabled' 0) ⍝..insert/add
[17]     (wz←wz+1.85 0 0 0) WBUTN  3 'Edit'    'WLVIEWH  3' ('enabled' 0) ⍝..edit
[18]     (wz←wz+1.85 0 0 0) WBUTN  4 'Cut/Del' 'WLVIEWH  4' ('enabled' 0) ⍝..cut/del
[19]     (wz←wz+1.85 0 0 0) WBUTN  5 'Copy'    'WLVIEWH  5' ('enabled' 0) ⍝..copy
[20]     (wz←wz+1.85 0 0 0) WBUTN  6 'Paste'   'WLVIEWH  6' ('enabled' 0) ⍝..paste
[21]    ⍝                                      'WLVIEWH  7'               ⍝..select
[22]     (wz←wz+1.85 0 0 0) WBUTN  8 'Clear'   'WLVIEWH  8'               ⍝..clear
[23]     (wz←wz+1.85 0 1 ¯5)WSPIN  (0 1000)    'WLVIEWH  9' 'Up' 'Down' 'Move items up or down'
[24]        (wflv,'.spin1') ⎕WI 'enabled' 0
[25]     (wz←wz+3    0 ¯1 5) WBUTN 10 'Sort'   'WLVIEWH 10'               ⍝..sort
[26]     (wz←wz+1.85 0 0 0) WBUTN 12 'Undo'    'WLVIEWH 12'               ⍝..undo
[27]     (wz←wz+1.85 0 0 0) WBUTN 13 'Save'    'WLVIEWH 13'               ⍝..save
[28]     (wz←wz+1.85 0 0 0) WBUTN 14 'OK'      'WLVIEWH 14'               ⍝..OK
[29]     (wz←wz+1.85 0 0 0) WBUTN 15 'Cancel'  'WLVIEWH 15'               ⍝..cancel
[30]     z←wflv ⎕WI 'Wait'
```

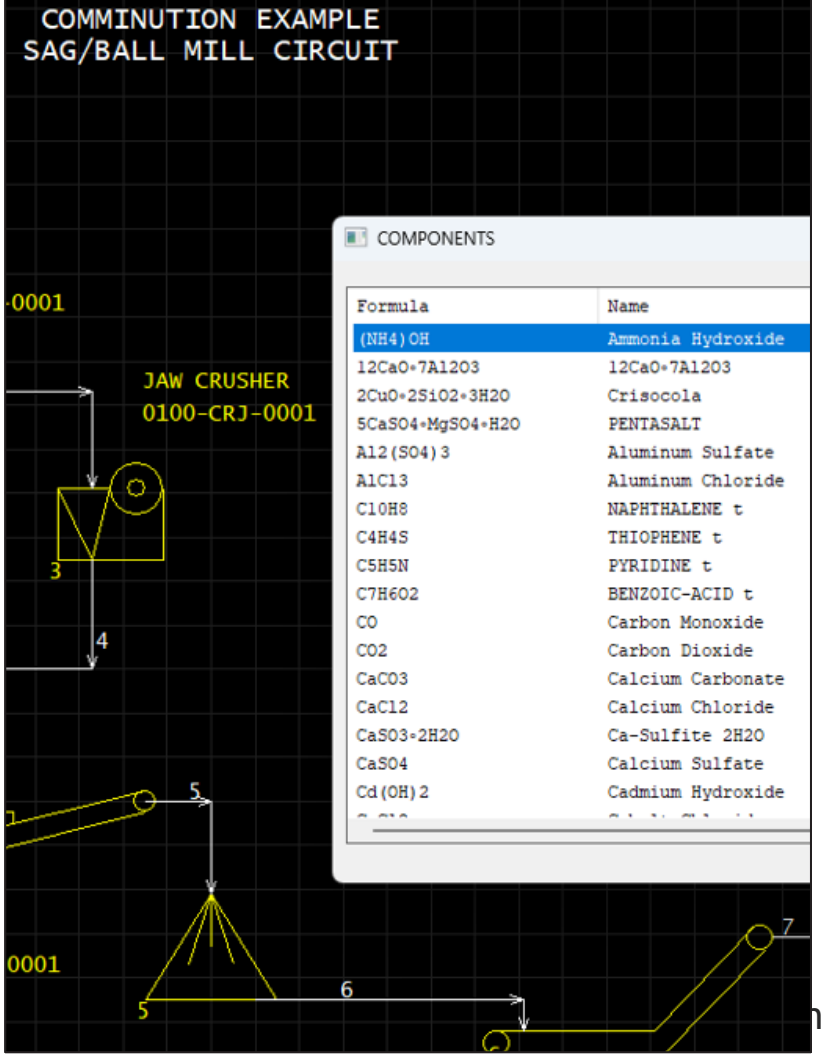
```
[0]     {name} WLVIEWC x;cap;clip;col;csel;cw;d;datd;datn;datp;datx;hed;i;wflv;wz;z;⎕PP
[1]      ΔLVIEW←1 ⋄ (datn col hed cap datp datx)←x      ⍝ Generic Listview Control
[2]      datd←⍴datn ⋄ i←0 ⋄ cw←⍳0 ⋄ csel←3
[3]     ΔL10:i←i+1 ⋄ cw←cw,2+(2↑⍴⍕datn[;col[i]])[2] ⋄ →(i<⍴col)/ΔL10
[4]      :if 2≠##.A2K.ΔNC 'dats' ⋄ dats←,1 ⋄ :end ⋄ clip←(0,⍴datp)⍴datp
[5]      wflv←'LVIEW' ⋄ 2 0 WFORM wflv cap 16 ('extent' 24.5 80)
[6]      wflv ΔWI 'onShow' ('wflv←''',wflv,''' ⋄ WLVIEWH 0 ⋄ WLVIEWH 1')
[7]      ΔWSELF←wflv ΔWI '.lv1.New' 'Listview' ⋄ ΔWI ΔFONTD
[8]      ΔWI 'where' 1 1 22 70                 ⋄ ΔWI 'style' 1
[9]      ΔWI 'onColClick' 'WLVIEWH 10'         ⋄ ΔWI 'onKe
[10]     ΔWI 'onSelect'   'WLVIEWH  7'         ⋄ ΔWI 'onDb
[11]     z←name datn dats datp datx clip col hed cap ⋄ wfl
[12]     z←0 4⍴ΔB ⋄ i←0
[13]    Δ30:i←i+1 ⋄ z←z,[1] (⊂hed[i]) (5⌈cw[i]⌈3+⍴⍕hed[i])
[14]     ΔWI 'columndisplay' z
[15]     WLVIEWH 1 ⋄ wz←1 72 1.35 7
[16]     (wz←wz+0     0 0 0) WBUTN  2 'Insert'   'WLVIEWH  2
[17]     (wz←wz+1.85 0 0 0) WBUTN  3 'Edit'     'WLVIEWH  3
[18]     (wz←wz+1.85 0 0 0) WBUTN  4 'Cut/Del'  'WLVIEWH  4
[19]     (wz←wz+1.85 0 0 0) WBUTN  5 'Copy'     'WLVIEWH  5
[20]     (wz←wz+1.85 0 0 0) WBUTN  6 'Paste'    'WLVIEWH  6
[21]    ⍝                                      'WLVIEWH  7
