Dyalog APL in Non-Windows Environments

Miscellaneous features etc which you might come across when using Dyalog APL on non-Windows platforms

Andy Shiers, COO Dyalog Ltd

Supported Platforms

- Windows: 32- and 64-bit, Classic and Unicode
- Linux: 64-bit, Classic and Unicode
- macOS: 64-bit Unicode
- Raspberry Pi: 32-bit Unicode on 32-bit Raspbian

- AIX: 32- and 64-bit, Classic and Unicode

- 64-bit Unicode for Windows, Linux and macOS, and 32-bit Unicode for Raspberry Pi can be downloaded from www.dyalog.com

Major differences

- 32-/64-bit: size of array/workspace, interaction with O/S
- Classic/Unicode: Number of supported characters, font limitations etc
- Windows vs the rest: full IDE vs tty interface
  - RIDE can be used to drive recent Dyalogs on any platform
- AIX vs the rest: Big Endian vs Little Endian

- Not a huge difference between AIX/Linux&Pi/macOS
  - But don't assume that they are always the same !

Major differences – workspace

- Workspaces and component files should be compatible across all platforms
  - Within the same version
    - Are upwardly compatible
    - Classic generates TRANSLATION ERRORs if a workspace contains characters which are not in □AVU
      - Watch out for such characters in □DM and □DMX !
    - GUI features will be deleted when loading a Windows workspace into a non-Windows version
      - Will get warning messages .. ignore them!
- WSEXT/CFEXT control what extensions workspace and component file have (if any)

Major differences – workspace and array size

- Workspace size (MAXWS)
  - Maximum size to allow; must be contiguous in the process space
  - 32-bit .. Maximum is between 1.2 to <2.0GB
  - 64-bit .. "unlimited" .. biggest we know of is 2.1TB
  - Don't make too big or O/S will start to swap which really slows things down
    - Consider size of workspace × number of concurrent instances

- An array must fit in workspace:
  - 32-bit can have up to ¯1+2*32 elements
  - 64-bit "unlimited"

Major differences – Classic/Unicode

- Classic has at most 256 characters
  - Limited further by font considerations
  - Much more flexibility with keyboard setup (APLK/APLKEYS etc)
- Unicode has considerably more
  - 1, 2 or 4 byte values

- Even in Classic better to use □UCS, not □AV/□TC

Big Endian/Little Endian

- AIX runs on POWER CPU in big-endian mode (the original mode)
- All other supported platforms are little-endian
  - Be aware that you may have to consider byte ordering
  - Intel/AMD/ARM: Little endian – 1 2 3 4
  - POWER (AIX): Big endian – 4 3 2 1
  - eg 11 □DR 123456

- Dyalog APL does most of the work for you
  - Always)SAVEs in current endianness (!)
  - Components written in local endianness

Major differences – IDE vs tty

- Windows IDE
  - GUI based, floating resizable windows
- tty interface
  - Assumes fixed sized, character based output
    - Responds to some resize requests – might need screen refresh
- RIDE
  - Client for all but AIX
  - Can talk to any Dyalog after 16.0
  - Zero-footprint RIDE (runs in browser)
    - included with all non-Windows versions of Dyalog

Auxillary Processors

- Example: 'xutils' sh''
  - Separate process which contains user-written c-code
  - Appears in workspace as locked functions
  - Erasing all the functions terminates the AP
  - Benefit: if an AP crashes, it doesn't bring the APL down too
    - But: data has to be passed to and from the AP .. Slow

- xutils
- nfiles
- unixfiles
- Customer written APs

□SM

- A screen manager
  - Available on Windows, but only to support existing applications
  - Available on non-Windows, tty interface only, only for existing applications

  - Assign/change □SM to update the screen
    - A complex nested array which can contain sub-screen manager arrays
  - Use □SR to pass control to user (and make □SM screen visible)
  - Use HK keystroke to toggle □SM screen when developing

UNIX cheat sheet

- Big difference with Windows:
  - Windows: erase *.*
    - The erase command sees "*.*" so can ask if you really meant to do it.
  - UNIX: rm *
    - The command shell expands "*" to all that matches it (ie everything). So rm has no idea that it's everything, so won't check
    - But: rm –i
  - Similar for all commands. Few have the interactive option.

