

DYALOG



APL Germany

New User Commands in Dyalog 18.2 and Ideas for the Future

Adám Brudzewsky



Overview

`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

+ Ideas for the Future



]OUTPUT.View



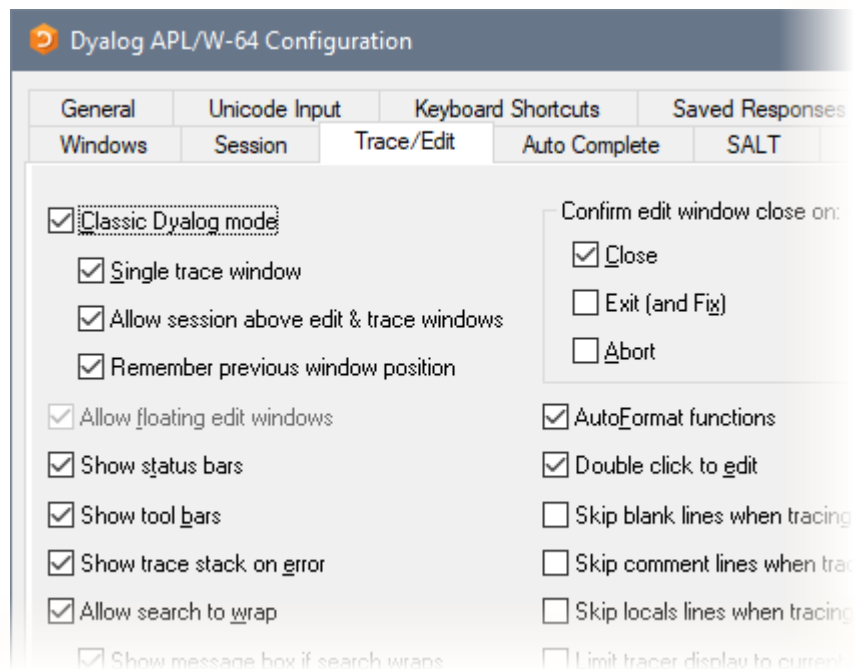
]OUTPUT.View

- ◆ Classic window mode



]OUTPUT.View

Classic window mode



]OUTPUT.View

- ◆ Classic window mode
- ◆ Inspect long output



]OUTPUT.View

- Classic window mode
- Inspect long output

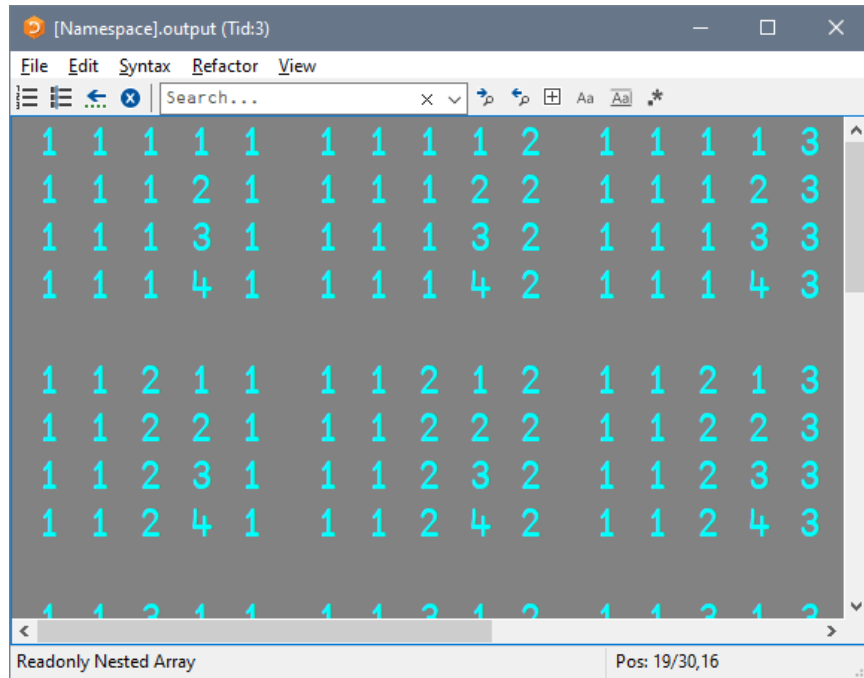
```
]view 115
```



]OUTPUT.View

- Classic window mode
- Inspect long output

]view 115



]OUTPUT.View

- ❖ Classic window mode
- ❖ Inspect long output
- ❖ Snapshot an APL item



]OUTPUT.View

- ❖ Classic window mode
- ❖ Inspect long output
- ❖ Snapshot an APL item

```
var←'oldvalue'
```



]OUTPUT.View

- ❖ Classic window mode
- ❖ Inspect long output
- ❖ Snapshot an APL item

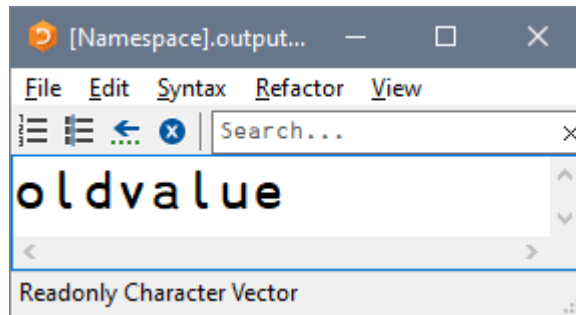
```
var←'oldvalue'  
]view var
```



]OUTPUT.View

- Classic window mode
- Inspect long output
- Snapshot an APL item

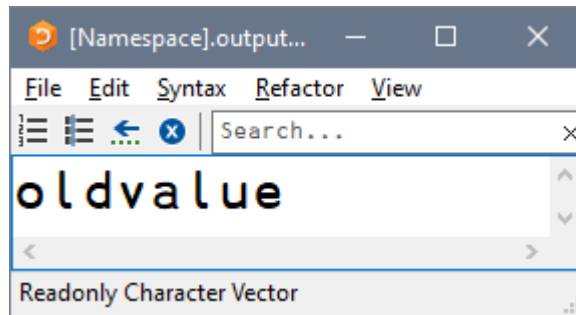
```
var←'oldvalue'  
]view var
```



]OUTPUT.View

- Classic window mode
- Inspect long output
- Snapshot an APL item

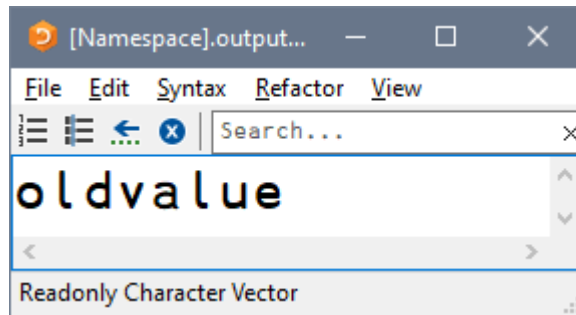
```
var←'oldvalue'  
]view var  
(3↑var)←'new'
```



]OUTPUT.View

- Classic window mode
- Inspect long output
- Snapshot an APL item

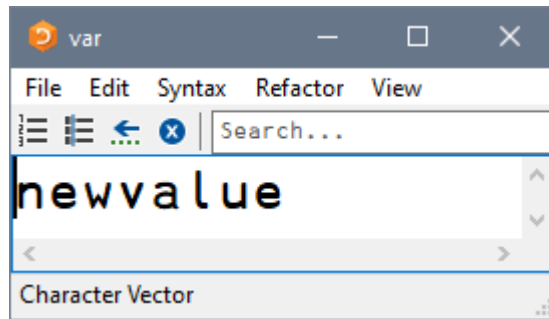
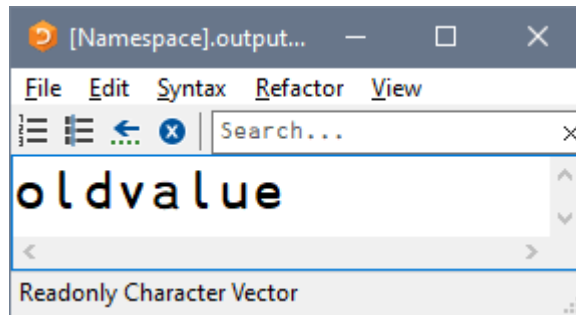
```
var←'oldvalue'  
]view var  
(3↑var)←'new'  
)ed var
```



]OUTPUT.View

- Classic window mode
- Inspect long output
- Snapshot an APL item

```
var←'oldvalue'  
]view var  
(3↑var)←'new'  
)ed var
```



]OUTPUT.View

- ❖ Classic window mode
- ❖ Inspect long output
- ❖ Snapshot an APL item
- ❖ Settings:]Box -view=



]OUTPUT.View

- Classic window mode
- Inspect long output
- Snapshot an APL item
- Settings:]Box -view=
]box on -v=max



]OUTPUT.View

- ◆ Classic window mode
- ◆ Inspect long output
- ◆ Snapshot an APL item
- ◆ Settings:]Box -view=
]box on -v=max
Was OFF -view=min



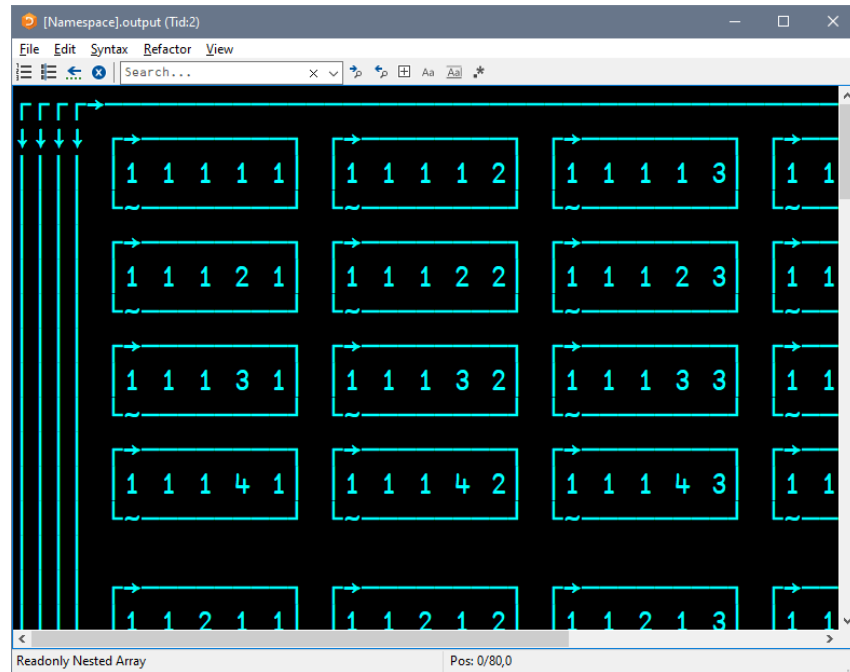
]OUTPUT.View

- ◆ Classic window mode
- ◆ Inspect long output
- ◆ Snapshot an APL item
- ◆ Settings:]Box -view=
]box on -v=max
Was OFF -view=min
]view 115



]OUTPUT.View

- Classic window mode
- Inspect long output
- Snapshot an APL item
- Settings: `]Box -view=`
 `]box on -v=max`
Was OFF `-view=min`
 `]view 115`



`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

+ Ideas for the Future



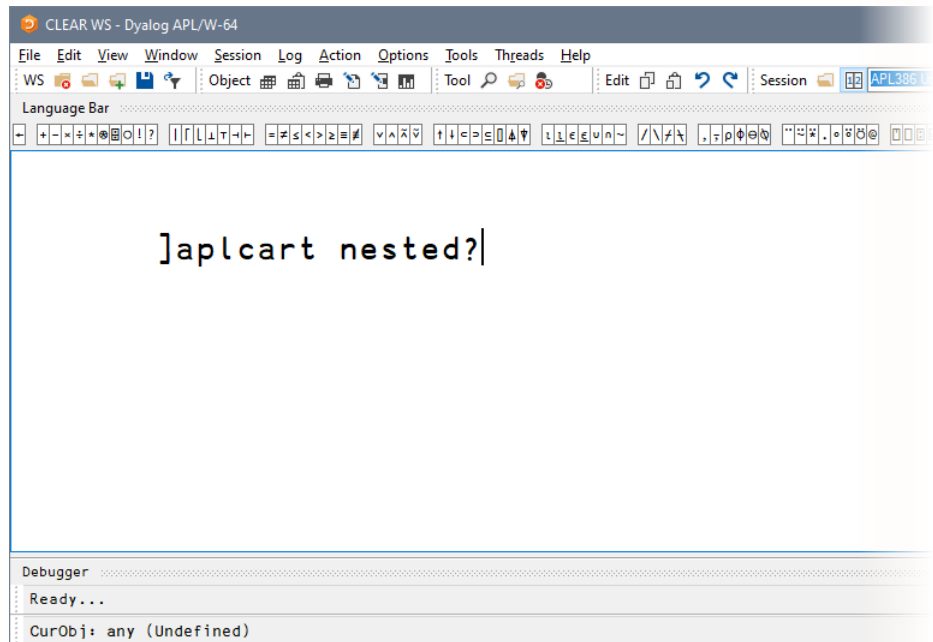
]TOOLS.APLCart

- Display in session



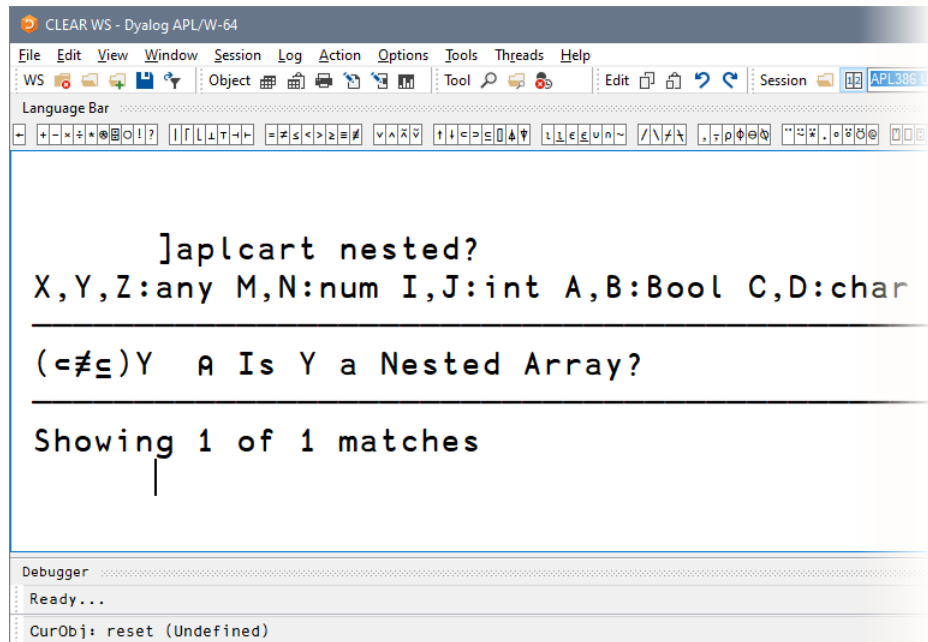
]TOOLS.APLCart

Display in session



]TOOLS.APLCart

Display in session



The screenshot shows the Dyalog APL IDE interface. The title bar reads "CLEAR WS - Dyalog APL/W-64". The menu bar includes File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, and Help. The toolbar contains icons for various functions like opening files, saving, and executing. The Language Bar shows a set of keyboard shortcuts. The main workspace displays the following APL code and output:

