

Tool Talk

Rex Swain
Independent Consultant
8 South Street
Washington, CT 06793
Tel 860-868-0131
rex@rexswain.com

Eighth Annual APL2000 User Conference

November 2003

To install:

```
]ufile [1.5] ucmdsrex
```

Required:

```
)EVLEVEL 2
```

Help:

```
]foo ?
```

If you have version 6.5 of the Araxis Merge product, you will need to change the JDIFF command slightly. (In version 6.5 Araxis changed the name of their application from "Merge2000" to "Merge65".) Here's how to fix the user command function:

```
]uload CMDDIFF /r /f=ucmdsrex  
]  
)edit CMDDIFF
```

Change...

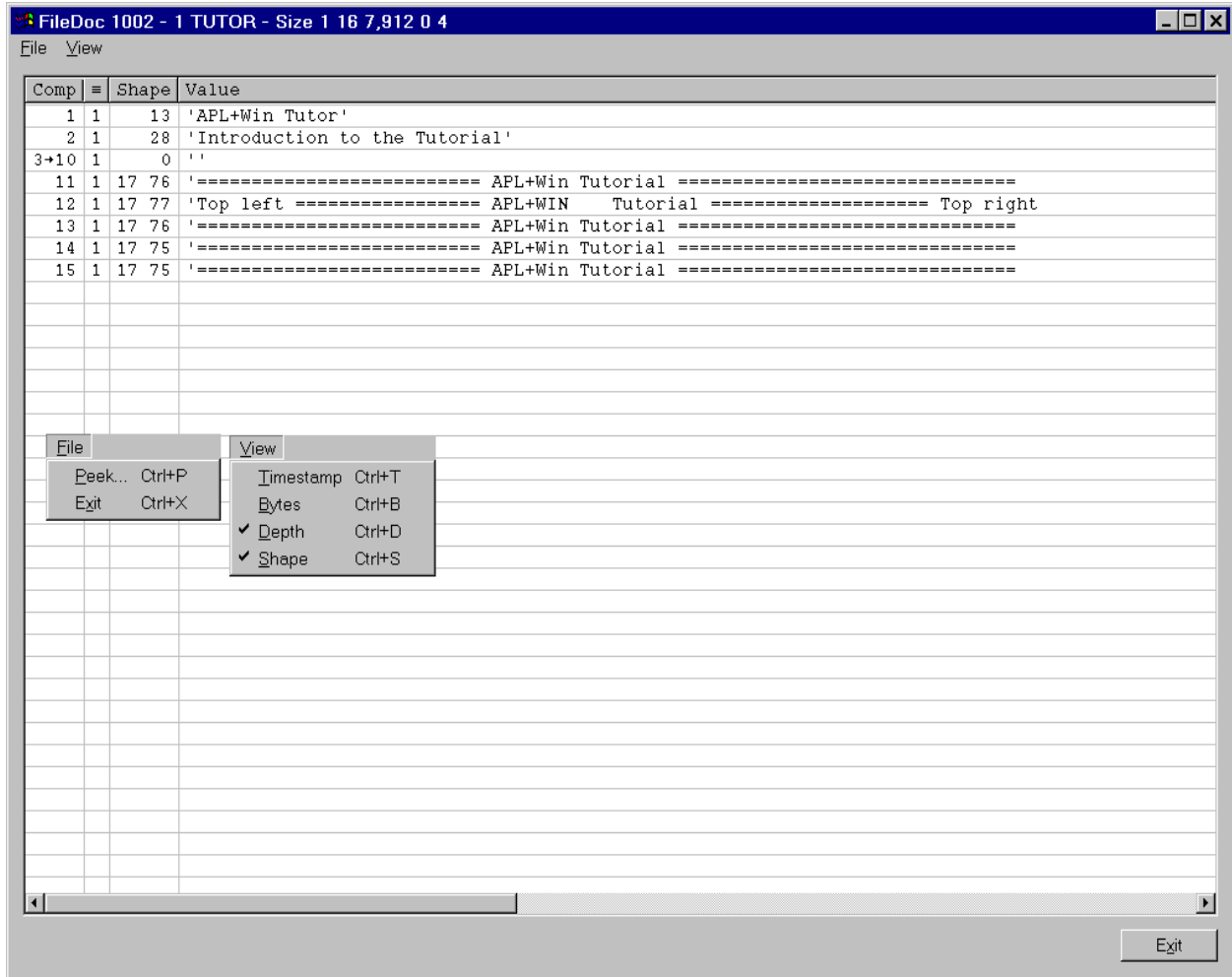
```
[340] □wself←'DIFF_AM' □wi 'Create' 'ActiveObject Merge2000.Application'
```