UNIX cheat sheet - directories

- cd
  - cd — go home
  - cd - — go to previous directory
  - cd adir — cd to adir

- mkdir adir
- rmdir adir — delete iff it's empty
- rm –r adir — recursively delete (be careful!)
- rm –rf adir — forcibly delete recursively (be exceedingly careful!)

UNIX cheat sheet –dealing with files

- ls
  - ls –lrt long listing in reverse time order – current directory
  - ls f1 f2
  - ls *
  - ls adir

- rm
  - rm f1 delete f1
  - rm * delete all files in current directory

UNIX cheat sheet – dealing with file content

- cat file
  - cat * — concatenate all files
  - cat andy* > out — concatenate all files whose name starts with andy into a file called out
- wc –l file — report number of lines
- grep andys /etc/passwd — report all lines which contain "andys" in the file /etc/passwd
- vi myfile — Use the standard UNIX editor to edit/look at myfile

UNIX cheat sheet – the file command

- file myfile
  - Attempt to describe the contents of the file
- file –m /opt/mdyalog/17.1/64/classic/p9/magic myfile
  - As above, but try to identify details of Dyalog files (the –m only needed on AIX)

```
andys@p9-72qa:/devt/tmp/andys$ file -m /opt/mdyalog/17.1/64/classic/p9/magic *
182.dws: Dyalog APL workspace type 18 subtype 33 64-bit unicode little-endian
big.dws: Dyalog APL workspace type 17 subtype 16 64-bit classic big-endian
isfile: Dyalog APL workspace type 12 subtype 5 64-bit classic big-endian
isfile121: Dyalog APL workspace type 12 subtype 5 64-bit classic big-endian
runall: c program text  File contains 8 bit characters.
types.dws: Dyalog APL workspace type 18 subtype 4 64-bit unicode big-endian
xx.dcf: Dyalog APL component file 64-bit level 1 journaled checksummed
```

UNIX cheat sheet – filesystems

- Windows "sees" separate drives, each with its own filesystem
- UNIX "sees" a single directory structure, made up of one or more filesystems
  - df # report space available in each filesystem
  - df –m # report in MB

  - du myfile # report space used by myfile
  - du –ms * # report space used by each object in current directory

  - du and df both based on number of blocks used. df is more precise

UNIX cheat sheet – pipes, redirection, exit codes

- mycmd <infile >outfile 2>errfile
  - 0: stdin (usually the keyboard)
  - 1: stdout – where output goes (usually the terminal)
  - 2: stderr – where error messages go (usually the terminal)
- mycmd <infile >outfile 2>&1
  - Redirect stderr to wherever stdout is currently redirected to
- grep dyalog /etc/passwd | wc –l
  - identify all lines in /etc/passwd which have dyalog in them, and return how many that is
- echo $?
  - Print the exit code from the last command run. 0 indicates success, anything else failure of some sort

UNIX cheat sheet – processes

- ps
  - ps –ef list all processes with useful information about each
  - ps –elf list full information about all processes
  - ps –fu andys list all the processes belonging to andys
- top/topas/nmon (AIX)
  - Show most active processes. Ctrl-c stops this
- kill
  - kill -2 pid signal a weak interrupt to process with process id pid
  - kill -3 pid signal a strong interrupt to process with process id pid
  - kill -9 pid unconditionally and immediately kill process with process id pid
    - This should be a last resort, not a first resort !

UNIX cheat sheet – shell/environment variables

```
ANDY=1              # defines a shell variable, only visible to this shell
export ANDY         # is now available to all commands started from this
                    # shell.  An Environment Variable
export ANDY=1       # both the above in one command


echo $ANDY          # Must put a $ in front of a variable to reference it
1
```

UNIX cheat sheet – watch out for quotes

```
echo "echo $ANDY"        # Substitute value of variable
echo 1

echo 'echo $ANDY'        # exactly as was typed
echo $ANDY

echo `echo $ANDY`        # treat as though a command
1
```

UNIX cheat sheet – common environment vars

$HOME # my home directory

$PATH # list of directories to look for commands in

$LANG # defines your language environment

$TZ # defines your time zone

$TERM # defines what terminal type you are using

- Normally has the value xterm
- Get this wrong and you may have an unusable terminal session