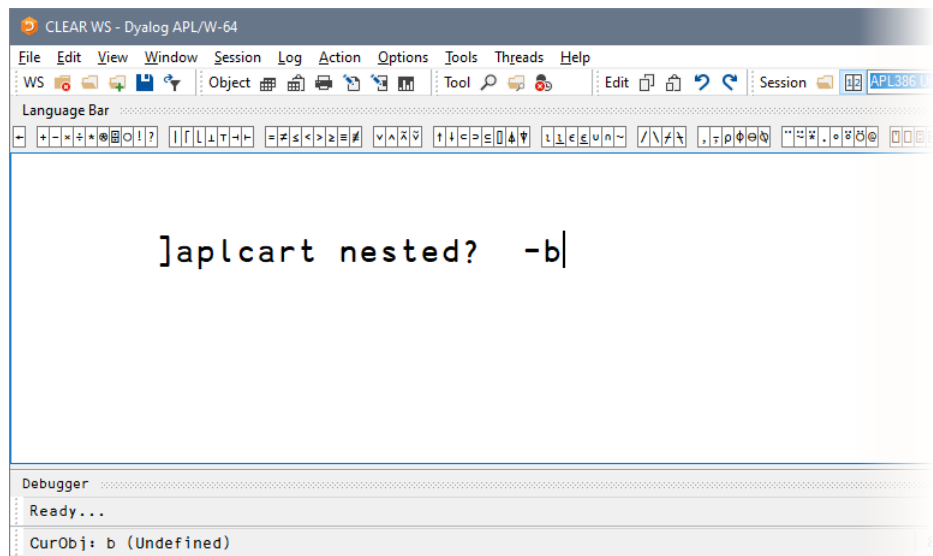
```
]aplcart nested?  
X,Y,Z:any M,N:num I,J:int A,B:Bool C,D:char  
  
(<≠≡)Y  A Is Y a Nested Array?  
  
Showing 1 of 1 matches
```

The Debugger window at the bottom shows "Ready..." and "CurObj: reset (Undefined)".