[22]     (wz←wz+1.85 0 0 0) WBUTN  8 'Clear'    'WLVIEWH  8
[23]     (wz←wz+1.85 0 1 ¯5)WSPIN  (0 1000)     'WLVIEWH  9
[24]        (wflv,'.spin1') ΔWI 'enabled' 0
[25]     (wz←wz+3    0 ¯1 5) WBUTN 10 'Sort'    'WLVIEWH 10
[26]     (wz←wz+1.85 0 0 0) WBUTN 12 'Undo'     'WLVIEWH 12
[27]     (wz←wz+1.85 0 0 0) WBUTN 13 'Save'     'WLVIEWH 13
[28]     (wz←wz+1.85 0 0 0) WBUTN 14 'OK'       'WLVIEWH 14
[29]     (wz←wz+1.85 0 0 0) WBUTN 15 'Cancel'   'WLVIEWH 15
[30]     z←wflv ΔWI 'Wait'
```

COMPONENTS

Formula	Name	Phase	Reference
(NH4)OH	Ammonia Hydroxide	AQUEOUS	NBS-est
12CaO·7Al2O3	12CaO·7Al2O3	MOLTEN3	GLYNN
2CuO·2SiO2·3H2O	Crisocola	SOLIDS	EST
5CaSO4·MgSO4·H2O	PENTASALT	SOLIDS	
Al2(SO4)3	Aluminum Sulfate	AQUEOUS	NBS-est
AlCl3	Aluminum Chloride	AQUEOUS	NBS-est
C10H8	NAPHTHALENE t	SOLIDS	YAWS
C4H4S	THIOPHENE t	SOLIDS	YAWS
C5H5N	PYRIDINE t	SOLIDS	YAWS
C7H6O2	BENZOIC-ACID t	SOLIDS	YAWS
CO	Carbon Monoxide	GASEOUS	JANAF
CO2	Carbon Dioxide	GASEOUS	JANAF
CaCO3	Calcium Carbonate	AQUEOUS	NBS-est
CaCl2	Calcium Chloride	AQUEOUS	NBS-est
CaSO3·2H2O	Ca-Sulfite 2H2O	SOLIDS	NBS
CaSO4	Calcium Sulfate	AQUEOUS	NBS-est
Cd(OH)2	Cadmium Hydroxide	SOLIDS	

Insert
Edit
Cut/Del
Copy
Paste
Clear
Up
Down
Sort
Undo
Save
OK
Cancel

```
[0]     {wz} WBUTN X;wn;wc;wa
[1]     (wn wc wa)←3↑X ⋄ wl←wf,'.bn',⍕wn ⋄ wl ΔWI 'Delete'
[2]     wl←wl ΔWI 'New' 'Button' ('where' wz)('caption' wc)('onClick' wa),3↓X
```


COMMINUTION EXAMPLE
SAG/BALL MILL CIRCUIT
-0001
JAW CRUSHER
0100-CRJ-0001
3
4
5
6
7
0001
COMPONENTS
Formula
Name
(NH4)OH
Ammonia Hydroxide
12CaO•7Al2O3
12CaO•7Al2O3
2CuO•2SiO2•3H2O
Crisocola
5CaSO4•MgSO4•H2O
PENTASALT
Al2(SO4)3
Aluminum Sulfate
AlCl3
Aluminum Chloride
C10H8
NAPHTHALENE t
C4H4S
THIOPHENE t
C5H5N
PYRIDINE t
C7H6O2
BENZOIC-ACID t
CO
Carbon Monoxide
CO2
Carbon Dioxide
CaCO3
Calcium Carbonate
CaCl2
Calcium Chloride
CaSO3•2H2O
Ca-Sulfite 2H2O
CaSO4
Calcium Sulfate
Cd(OH)2
Cadmium Hydroxide

⎕WI Status November 2024

Some Support

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

No Support (yet)

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

* listview since Dyalog'24

Declaration of Intent

- We have hired two new APL developers in 2024
- METSIM® migration complete expected early 2025
- Our partners in Germany, USA and Sweden are gaining experience of migrations

Declaration of Intent

- 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 Dyalog v20.0
  - For example `:LeaveIf`

EWC + ΔWI =

- From APL+Win direct to the Cloud