□sh and)sh

- Run a command from within APL
  - AIX: !!! time taken to call □SH proportional to size of calling process !!!
    - Avoid using □SH whereever possible

```
      )sh echo $PPID
123456
      □sh 'grep 620 /etc/passwd'
jenkins:!:6203:6116::/home/jenkins:/usr/bin/bash
nicolas:!:6204:6116::/home/nicolas:/usr/bin/bash
      +□sh 'grep 620 /etc/passwd'
 jenkins:!:6203:6116::/home/jenkins:/usr/bin/bash nicolas:!:6204:6116::
/home/nicolas:/usr/bin/bash
```

- Quite intentional !

When it goes horribly wrong

- Dyalog APL can "crash"
  - Be careful when using the word crash
    - Do you mean the APL code has errored, but you're still in APL ?
    - Do you mean the APL interpreter has crashed ?

- Types of abnormal termination
  - Syserror - generates an aplcore
  - Core dump - generates a core file if O/S is configured to do so
  - Process dump - need to configure Windows to do so
  - Silently disappears
    - Are you sure the APL code didn't call ⎕OFF ?

When it goes horribly wrong

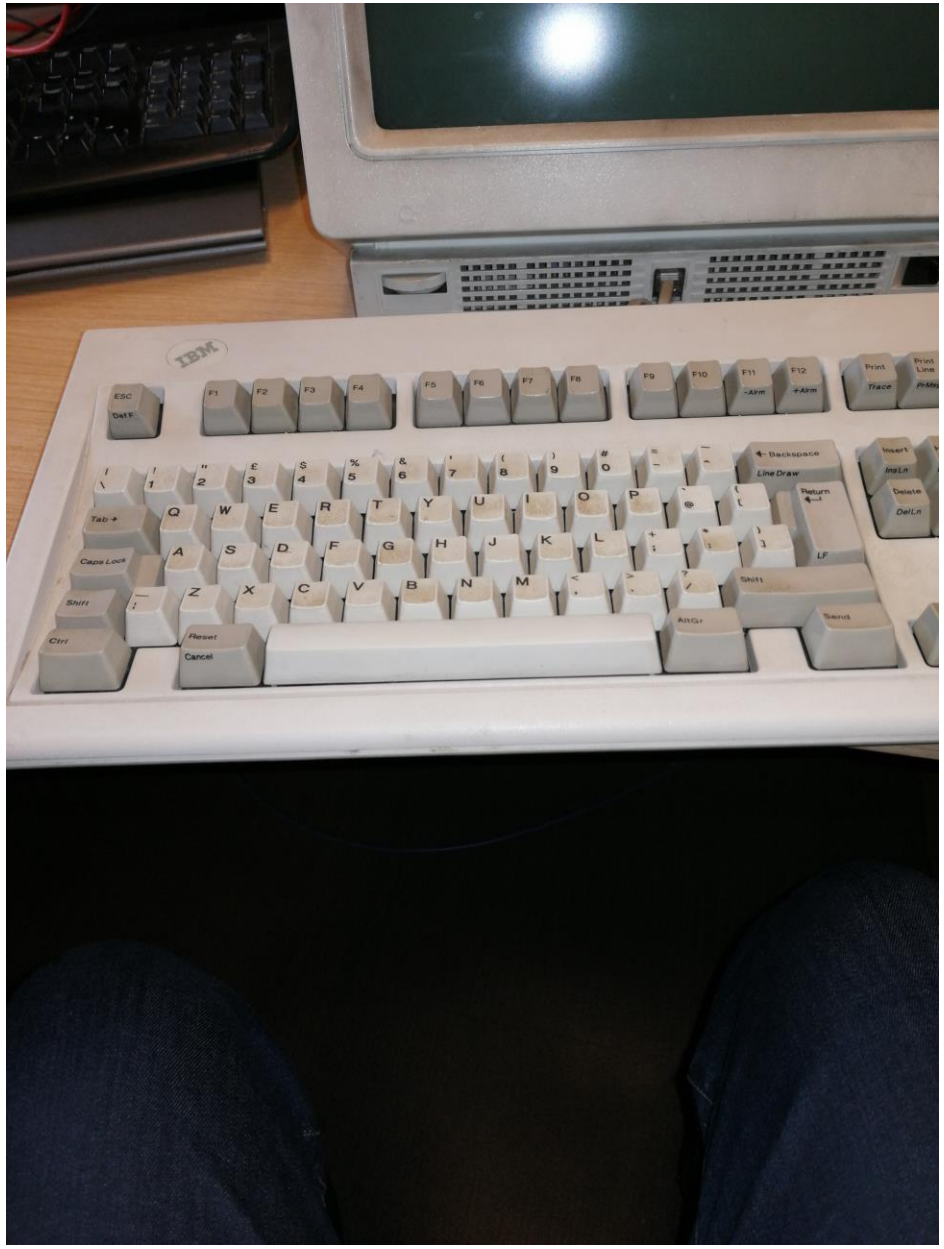
- Your tty session may be unusable
  - Close it down and open a new one
  - Ctrl-j stty sane ctrl-j may get control back