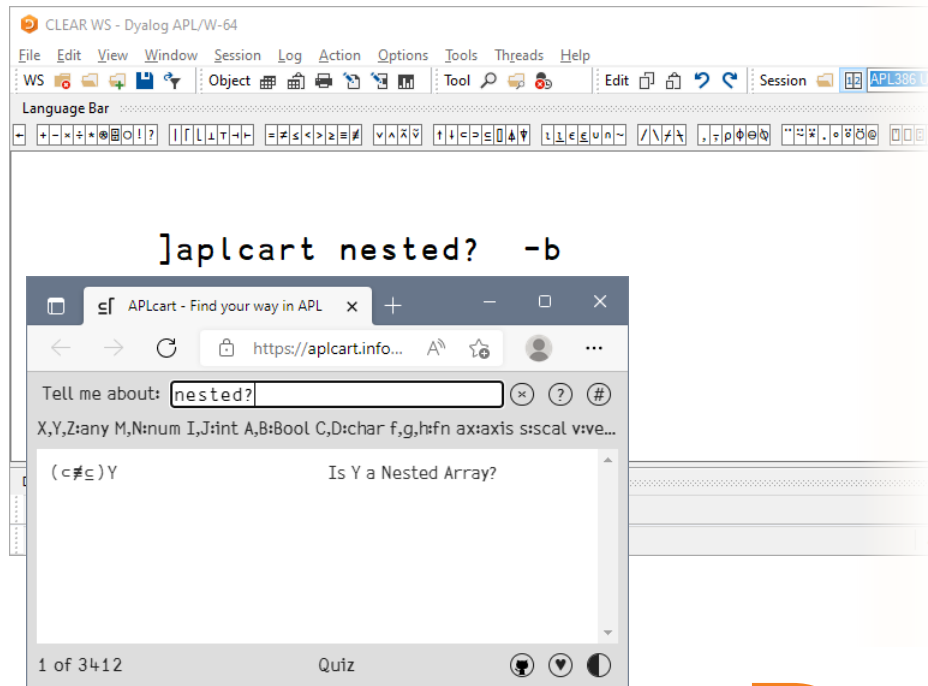
]TOOLS.APLCart

- Display in session
- Display in browser



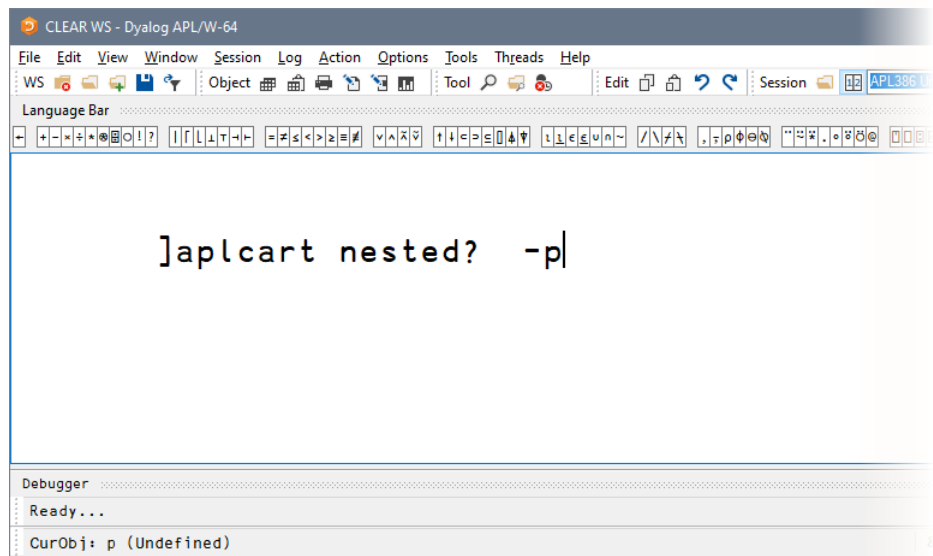
]TOOLS.APLCart

- Display in session
- Display in browser



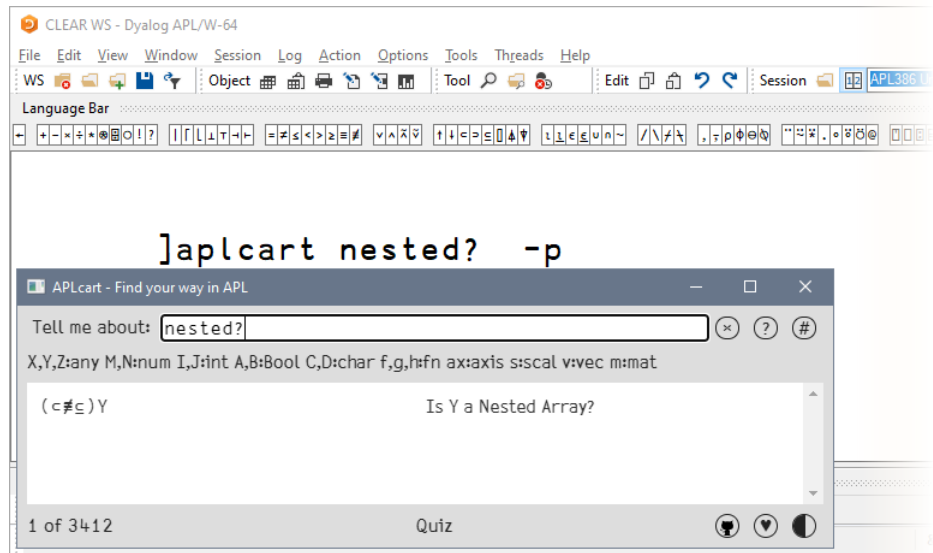
]TOOLS.APLCart

- Display in session
- Display in browser
- Display in popup



]TOOLS.APLCart

- Display in session
- Display in browser
- Display in popup



`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

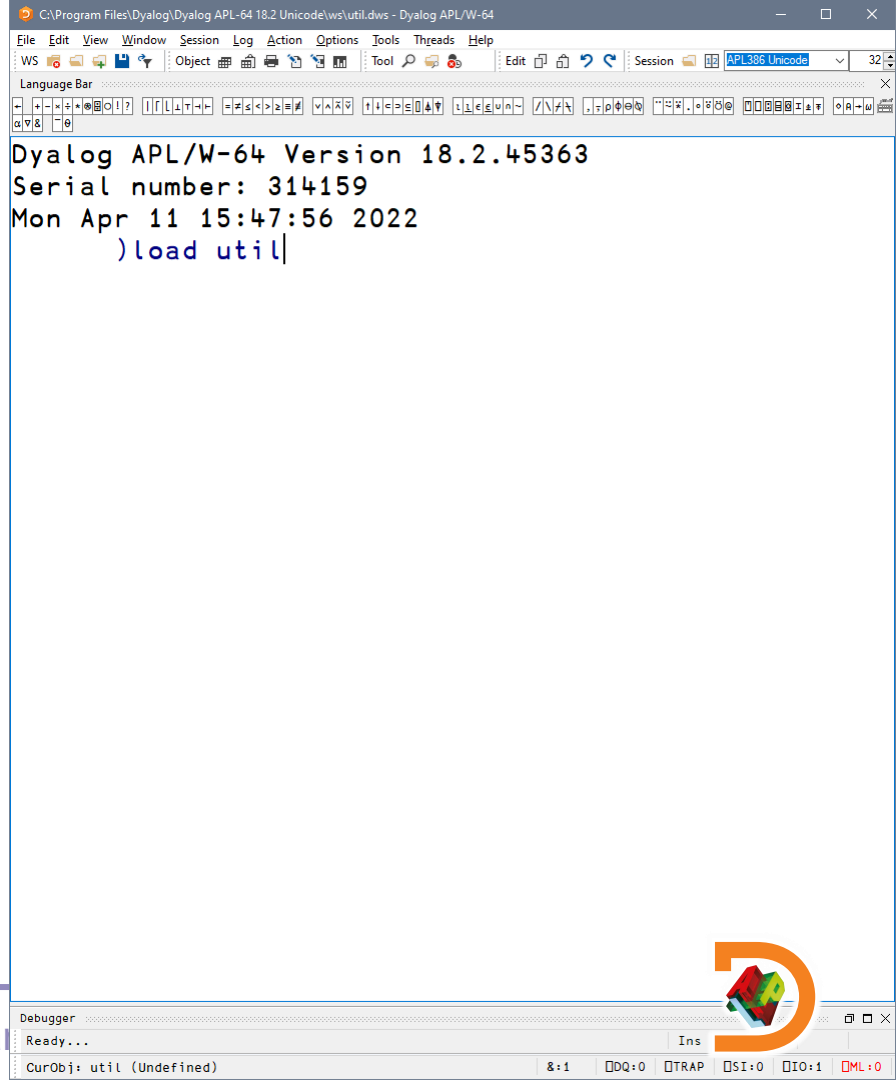
+ Ideas for the Future



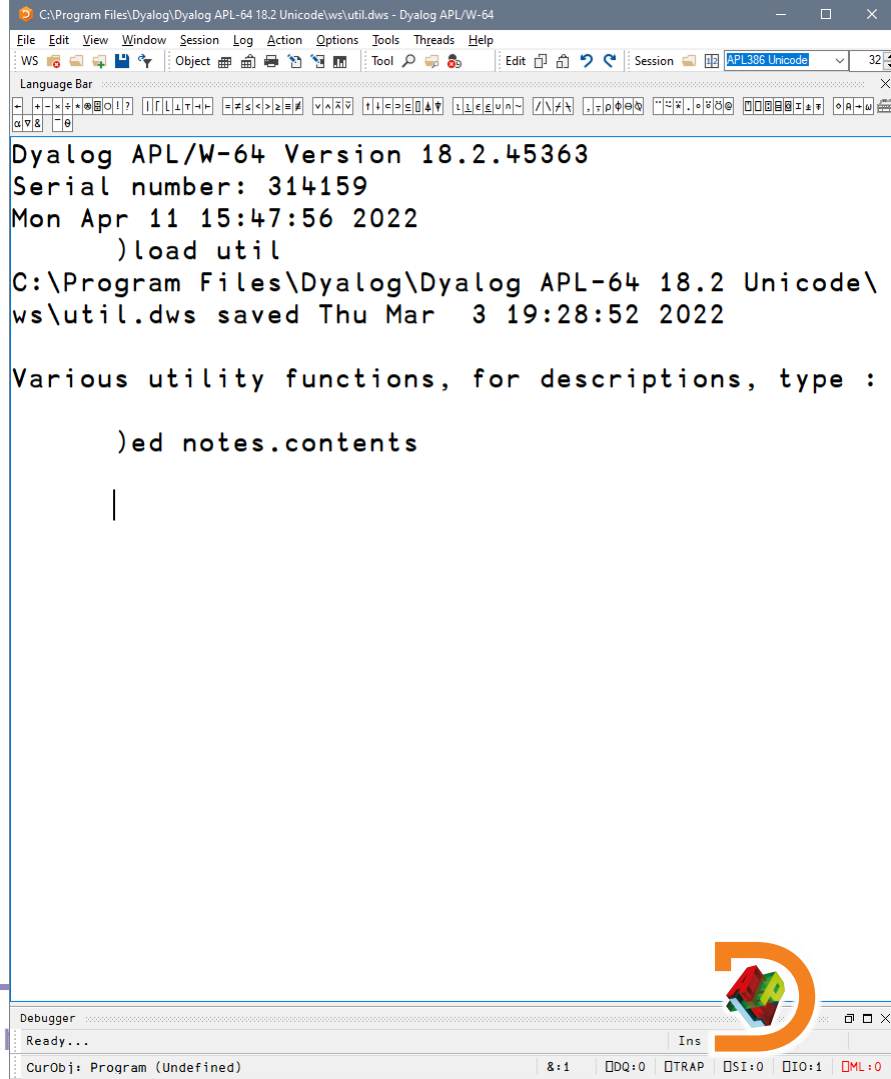
]WS.Names



]WS.Names



]WS.Names



```
C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\ws\util.dws - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL 386 Unicode 32
Language Bar
Dyalog APL/W-64 Version 18.2.45363
Serial number: 314159
Mon Apr 11 15:47:56 2022
)load util
C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\
ws\util.dws saved Thu Mar 3 19:28:52 2022

Various utility functions, for descriptions, type :

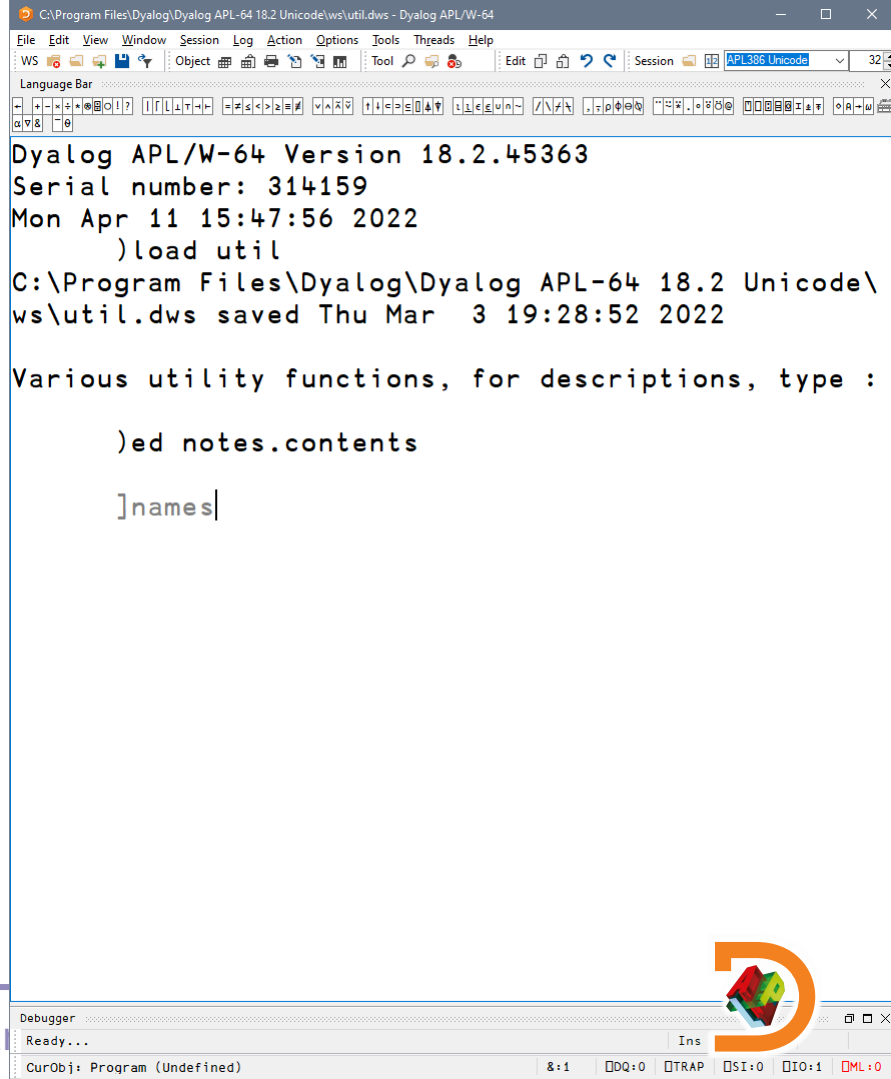
)ed notes.contents

|

Debugger Ready...
CurObj: Program (Undefined) &1 DQ:0 TRAP SI:0 IO:1 ML:0
```


]WS.Names

List all



The screenshot shows the Dyalog APL/W-64 IDE interface. The title bar indicates the file path: C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\ws\util.dws - Dyalog APL/W-64. The menu bar includes File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, and Help. The toolbar contains icons for various functions like opening files, saving, and running. The Language Bar shows the current encoding as APL 386 Unicode. The main text area displays the output of the]WS.Names command:

```
Dyalog APL/W-64 Version 18.2.45363
Serial number: 314159
Mon Apr 11 15:47:56 2022
)load util
C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\
ws\util.dws saved Thu Mar 3 19:28:52 2022

Various utility functions, for descriptions, type :

)ed notes.contents

]names|
```

The bottom status bar shows the Debugger is ready, and the current object is Program (Undefined). The status bar also displays various system metrics like &1, DQ:0, TRAP, SI:0, IO:1, and ML:0.

]WS.Names

List all



The screenshot shows the Dyalog APL/W-64 Version 18.2.45363 interface. The title bar indicates the file path: C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\ws\util.dws - Dyalog APL/W-64. The menu bar includes File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, and Help. The toolbar contains various icons for file operations, editing, and debugging. The Language Bar is visible at the top. The main window displays the following text:

```
Dyalog APL/W-64 Version 18.2.45363
Serial number: 314159
Mon Apr 11 15:47:56 2022
)load util
C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\
ws\util.dws saved Thu Mar 3 19:28:52 2022

Various utility functions, for descriptions, type :

    )ed notes.contents

]names
APLVERSION  FIND      RJUST      XWS
BIG_ENDIAN  FNGREP     SET        Z
BMVIEW      FNREPL     SETWX     _finddefine
CENTRE      FTREE      SM_TS      ltom
CLIP2PR     Globals    SNAP      notes
DETRAIL     KEYPRESS   TRAV      ss
DIR          LJUST      TREE      ΔMAPPEND
DISP         MAKEMAT    TS_SM     ΔMPUT
DISPLAY      MATRIX     Time      ΔVARS
DOIF         NTREE      WSDIFF    Δmappend
ECHO         PROP       WSPACK    Δmut
ENLIST       PROPS      XVAR
```

The bottom of the screenshot shows the Debugger window with the status "Ready..." and the current object "CurObj: Program (Undefined)". The status bar at the very bottom displays various system indicators: 8:1, DDQ:0, ITRAP, ISI:0, IO:1, and DML:0.

]WS.Names

- List all
- Group by nameclass

```
Dyalog APL/W-64 Version 18.2.45363
Serial number: 314159
Mon Apr 11 15:47:56 2022
)load util
C:\Program Files\Dyalog\Dyalog APL-64 18.2 Unicode\
ws\util.dws saved Thu Mar 3 19:28:52 2022

Various utility functions, for descriptions, type :

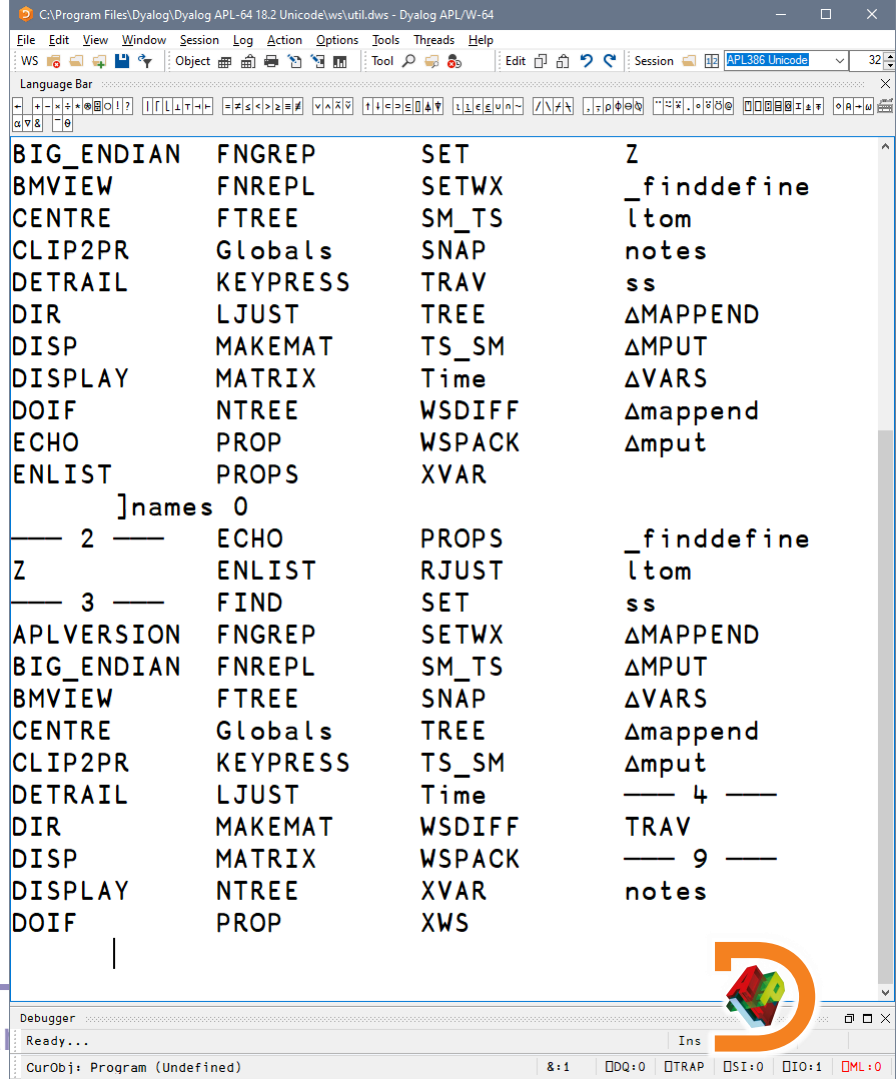
    )ed notes.contents

    ]names
APLVERSION  FIND      RJUST      XWS
BIG_ENDIAN  FNGREP     SET       Z
BMVIEW      FNREPL     SETWX     _finddefine
CENTRE      FTREE      SM_TS     ltom
CLIP2PR     Globals    SNAP      notes
DETRAIL     KEYPRESS   TRAV      ss
DIR         LJUST      TREE      ΔMAPPEND
DISP        MAKEMAT   TS_SM     ΔMPUT
DISPLAY     MATRIX     Time      ΔVARS
DOIF        NTREE     WSDIFF    Δmappend
ECHO        PROP      WSPACK    Δmput
ENLIST      PROPS     XVAR

    ]names 0
```

]WS.Names

- List all
- Group by nameclass



The screenshot shows the Dyalog APL 18.2 IDE with the command `]WS.Names` executed. The output is displayed in the main workspace, showing a list of names grouped by their nameclass. The names are listed in four columns, and the nameclass is indicated by a number (0, 2, 3, 4, 9) followed by a horizontal line. The names are: BIG_ENDIAN, BMVIEW, CENTRE, CLIP2PR, DETRAIL, DIR, DISP, DISPLAY, DOIF, ECHO, ENLIST, FNGREP, FNREPL, FTREE, Globals, KEYPRESS, LJUST, MAKEMAT, MATRIX, NTREE, PROP, PROPS, SET, SETWX, SM_TS, SNAP, TRAV, TREE, TS_SM, Time, WSDIFF, WSPACK, XVAR, Z, _finddefine, ltom, notes, ss, ΔMAPPEND, ΔMPUT, ΔVARS, Δmappend, Δmput, _finddefine, ltom, ss, ΔMAPPEND, ΔMPUT, ΔVARS, Δmappend, Δmput, TRAV, notes.

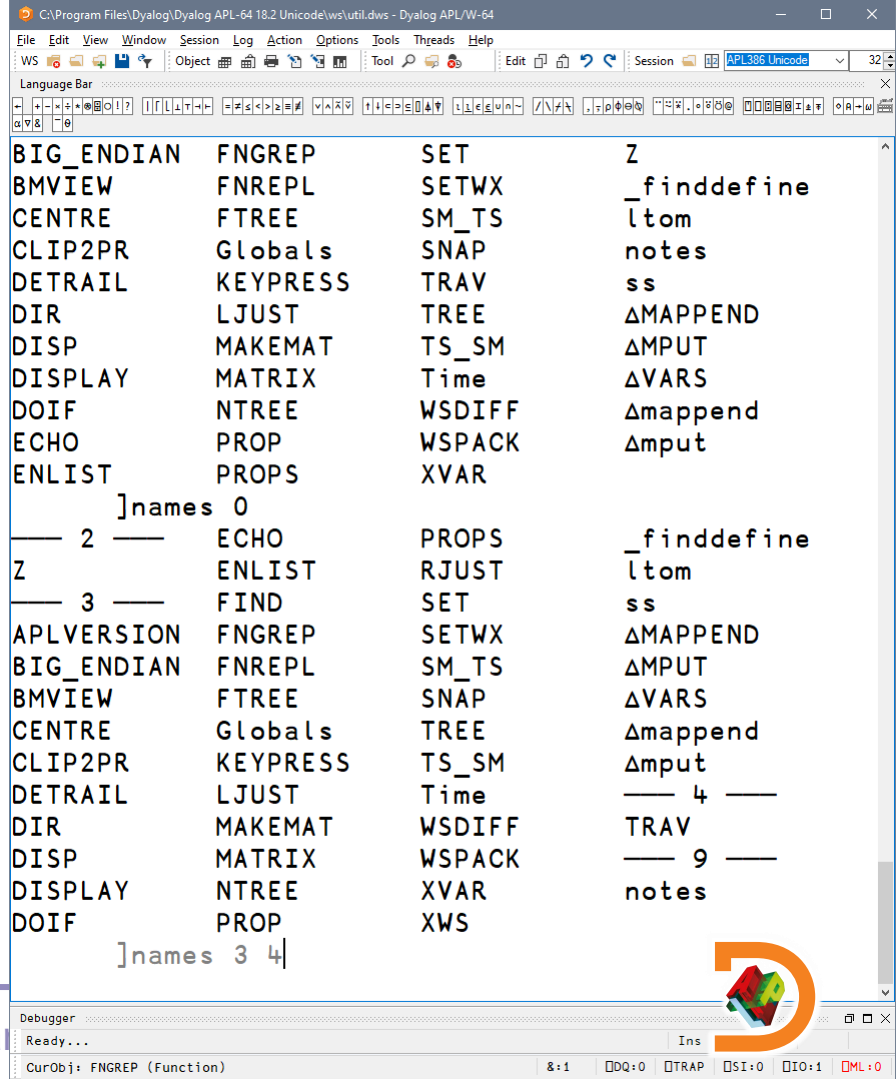
```

BIG_ENDIAN  FNGREP      SET          Z
BMVIEW      FNREPL      SETWX        _finddefine
CENTRE      FTREE       SM_TS        ltom
CLIP2PR     Globals     SNAP         notes
DETRAIL     KEYPRESS    TRAV         ss
DIR          LJUST       TREE         ΔMAPPEND
DISP        MAKEMAT     TS_SM        ΔMPUT
DISPLAY     MATRIX      Time         ΔVARS
DOIF        NTREE       WSDIFF       Δmappend
ECHO        PROP        WSPACK       Δmput
ENLIST      PROPS       XVAR
]names 0
—— 2 ——    ECHO        PROPS        _finddefine
Z           ENLIST      RJUST        ltom
—— 3 ——    FIND        SET          ss
APLVERSION FNGREP      SETWX        ΔMAPPEND
BIG_ENDIAN FNREPL      SM_TS        ΔMPUT
BMVIEW     FTREE       SNAP         ΔVARS
CENTRE     Globals   TREE         Δmappend
CLIP2PR    KEYPRESS TS_SM        Δmput
DETRAIL    LJUST      Time         —— 4 ——
DIR        MAKEMAT  WSDIFF       TRAV
DISP       MATRIX   WSPACK       —— 9 ——
DISPLAY    NTREE     XVAR         notes
DOIF       PROP     XWS

```

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass



```
File Edit View Window Session Log Action Options Tools Threads Help
WS [Icons] Object [Icons] Tool [Icons] Edit [Icons] Session [Icons] APL 386 Unicode 32
Language Bar
BIG_ENDIAN FNGREP SET Z
BMVIEW FNREPL SETWX _finddefine
CENTRE FTREE SM_TS ltom
CLIP2PR Globals SNAP notes
DETRAIL KEYPRESS TRAV ss
DIR LJUST TREE ΔMAPPEND
DISP MAKEMAT TS_SM ΔMPUT
DISPLAY MATRIX Time ΔVARS
DOIF NTREE WSDIFF Δmappend
ECHO PROP WSPACK Δmput
ENLIST PROPS XVAR

]names 0
— 2 — ECHO PROPS _finddefine
Z ENLIST RJUST ltom
— 3 — FIND SET ss
APLVERSION FNGREP SETWX ΔMAPPEND
BIG_ENDIAN FNREPL SM_TS ΔMPUT
BMVIEW FTREE SNAP ΔVARS
CENTRE Globals TREE Δmappend
CLIP2PR KEYPRESS TS_SM Δmput
DETRAIL LJUST Time — 4 —
DIR MAKEMAT WSDIFF TRAV
DISP MATRIX WSPACK — 9 —
DISPLAY NTREE XVAR notes
DOIF PROP XWS

]names 3 4
```

Debugger Ready... Ins

CurObj: FNGREP (Function) 8:1 □DQ:0 □TRAP □SI:0 □IO:1 □ML:0

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass

```

Z
—— 3 ——
APLVERSION  FNGREP  SETWX  ltom
BIG_ENDIAN  FNREPL  SM_TS  ss
BMVIEW      FTREE   SNAP   ΔMAPPEND
CENTRE      Globals TREE   ΔMPUT
CLIP2PR     KEYPRESS TS_SM  ΔMPUT
DETRAIL     LJUST   Time   — 4 —
DIR         MAKEMAT WSDIFF  TRAV
DISP        MATRIX  WSPACK — 9 —
DISPLAY     NTREE   XVAR   notes
DOIF        PROP    XWS

]names 3 4
—— 3 ——
APLVERSION  FNGREP  SETWX  XWS
BIG_ENDIAN  FNREPL  SM_TS  _finddefine
BMVIEW      FTREE   SNAP   ltom
CENTRE      Globals TREE   ss
CLIP2PR     KEYPRESS TS_SM  ΔMAPPEND
DETRAIL     LJUST   Time   ΔMPUT
DIR         MAKEMAT WSDIFF  ΔMPUT
DISP        MATRIX  WSPACK — 4 —
DISPLAY     NTREE   XVAR   TRAV
DOIF        PROP
ECHO

```

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace

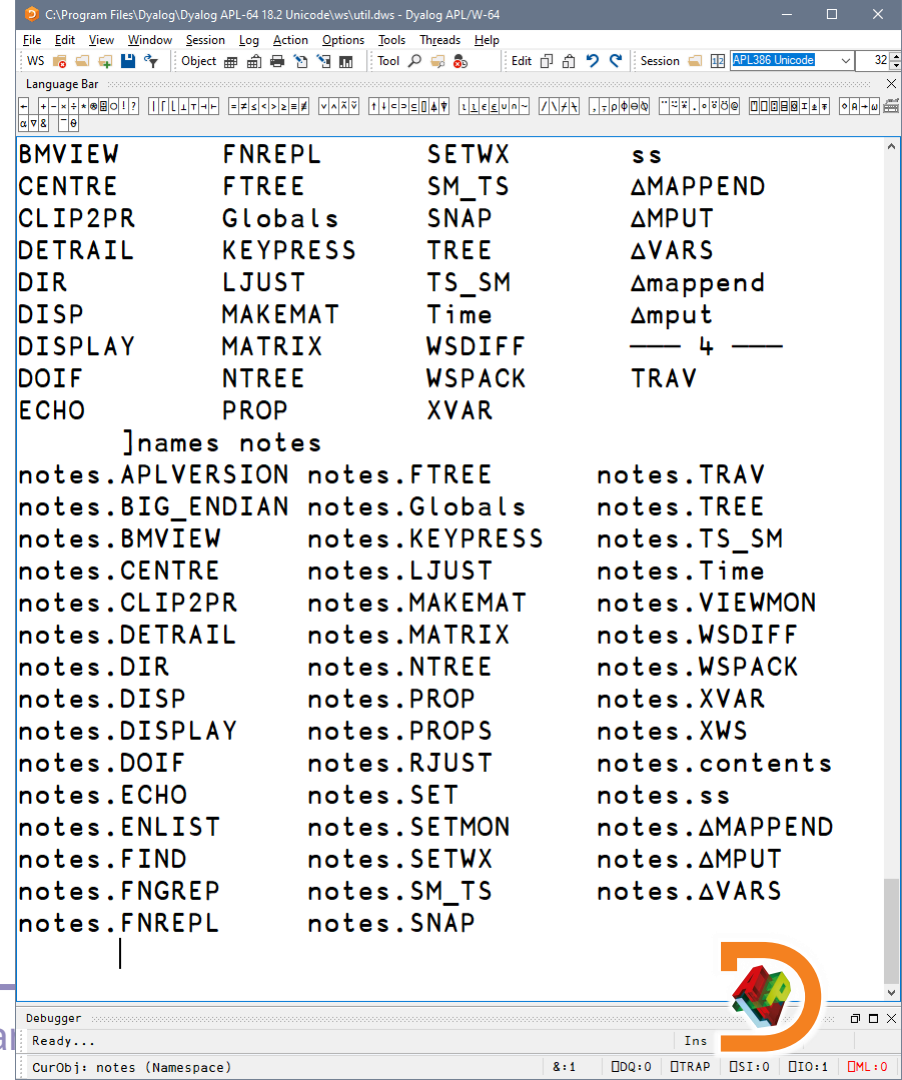
```

Z
—— 3 ——
APLVERSION
BIG_ENDIAN
BMVIEW
CENTRE
CLIP2PR
DETRAIL
DIR
DISP
DISPLAY
DOIF
]names 3 4
—— 3 ——
APLVERSION
BIG_ENDIAN
BMVIEW
CENTRE
CLIP2PR
DETRAIL
DIR
DISP
DISPLAY
DOIF
ECHO
ENLIST
FNGREP
FNREPL
FTREE
Globals
KEYPRESS
LJUST
MAKEMAT
MATRIX
NTREE
PROP
RJUST
SET
SETWX
SM_TS
SNAP
TREE
TS_SM
Time
WSDIFF
WSPACK
XVAR
XWS
ltom
ss
ΔMAPPEND
ΔMPUT
ΔVARS
Δmappend
Δmput
TRAV
notes
]names 3 4
—— 3 ——
APLVERSION
BIG_ENDIAN
BMVIEW
CENTRE
CLIP2PR
DETRAIL
DIR
DISP
DISPLAY
DOIF
ECHO
ENLIST
FNGREP
FNREPL
FTREE
Globals
KEYPRESS
LJUST
MAKEMAT
MATRIX
NTREE
PROP
RJUST
SET
SETWX
SM_TS
SNAP
TREE
TS_SM
Time
WSDIFF
WSPACK
XVAR
XWS
ltom
ss
ΔMAPPEND
ΔMPUT
ΔVARS
Δmappend
Δmput
TRAV
notes
]names notes|

```

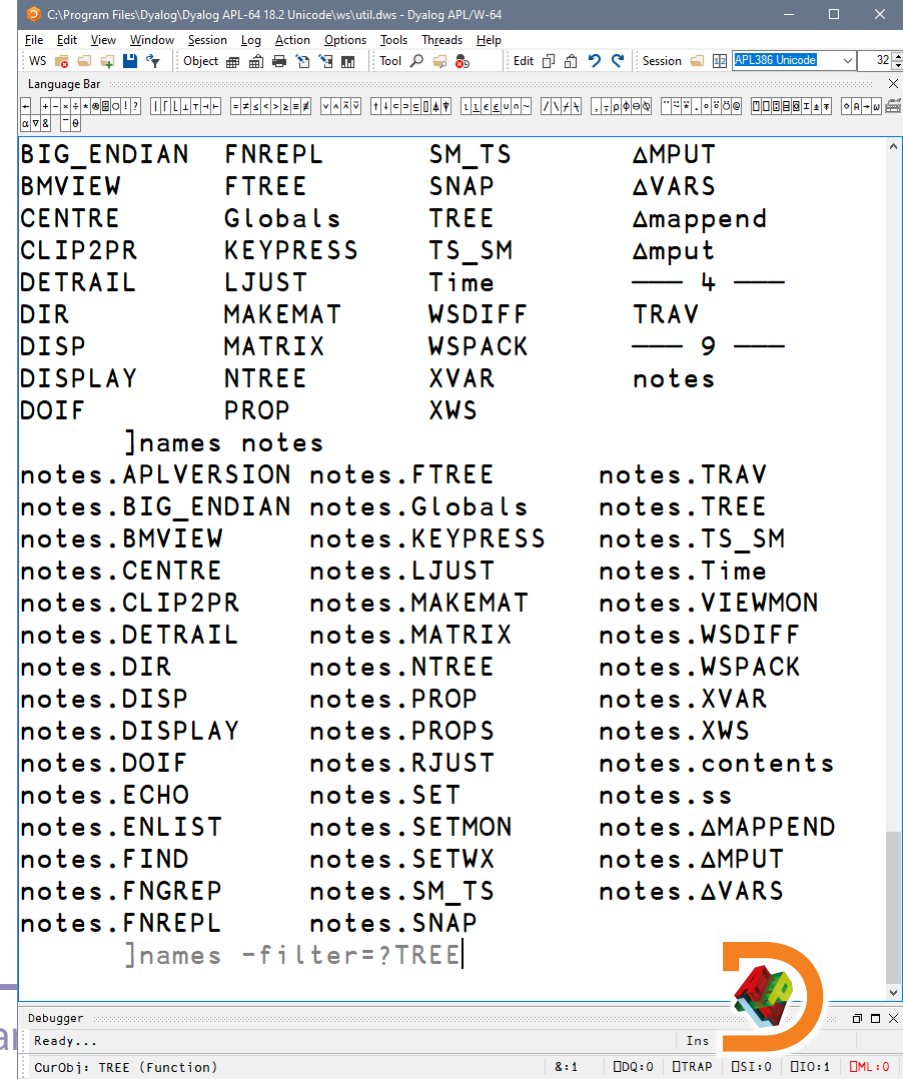
]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace



]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards



The screenshot shows the Dyalog APL 18.2 IDE with the command `]WS.Names` executed. The output is displayed in the main workspace, showing a list of names and their corresponding namespaces. The names are grouped by namespace, with some names having a trailing underscore and a number. The namespaces are listed on the right side of the output.

```

BIG_ENDIAN  FNREPL      SM_TS      ΔMPUT
BMVIEW      FTREE        SNAP      ΔVARS
CENTRE      Globals      TREE      Δmappend
CLIP2PR     KEYPRESS    TS_SM     Δmput
DETRAIL     LJUST       Time      — 4 —
DIR         MAKEMAT     WSDIFF    TRAV
DISP        MATRIX      WSPACK    — 9 —
DISPLAY     NTREE       XVAR      notes
DOIF        PROP        XWS

]names notes
notes.APLVERSION notes.FTREE      notes.TRAV
notes.BIG_ENDIAN notes.Globals    notes.TREE
notes.BMVIEW     notes.KEYPRESS  notes.TS_SM
notes.CENTRE     notes.LJUST     notes.Time
notes.CLIP2PR    notes.MAKEMAT   notes.VIEWMON
notes.DETRAIL    notes.MATRIX    notes.WSDIFF
notes.DIR        notes.NTREE     notes.WSPACK
notes.DISP       notes.PROP      notes.XVAR
notes.DISPLAY    notes.PROPS     notes.XWS
notes.DOIF       notes.RJUST     notes.contents
notes.ECHO       notes.SET       notes.ss
notes.ENLIST     notes.SETMON    notes.ΔMAPPEND
notes.FIND       notes.SETWX     notes.ΔMPUT
notes.FNGREP     notes.SM_TS     notes.ΔVARS
notes.FNREPL     notes.SNAP

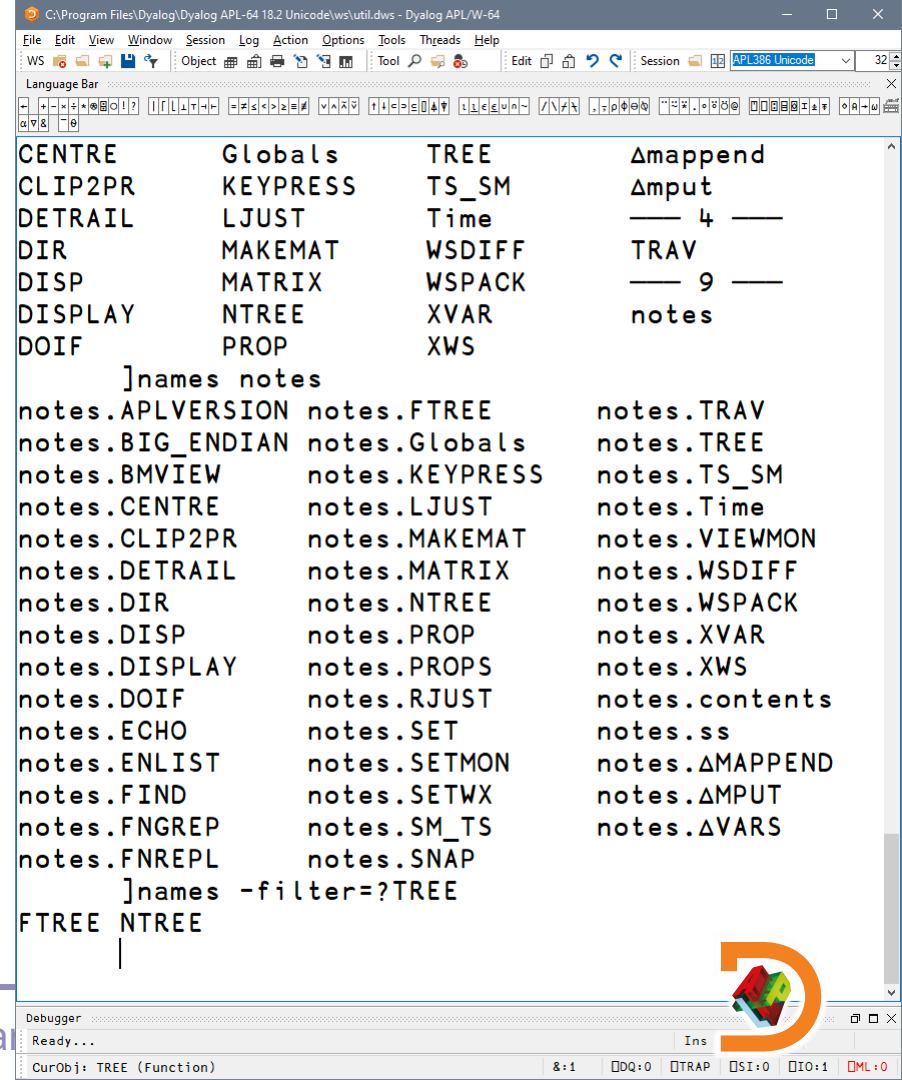
]names -filter=?TREE|

```

The IDE interface includes a menu bar (File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, Help), a toolbar, and a status bar at the bottom showing the current object as `TREE (Function)`.

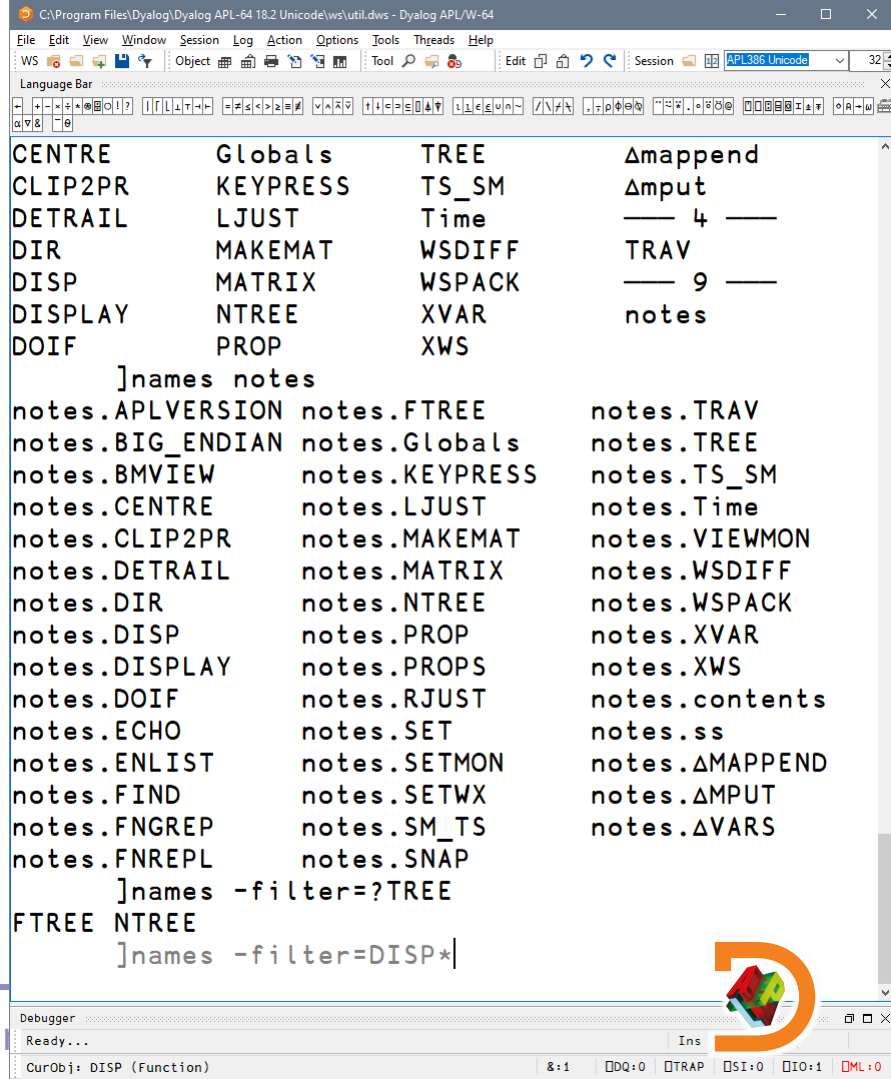
]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards



]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards



The screenshot shows the Dyalog APL 18.2 IDE with the following components:

- Title Bar:** C:\Program Files\Dyalog\APL 18.2 Unicode\ws\util.dws - Dyalog APL/W-64
- Menu Bar:** File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, Help
- Toolbar:** Includes icons for File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, Help, and a search icon.
- Language Bar:** Shows the current language as 'WS'.
- Main Window:** Displays the output of the `]WS.Names` command. The output is a table of names and their classes, followed by a list of names filtered by the `-filter=?TREE` and `-filter=DISP*` options.
- Debugger:** Shows the current function being executed as `CurObj: DISP (Function)`.

The output of `]WS.Names` is as follows:

NAME	CLASS
CENTRE	Globals
CLIP2PR	KEYPRESS
DETRAIL	LJUST
DIR	MAKEMAT
DISP	MATRIX
DISPLAY	NTREE
DOIF	PROP
FTREE	NTREE
NTREE	NTREE
SNAP	NTREE
TS_SM	TS_SM
Time	Time
WSDIFF	WSDIFF
WSPACK	WSPACK
XVAR	XVAR
XWS	XWS
Δmappend	Δmappend
Δmput	Δmput
TRAV	TRAV
notes	notes

Names filtered by `-filter=?TREE`:

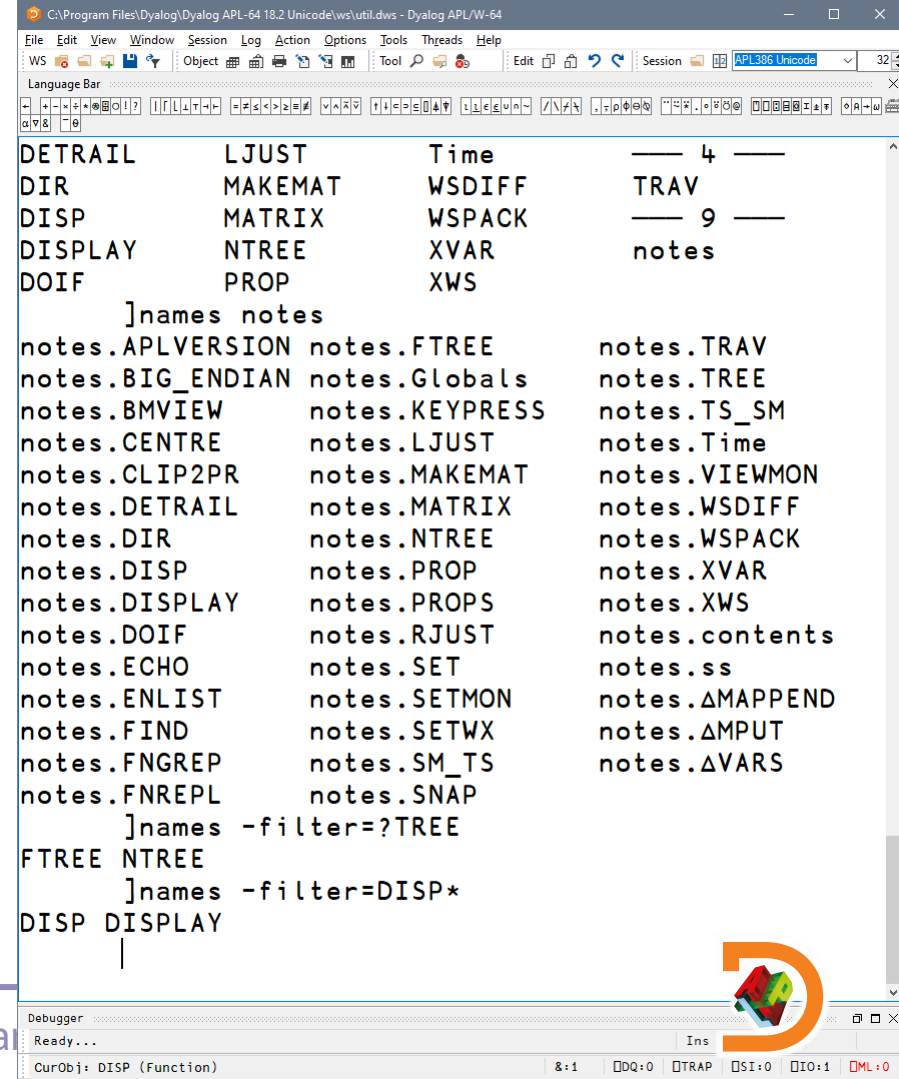
- notes.APLVERSION
- notes.BIG_ENDIAN
- notes.BMVIEW
- notes.CENTRE
- notes.CLIP2PR
- notes.DETRAIL
- notes.DIR
- notes.DISP
- notes.DISPLAY
- notes.DOIF
- notes.ECHO
- notes.ENLIST
- notes.FIND
- notes.FNGREP
- notes.FNREPL