- With a syserror you may be able to see where the problem lies with

  sed -n '/======== Interesting Information/,$p' aplcore

- You may be able to)copy or ⎕cy from an aplcore
  -)copy copies global value, ⎕CY the most local

IBM 3151 ASCII terminal


IBM
ESC
Def F
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Print
Trace
Backspace
Line Draw
Insert
Ins Ln
Delete
Del Ln
Return
LF
Tab
Caps Lock
Shift
Ctrl
Reset
Cancel
AltGr
Send


ESC
Def F
F1
F2
F3
F4
Tab
Caps Lock
Shift
Ctrl
Reset
Cancel

Terminal Emulator

- Windows
  - PuTTY
    - Needs IME and APL-385 font
  - KEA Attachmate
    - Customer configured keyboard
- Linux
  - Xterm and other terminal windows
    - Generally Unicode

Classic issues

- Can't use ⊆ ⌸ ⌹ ⍠ ⍤ ⍥ ⍳
  - No slots free in □av (nor some fonts)
  - Write cover functions/operators and use them

  - Many classic workspaces still use the underscored alphabet
    - Problems rendering them in terminal emulators
    - Lack of keystrokes available
    - May not paste well between APL and other applications

Default keyboard layouts

- Classic / Windows
  - Various different layouts over the years, last updated with 12.1
- Unicode / Windows
  - Uses IME
- Unicode/Linux terminal emulators
  - Supplied with recent Linux distributions
- Classic / PuTTY
  - Uses the Unicode layout & IME
- Classic/KEA
  - Ctrl-o Ctrl-n, underscored alphabet

Keyboard commands

- Listed in UNIX User Guide, Appendix A
  - http://help.dyalog.com/17.1/#UNIX_IUG/Appendix%20A%20Table%20of%20keycodes-te.htm
- Classic: use □KL to see what the keystroke is
  - Need to know the 2 character command mnemonic
  - Can have multiple keystrokes for one command
  - Will be in 18.2 for Unicode

Configuration parameters (Dyalog env vars)

- Are environment variables or command line parameters
- On UNIX MUST be in uppercase
- From 17.1 can appear in config files in $HOME/.dyalog

```
MAXWS=1G mapl          # Start APL with a 1G workspace
APLK=kea_apl mapl      # Set my input translate table to kea_apl
```

APLK/APLKEYS, APLT/APLTRANS

- Classic only
- APLK/APLKEYS
  - Defines what keystroke is used to enter each character
  - Defines what keystroke is used to enter each keyboard command
- APLT/APLTRANS
  - Defines what the interpreter must send to generate each character
  - Defines how to generate colours

  - APLKEYS/APLTRANS are similar to PATH: a list of directories to look in
  - APLK/APLK0, APLT/APLT1/APLT2 is the name of the file to use
    - Generally is the same as the value of $TERM

Useful URLs

- help.dyalog.com/17.1 and docs.dyalog.com/17.1
  - UNIX-specific documentation
- www.dyalog.com/user-meetings/index.htm
  - Uncle Andy's Fireside chats (and The Doctor is in)

- www.ibm.com/docs/en/aix/7.2?topic=commands
  - On most UNIX boxes,
    - command -?
    - command –h
    - man command