Names filtered by `-filter=DISP*`:

- notes.FTREE
- notes.Globals
- notes.KEYPRESS
- notes.LJUST
- notes.MAKEMAT
- notes.MATRIX
- notes.NTREE
- notes.PROP
- notes.PROPS
- notes.RJUST
- notes.SET
- notes.SETMON
- notes.SETWX
- notes.SM_TS
- notes.SNAP

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards




The screenshot shows the Dyalog APL 18.2 IDE with the following content:

File Edit View Window Session Log Action Options Tools Threads Help
WS [Icons] Object [Icons] Tool [Icons] Edit [Icons] Session [Icons] APL 386 Unicode 32

Language Bar [Icons]

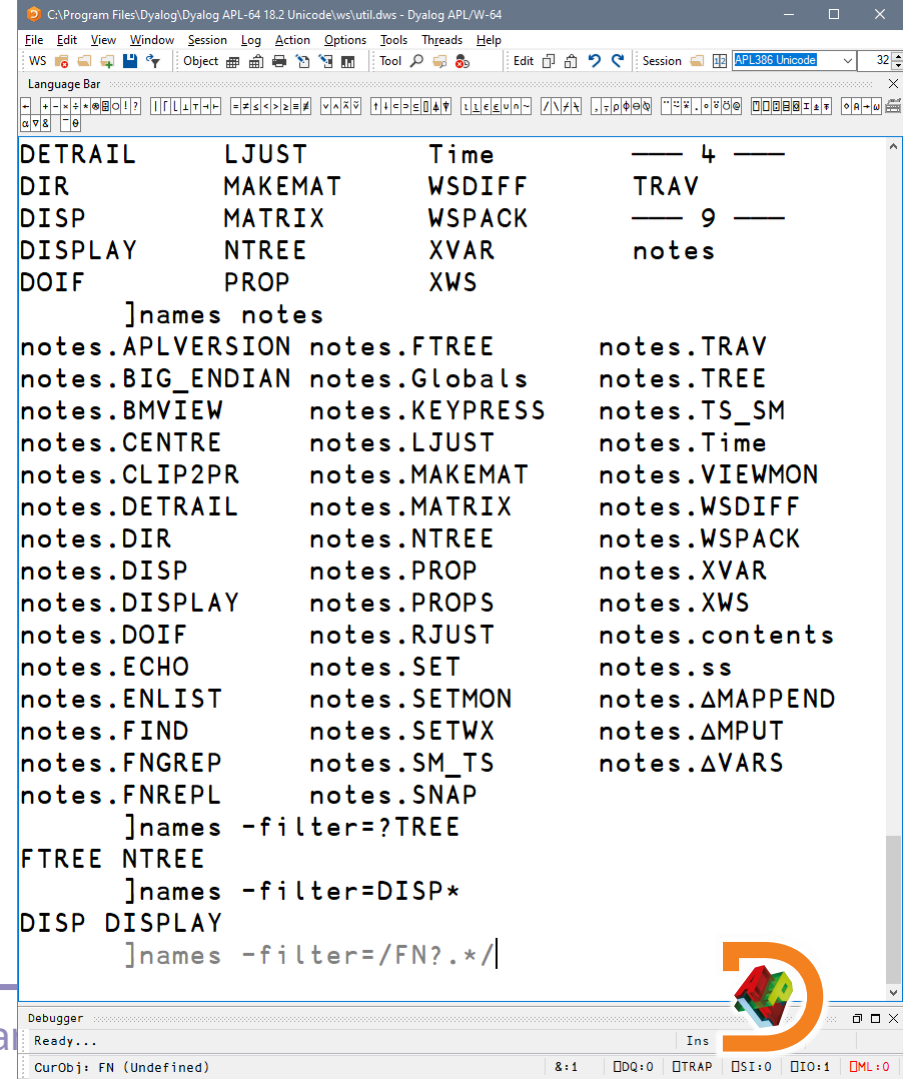
```
DETRAIL      LJUST      Time      — 4 —  
DIR           MAKEMAT    WSDIFF    TRAV  
DISP          MATRIX     WSPACK    — 9 —  
DISPLAY      NTREE      XVAR      notes  
DOIF          PROP       XWS  
  
]names notes  
notes.APLVERSION notes.FTREE      notes.TRAV  
notes.BIG_ENDIAN  notes.Globals  notes.TREE  
notes.BMVIEW      notes.KEYPRESS notes.TS_SM  
notes.CENTRE       notes.LJUST    notes.Time  
notes.CLIP2PR      notes.MAKEMAT  notes.VIEWMON  
notes.DETRAIL      notes.MATRIX   notes.WSDIFF  
notes.DIR          notes.NTREE    notes.WSPACK  
notes.DISP         notes.PROP     notes.XVAR  
notes.DISPLAY      notes.PROPS    notes.XWS  
notes.DOIF         notes.RJUST    notes.contents  
notes.ECHO         notes.SET      notes.ss  
notes.ENLIST       notes.SETMON   notes.ΔMAPPEND  
notes.FIND         notes.SETWX    notes.ΔMPUT  
notes.FNGREP       notes.SM_TS    notes.ΔVARS  
notes.FNREPL       notes.SNAP  
  
]names -filter=?TREE  
FTREE NTREE  
]names -filter=DISP*  
DISP DISPLAY  
|
```

Debugger
Ready... Ins
CurObj: DISP (Function) 8:1 □□Q:0 □TRAP □SI:0 □IO:1 □ML:0



]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards
- Filter using regex



The screenshot shows the Dyalog APL 18.2 IDE with the following content:

File Edit View Window Session Log Action Options Tools Threads Help
WS WS Object Object Tool Tool Session APL 386 Unicode 32

Language Bar

DETRAIL	LJUST	Time	4
DIR	MAKEMAT	WSDIFF	TRAV
DISP	MATRIX	WSPACK	9
DISPLAY	NTREE	XVAR	notes
DOIF	PROP	XWS	

```
]names notes
notes.APLVERSION notes.FTREE notes.TRAV
notes.BIG_ENDIAN notes.Globals notes.TREE
notes.BMVIEW notes.KEYPRESS notes.TS_SM
notes.CENTRE notes.LJUST notes.Time
notes.CLIP2PR notes.MAKEMAT notes.VIEWMON
notes.DETRAIL notes.MATRIX notes.WSDIFF
notes.DIR notes.NTREE notes.WSPACK
notes.DISP notes.PROP notes.XVAR
notes.DISPLAY notes.PROPS notes.XWS
notes.DOIF notes.RJUST notes.contents
notes.ECHO notes.SET notes.ss
notes.ENLIST notes.SETMON notes.ΔMAPPEND
notes.FIND notes.SETWX notes.ΔMPUT
notes.FNGREP notes.SM_TS notes.ΔVARS
notes.FNREPL notes.SNAP

]names -filter=?TREE
FTREE NTREE

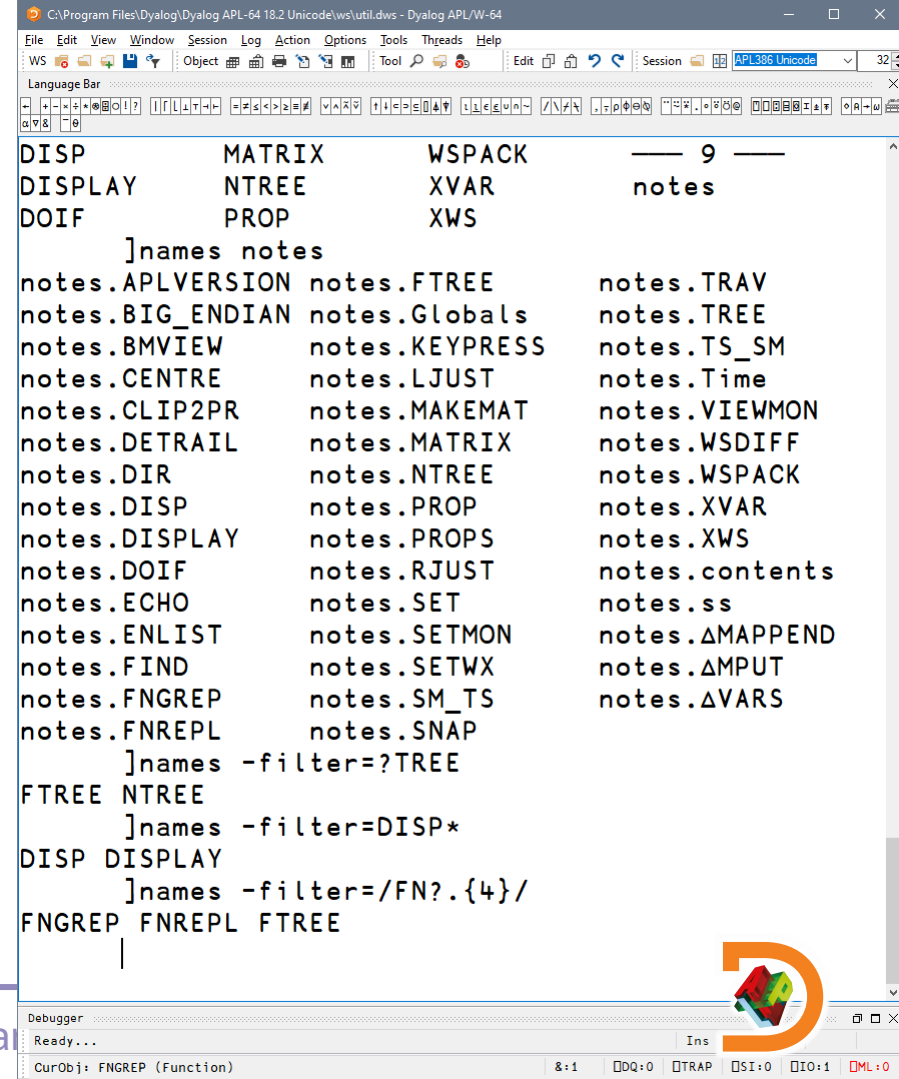
]names -filter=DISP*
DISP DISPLAY

]names -filter=/FN?.*/
```

Debugger
Ready... Ins
CurObj: FN (Undefined) 8:1 □DQ:0 □TRAP □SI:0 □IO:1 □ML:0

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards
- Filter using regex



The screenshot shows the Dyalog APL 18.2 IDE with the following components:

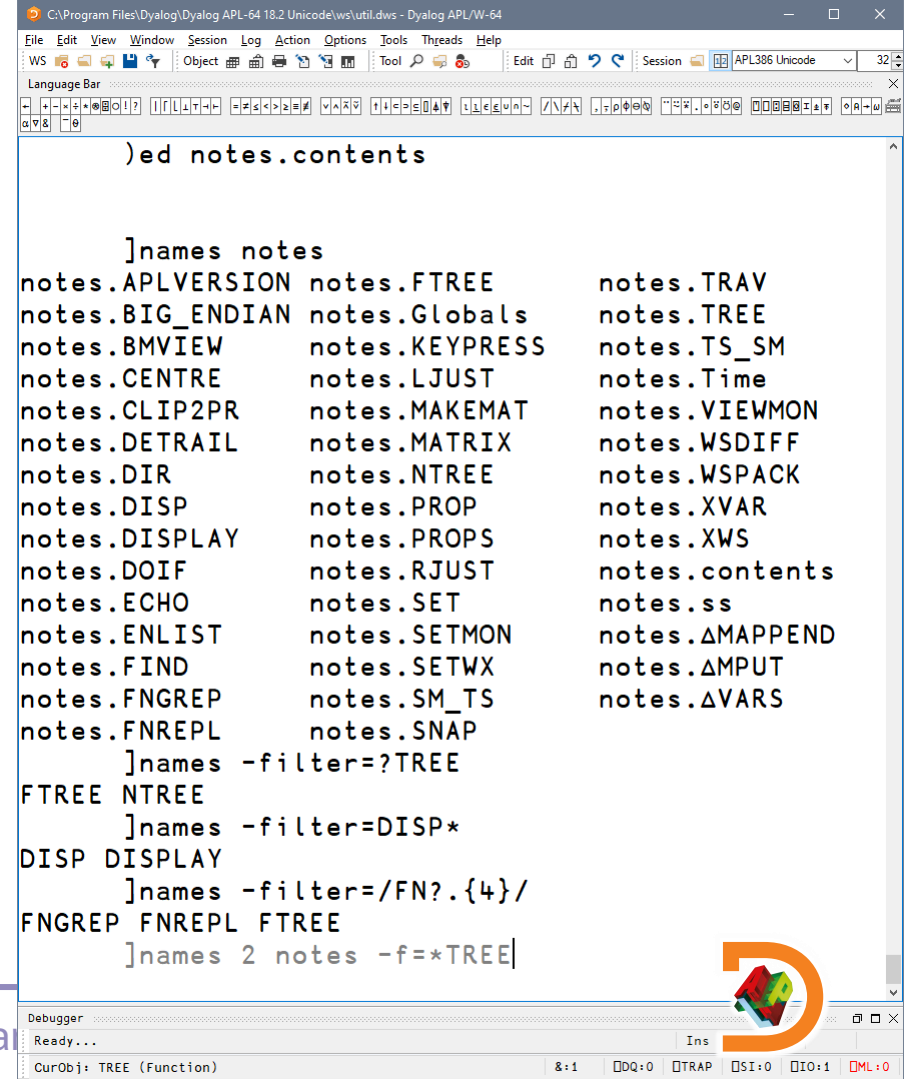
- Language Bar:** Located at the top, it includes a menu bar (File, Edit, View, Window, Session, Log, Action, Options, Tools, Threads, Help) and a toolbar with various icons for editing and navigation.
- Main Window:** Displays the output of the `]WS.Names` command. The output is a table with four columns: `DISP`, `MATRIX`, `WSPACK`, and `notes`. The first column lists various APL namespaces, while the other three columns show their respective nameclasses. Below the table, there are several lines of APL code, including `]names notes`, `]names -filter=?TREE`, `]names -filter=DISP*`, `]names -filter=/FN?.{4}/`, and `FTREE NTREE`.
- Debugger:** Located at the bottom, it shows the current object being debugged as `CurObj: FNGREP (Function)` and the status as `Ready...`.

DISP	MATRIX	WSPACK	notes
DISPLAY	NTREE	XVAR	
DOIF	PROP	XWS	
]names notes			
notes.APLVERSION	notes.FTREE	notes.TRAV	
notes.BIG_ENDIAN	notes.Globals	notes.TREE	
notes.BMVIEW	notes.KEYPRESS	notes.TS_SM	
notes.CENTRE	notes.LJUST	notes.Time	
notes.CLIP2PR	notes.MAKEMAT	notes.VIEWMON	
notes.DETRAIL	notes.MATRIX	notes.WSDIFF	
notes.DIR	notes.NTREE	notes.WSPACK	
notes.DISP	notes.PROP	notes.XVAR	
notes.DISPLAY	notes.PROPS	notes.XWS	
notes.DOIF	notes.RJUST	notes.contents	
notes.ECHO	notes.SET	notes.ss	
notes.ENLIST	notes.SETMON	notes.ΔMAPPEND	
notes.FIND	notes.SETWX	notes.ΔMPUT	
notes.FNGREP	notes.SM_TS	notes.ΔVARS	
notes.FNREPL	notes.SNAP		
]names -filter=?TREE			
FTREE	NTREE		
]names -filter=DISP*			
DISP	DISPLAY		
]names -filter=/FN?.{4}/			
FNGREP	FNREPL	FTREE	

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards
- Filter using regex

Combine all of these!



The screenshot shows the Dyalog APL 18.2 IDE. The main window displays the command `]WS.Names` and its output, which is a list of names and their corresponding nameclasses. The output is organized into three columns. The first column lists names, the second column lists nameclasses, and the third column lists the contents of the notes object. The names listed are: `notes.APLVERSION`, `notes.BIG_ENDIAN`, `notes.BMVIEW`, `notes.CENTRE`, `notes.CLIP2PR`, `notes.DETRAIL`, `notes.DIR`, `notes.DISP`, `notes.DISPLAY`, `notes.DOIF`, `notes.ECHO`, `notes.ENLIST`, `notes.FIND`, `notes.FNGREP`, `notes.FNREPL`, `notes.FTREE`, `notes.Globals`, `notes.KEYPRESS`, `notes.LJUST`, `notes.MAKEMAT`, `notes.MATRIX`, `notes.NTREE`, `notes.PROP`, `notes.PROPS`, `notes.RJUST`, `notes.SET`, `notes.SETMON`, `notes.SETWX`, `notes.SM_TS`, `notes.SNAP`, `notes.TRAV`, `notes.TREE`, `notes.TS_SM`, `notes.Time`, `notes.VIEWMON`, `notes.WSDIFF`, `notes.WSPACK`, `notes.XVAR`, `notes.XWS`, `notes.contents`, `notes.ss`, `notes.ΔMAPPEND`, `notes.ΔMPUT`, and `notes.ΔVARS`. The command `]WS.Names` is also shown at the bottom of the window.

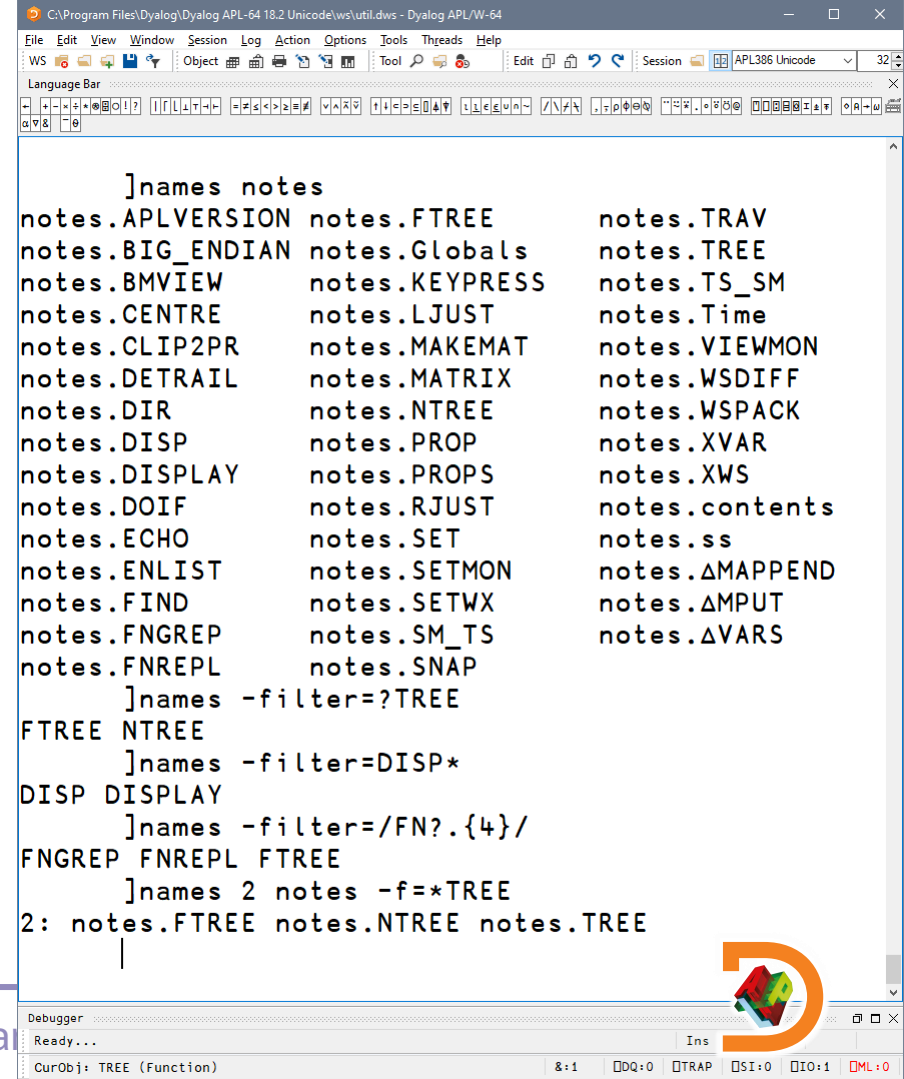
```
]ed notes.contents

]names notes
notes.APLVERSION notes.FTREE      notes.TRAV
notes.BIG_ENDIAN  notes.Globals   notes.TREE
notes.BMVIEW      notes.KEYPRESS  notes.TS_SM
notes.CENTRE      notes.LJUST     notes.Time
notes.CLIP2PR     notes.MAKEMAT   notes.VIEWMON
notes.DETRAIL     notes.MATRIX    notes.WSDIFF
notes.DIR         notes.NTREE     notes.WSPACK
notes.DISP        notes.PROP       notes.XVAR
notes.DISPLAY     notes.PROPS     notes.XWS
notes.DOIF        notes.RJUST     notes.contents
notes.ECHO        notes.SET       notes.ss
notes.ENLIST      notes.SETMON    notes.ΔMAPPEND
notes.FIND        notes.SETWX     notes.ΔMPUT
notes.FNGREP      notes.SM_TS     notes.ΔVARS
notes.FNREPL     notes.SNAP
]names -filter=?TREE
FTREE NTREE
]names -filter=DISP*
DISP DISPLAY
]names -filter=/FN?.{4}/
FNGREP FNREPL FTREE
]names 2 notes -f=*TREE|
```

]WS.Names

- List all
- Group by nameclass
- Filter by nameclass
- Specific namespace
- Filter using wildcards
- Filter using regex

Combine all of these!



The screenshot shows the Dyalog APL IDE interface. The main window displays the command `]WS.Names` and its output, which is a list of names and their corresponding namespaces. The output is organized into three columns. The first column lists names, the second column lists namespaces, and the third column lists namespaces. The names are: `notes.APLVERSION`, `notes.BIG_ENDIAN`, `notes.BMVIEW`, `notes.CENTRE`, `notes.CLIP2PR`, `notes.DETRAIL`, `notes.DIR`, `notes.DISP`, `notes.DISPLAY`, `notes.DOIF`, `notes.ECHO`, `notes.ENLIST`, `notes.FIND`, `notes.FNGREP`, `notes.FNREPL`, `FTREE`, `NTREE`, `DISP`, `DISPLAY`, `FNGREP`, `FNREPL`, `FTREE`. The namespaces are: `notes.FTREE`, `notes.Globals`, `notes.KEYPRESS`, `notes.LJUST`, `notes.MAKEMAT`, `notes.MATRIX`, `notes.NTREE`, `notes.PROP`, `notes.PROPS`, `notes.RJUST`, `notes.SET`, `notes.SETMON`, `notes.SETWX`, `notes.SM_TS`, `notes.SNAP`, `notes.TREE`, `notes.TRAV`, `notes.TIME`, `notes.VIEWMON`, `notes.WSDIFF`, `notes.WSPACK`, `notes.XVAR`, `notes.XWS`, `notes.contents`, `notes.ss`, `notes.ΔMAPPEND`, `notes.ΔMPUT`, `notes.ΔVARS`. The command `]WS.Names` is shown at the bottom of the window, with the output displayed above it. The IDE interface includes a menu bar, a toolbar, and a status bar.

```
]names notes
notes.APLVERSION notes.FTREE notes.TRAV
notes.BIG_ENDIAN notes.Globals notes.TREE
notes.BMVIEW notes.KEYPRESS notes.TS_SM
notes.CENTRE notes.LJUST notes.Time
notes.CLIP2PR notes.MAKEMAT notes.VIEWMON
notes.DETRAIL notes.MATRIX notes.WSDIFF
notes.DIR notes.NTREE notes.WSPACK
notes.DISP notes.PROP notes.XVAR
notes.DISPLAY notes.PROPS notes.XWS
notes.DOIF notes.RJUST notes.contents
notes.ECHO notes.SET notes.ss
notes.ENLIST notes.SETMON notes.ΔMAPPEND
notes.FIND notes.SETWX notes.ΔMPUT
notes.FNGREP notes.SM_TS notes.ΔVARS
notes.FNREPL notes.SNAP
]names -filter=?TREE
FTREE NTREE
]names -filter=DISP*
DISP DISPLAY
]names -filter=/FN?.{4}/
FNGREP FNREPL FTREE
]names 2 notes -f=*TREE
2: notes.FTREE notes.NTREE notes.TREE
```


`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

+ Ideas for the Future



]OUTPUT.Repr

]repr *expression*



]OUTPUT.Repr

]repr *expression* -format=APL

single-line APL expression



]OUTPUT.Repr

]repr *expression* -format=APL single-line APL expression
 APLAN APL Array Notation



]OUTPUT.Repr

]repr *expression* -format=APL

single-line APL expression

APLAN

APL Array Notation

JS

JavaScript for inclusion in code



]OUTPUT.Repr

]repr <i>expression</i> -format=APL	single-line APL expression
APLAN	APL Array Notation
JS	JavaScript for inclusion in code
JSON	JavaScript Object Notation



]OUTPUT.Repr

]repr <i>expression</i> -format=	APL	single-line APL expression
	APLAN	APL Array Notation
	JS	JavaScript for inclusion in code
	JSON	JavaScript Object Notation
	XML	Extensible Markup Language



]OUTPUT.Repr

]repr <i>expression</i> -format=	APL	single-line APL expression
	APLAN	APL Array Notation
	JS	JavaScript for inclusion in code
	JSON	JavaScript Object Notation
	XML	Extensible Markup Language
	CSV	Comma-Separated Values

,



]OUTPUT.Repr

<code>]repr <i>expression</i> -format=</code>	single-line APL expression
<code>APLAN</code>	APL Array Notation
<code>JS</code>	JavaScript for inclusion in code
<code>JSON</code>	JavaScript Object Notation
<code>XML</code>	Extensible Markup Language
<code>CSV</code>	Comma-Separated Values ,
<code>SSV</code>	Semicolon-Separated Values ;



]OUTPUT.Repr

]repr <i>expression</i> -format=		
APL	single-line APL expression	
APLAN	APL Array Notation	
JS	JavaScript for inclusion in code	
JSON	JavaScript Object Notation	
XML	Extensible Markup Language	
CSV	Comma-Separated Values	,
SSV	Semicolon-Separated Values	;
PSV	Pipe-Separated Values	



]OUTPUT.Repr

]repr <i>expression</i> -format=		APL	single-line APL expression
		APLAN	APL Array Notation
		JS	JavaScript for inclusion in code
		JSON	JavaScript Object Notation
		XML	Extensible Markup Language
		CSV	Comma-Separated Values ,
		SSV	Semicolon-Separated Values ;
		PSV	Pipe-Separated Values
		TSV	Tab-Separated Values ⇄



]OUTPUT.Repr

]repr *expression* -format=*format*



]OUTPUT.Repr

```
]repr expression -format=format -outfile=filename
```



]OUTPUT.Repr

```
]repr ι1 2
```



]OUTPUT.Repr

```
]repr ι1 2  
(1 2ρ(2ρ1) (1 2))
```



]OUTPUT.Repr

```
]repr ι1 2  
(1 2ρ(2ρ1) (1 2))
```

```
]repr +÷1[≠
```



]OUTPUT.Repr

```
]repr ι1 2  
(1 2ρ(2ρ1) (1 2))
```

```
]repr +÷1⌈≠  
(+÷)÷(1⌈≠)
```



]OUTPUT.Repr

```
]repr ι1 2  
(1 2ρ(2ρ1) (1 2))
```

```
]repr +÷1⌈≠  
(+÷)÷(1⌈≠)
```

```
Merge←ιö≠ö¬[]ο⊃ö0[]ö0 99
```



]OUTPUT.Repr

```
]repr ι1 2  
(1 2ρ(2ρ1) (1 2))
```

```
]repr +÷1⌈≠  
(+÷)÷(1⌈≠)
```

```
Merge←ιö≠ö¬[]ο≡ö0[]ö0 99  
]repr Merge
```



]OUTPUT.Repr

```
]repr ι1 2  
(1 2ρ(2ρ1) (1 2))
```

```
]repr +÷1⌈≠  
(+÷)÷(1⌈≠)
```

```
Merge←ιö≠ö¬[]◦⊃ö0[]ö0 99  
]repr Merge  
Merge←((ιö≠)ö¬)(([]◦⊃)ö0)([]ö0 99)
```



]OUTPUT.Repr

```
]repr ι1 2 -f=aplan
```



]OUTPUT.Repr

```
]repr ι1 2 -f=aplan  
[  
  (  
    1 1  
    1 2  
  )  
]
```



]OUTPUT.Repr

Hook $\leftarrow \{\alpha \leftarrow \omega \quad \diamond \quad \omega \quad \omega \omega \ddot{\sim} \alpha \alpha \quad \omega\}$



]OUTPUT.Repr

```
Hook←{α←ω  ⋄  ω  ωω~αα  ω}  
]repr Hook
```



]OUTPUT.Repr

```
Hook←{α←ω ⋄ ω ωω~αα ω}  
]repr Hook  
□FX(,c' Hook←{α←ω ⋄ ω ωω~αα ω}' )
```



]OUTPUT.Repr

```
Hook←{α←ω ⋄ ω ωω~αα ω}
```

```
]repr Hook
```

```
□FX(,c' Hook←{α←ω ⋄ ω ωω~αα ω}' )
```

```
]repr Hook -f=aplan
```



]OUTPUT.Repr

```
Hook←{α←ω ◇ ω ωω~αα ω}  
]repr Hook  
□FX(,c' Hook←{α←ω ◇ ω ωω~αα ω}' )  
]repr Hook -f=aplan  
Hook←{α←ω ◇ ω ωω~αα ω}
```



]OUTPUT.Repr

'ns.sub'␣NSθ



]OUTPUT.Repr

```
'ns.sub'⊞NSθ  
ns.sub.val←,42
```



]OUTPUT.Repr

```
'ns.sub'⊞NS⊥  
ns.sub.val←,42  
]repr ns -f=aplan
```



]OUTPUT.Repr

```
      'ns.sub'␣NS⌀  
      ns.sub.val←,42  
      ]repr ns -f=aplan  
(  
  sub:(  
    val:(42⋄)  
  )  
)
```



]OUTPUT.Repr

```
'ns.sub'␣NS⊖  
ns.sub.val←,42  
]repr ns -f=js
```



]OUTPUT.Repr

```
'ns.sub'␣NS⌀  
ns.sub.val←,42  
]repr ns -f=js  
{  
  sub: {  
    val: [42],  
  },  
}
```



]OUTPUT.Repr

```
'ns.sub'␣NS␣  
ns.sub.val←,42  
]repr ns -f=js  
{  
  sub: {  
    val: [42],  
  },  
}
```

```
]repr ns -f=json
```



]OUTPUT.Repr

```
'ns.sub'␣NS␣  
ns.sub.val←,42  
]repr ns -f=js  
  
{  
  sub: {  
    val: [42],  
  },  
}
```

```
]repr ns -f=json  
  
{  
  "sub": {  
    "val": [42]  
  }  
}
```



]OUTPUT.Repr

```
'ns.sub'␣NS␣  
ns.sub.val←,42  
]repr ns -f=js
```

```
{  
  sub: {  
    val: [42],  
  },  
}
```

```
]repr ns -f=json
```

```
{  
  "sub": {  
    "val": [42]  
  }  
}
```



]OUTPUT.Repr

```
]repr 3 3pi9 -f=xml  
<array>  
  <shape>3 3</shape>  
  <list>1 2 3 4 5 6 7 8 9</list>  
</array>
```



]OUTPUT.Repr

```
]repr 3 3p19 -f=csv
```

```
1,2,3
```

```
4,5,6
```

```
7,8,9
```



]OUTPUT.Repr

```
]repr 3 3p19 -f=ssv
```

```
1;2;3
```

```
4;5;6
```

```
7;8;9
```



`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

+ Ideas for the Future



]EXPERIMENTAL.Get

“get pretty much anything from pretty much anywhere”



]EXPERIMENTAL.Get

“get pretty much anything from pretty much anywhere”

- One-stop utility for quickly getting bringing resources in.



]EXPERIMENTAL.Get

“get pretty much anything from pretty much anywhere”

- One-stop utility for quickly getting bringing resources in.
- Do not use at run time, as exact results may vary.



]EXPERIMENTAL.Get

Syntax

```
]get something
```



]EXPERIMENTAL.Get

Syntax

```
]get something
```

```
#.something
```



]EXPERIMENTAL.Get

Plain name heuristics



]EXPERIMENTAL.Get

Plain name heuristics

]get dfns workspace from WS path
#.dfns



]EXPERIMENTAL.Get

Plain name heuristics

<code>]get dfns</code>	workspace from WS path
<code>#.dfns</code>	
<code>]get HttpCommand</code>	source file from SALT path
<code>#.HttpCommand</code>	



]EXPERIMENTAL.Get

Plain name heuristics

]get dfns	workspace from WS path
#.dfns	
]get HttpCommand	source file from SALT path
#.HttpCommand	
]get tools	directory relative to current
#.tools	



]EXPERIMENTAL.Get

File extension detection



]EXPERIMENTAL.Get

File extension detection

]get MyFn.aplf Link-type source files
#.MyFn



]EXPERIMENTAL.Get

File extension detection

]get MyFn.aplf Link-type source files
#.MyFn

]get myVar.apla APL Array Notation files
#.myVar



]EXPERIMENTAL.Get

File extension detection

]get MyFn.aplf Link-type source files
#.MyFn

]get myVar.apla APL Array Notation files
#.myVar

]get text.charmat Acre-type source files
#.text



]EXPERIMENTAL.Get

Data format conversion



]EXPERIMENTAL.Get

Data format conversion

]get obj.json JS, JSON, JSON5, config files
#.obj



]EXPERIMENTAL.Get

Data format conversion

<code>]get obj.json</code>	JS, JSON, JSON5, config files
<code>#.obj</code>	
<code>]get table.csv</code>	Comma Separated Values
<code>#.table</code>	



]EXPERIMENTAL.Get

Data format conversion

<code>]get obj.json</code> <code>#.obj</code>	JS, JSON, JSON5, config files
<code>]get table.csv</code> <code>#.table</code>	Comma Separated Values
<code>]get data.xml</code> <code>#.data</code>	eXtensible Markup Language



]EXPERIMENTAL.Get

Retrieval



]EXPERIMENTAL.Get

Retrieval

```
]get https://example.com/index.html  
#.index
```



]EXPERIMENTAL.Get

Retrieval

```
]get https://example.com/index.html  
#.index
```

```
]get https://github.com/abrudz/Kbd  
#.Kbd
```



]EXPERIMENTAL.Get

Protocols



]EXPERIMENTAL.Get

Protocols

```
]get https://github.com/abrudz/Kbd  
#.Kbd
```



]EXPERIMENTAL.Get

Protocols

```
]get https://github.com/abrudz/Kbd  
#.Kbd
```

```
]get ftp://ftp.software.ibm.com/software  
#.foo
```



]EXPERIMENTAL.Get

Protocols

```
]get https://github.com/abrudz/Kbd  
#.Kbd
```

```
]get ftp://ftp.software.ibm.com/software  
#.foo
```

```
]get github.com/abrudz/Kbd  
#.Kbd
```



]EXPERIMENTAL.Get

Unpacking



]EXPERIMENTAL.Get

Unpacking

```
]get main.zip  
#.main
```



]EXPERIMENTAL.Get

Unpacking

```
]get main.zip
```

```
#.main
```

```
]get github.com/user/repo/tree/main/sub
```

```
#.sub
```



]EXPERIMENTAL.Get

Unpacking



]EXPERIMENTAL.Get

Unpacking

```
]get dfns -only=dx,b,dab  
#.dfns
```



]EXPERIMENTAL.Get

Unpacking

```
]get dfns -only=dxb,dab  
#.dfns  
    ]map  
#  
·    dfns  
·    ·    ▽ dab dxb
```



]EXPERIMENTAL.Get

Unpacking



]EXPERIMENTAL.Get

Unpacking

```
        )clear  
clear ws
```



]EXPERIMENTAL.Get

Unpacking

```
        )clear  
clear ws  
        ]get dfns -only=dxb,dab -unpack  
#.dxb #.dab
```



]EXPERIMENTAL.Get

Unpacking

```
        )clear
clear ws
        ]get dfns -only=dxb,dab -unpack
#.dxb #.dab
        ]map
#
.      ▽ dab dxb
```



]EXPERIMENTAL.Get

Options



]EXPERIMENTAL.Get

Options

```
]get obj.json:txt
```

```
#.obj
```



]EXPERIMENTAL.Get

Options

```
]get obj.json:txt
```

```
#.obj
```

```
10↑obj
```

```
{"myKey":1
```



]EXPERIMENTAL.Get

Options

```
]get obj.json:txt
```

```
#.obj
```

```
10↑obj
```

```
{"myKey":1
```

```
]get thing -target=□SE.there
```



]EXPERIMENTAL.Get

Options

```
]get obj.json:txt
```

```
#.obj
```

```
10↑obj
```

```
{"myKey":1
```

```
]get thing -target=□SE.there
```

```
#.SE.there.thing
```



]EXPERIMENTAL.Get

Options



]EXPERIMENTAL.Get

Options

```
]get /tmp/repo -sync  
#.repo
```



]EXPERIMENTAL.Get

Options

```
]get /tmp/repo -sync  
#.repo  
]link.status
```



]EXPERIMENTAL.Get

Options

```
]get /tmp/repo -sync  
#.repo
```

```
]link.status
```

Namespace		Directory	Files
#.repo	↔	c:/tmp/repo	14



Overview

`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

+ Ideas for the Future



Ideas for the Future

Current issues:



Ideas for the Future

Current issues:

- ◆ SALT dependency



Ideas for the Future

Current issues:

- ❖ SALT dependency
- ❖ Performance



Ideas for the Future

Current issues:

- ✧ SALT dependency
- ✧ Performance
- ✧ Cache file invalidation



Ideas for the Future

Current issues:

- ✧ SALT dependency
- ✧ Performance
- ✧ Cache file invalidation
- ✧ Abuse for utility functions



Ideas for the Future

Current issues:

- ✧ SALT dependency
- ✧ Performance
- ✧ Cache file invalidation
- ✧ Abuse for utility functions
- ✧ Inflexibility



Ideas for the Future

Current issues:

- ✧ SALT dependency
- ✧ Performance
- ✧ Cache file invalidation
- ✧ Abuse for utility functions
- ✧ Inflexibility
- ✧ Old philosophy



Ideas for the Future

My ideas:



Ideas for the Future

My ideas:

- Based on SessionStartup



Ideas for the Future

My ideas:

- Based on SessionStartup
- Directories \Leftrightarrow namespaces \Leftrightarrow groups



Ideas for the Future

My ideas:

- Based on SessionStartup
- Directories \Leftrightarrow namespaces \Leftrightarrow groups
- Any function can declare itself as a user command



Ideas for the Future

My ideas:

- ✧ Based on SessionStartup
- ✧ Directories \Leftrightarrow namespaces \Leftrightarrow groups
- ✧ Any function can declare itself as a user command
- ✧ Standard format



Ideas for the Future

My ideas:

- Based on SessionStartup
- Directories \Leftrightarrow namespaces \Leftrightarrow groups
- Any function can declare itself as a user command
- Standard format

Left argument:
namespace of options
including utilities
Right argument:
argument raw/list



Ideas for the Future

My ideas:

- ◆ Based on SessionStartup
- ◆ Directories \Leftrightarrow namespaces \Leftrightarrow groups
- ◆ Any function can declare itself as a user command
- ◆ Standard format
- ◆ Framework searches \square SE

Left argument:
namespace of options
including utilities
Right argument:
argument raw/list



Ideas for the Future

Benefits:



Ideas for the Future

Benefits:

- One API function for every user command



Ideas for the Future

Benefits:

- One API function for every user command
- Easy to package, distribute, install



Ideas for the Future

Benefits:

- One API function for every user command
- Easy to package, distribute, install
- Can be part of a larger package



Ideas for the Future

Benefits:

- One API function for every user command
- Easy to package, distribute, install
- Can be part of a larger package
- Can co-exist with utilities and dependencies



Ideas for the Future

Benefits:

- One API function for every user command
- Easy to package, distribute, install
- Can be part of a larger package
- Can co-exist with utilities and dependencies
- Stability: Group name required under program control



Ideas for the Future

Benefits:

- One API function for every user command
- Easy to package, distribute, install
- Can be part of a larger package
- Can co-exist with utilities and dependencies
- Stability: Group name required under program control
- Customisability: Change group by renaming a directory



Questions?

`]OUTPUT.View`

`]TOOLS.APLCart`

`]WS.Names`

`]OUTPUT.Repr`

`]EXPERIMENTAL.Get`

+ Ideas for the Future

Adám Brudzewsky
adam@dyalog.com