To...

```
[340] □wself←'DIFF_AM' □wi 'Create' 'ActiveObject Merge65.Application'
```

```
]usave CMDDIFF /r /f=ucmdsrex
```

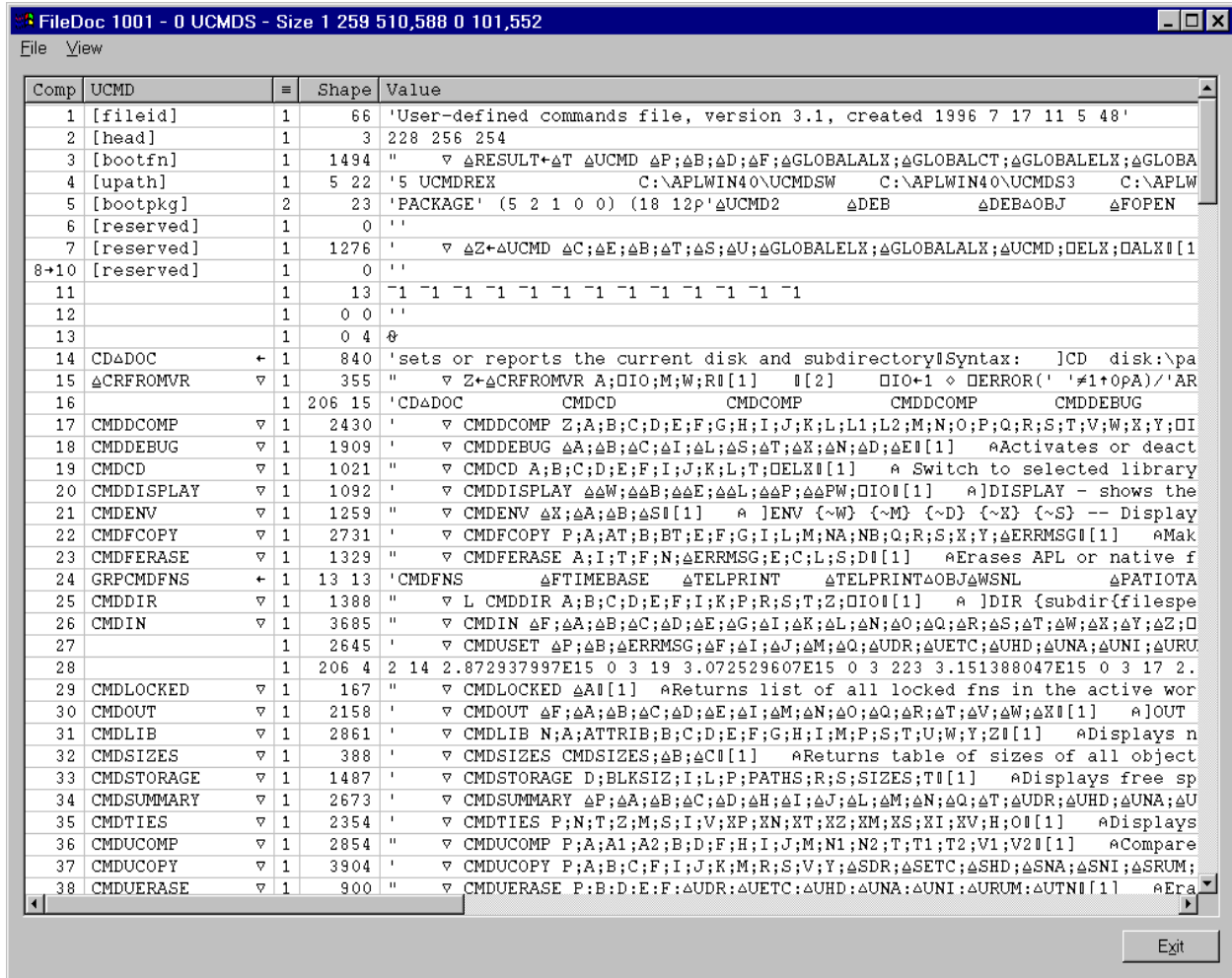
lfiledoc 1 tutor



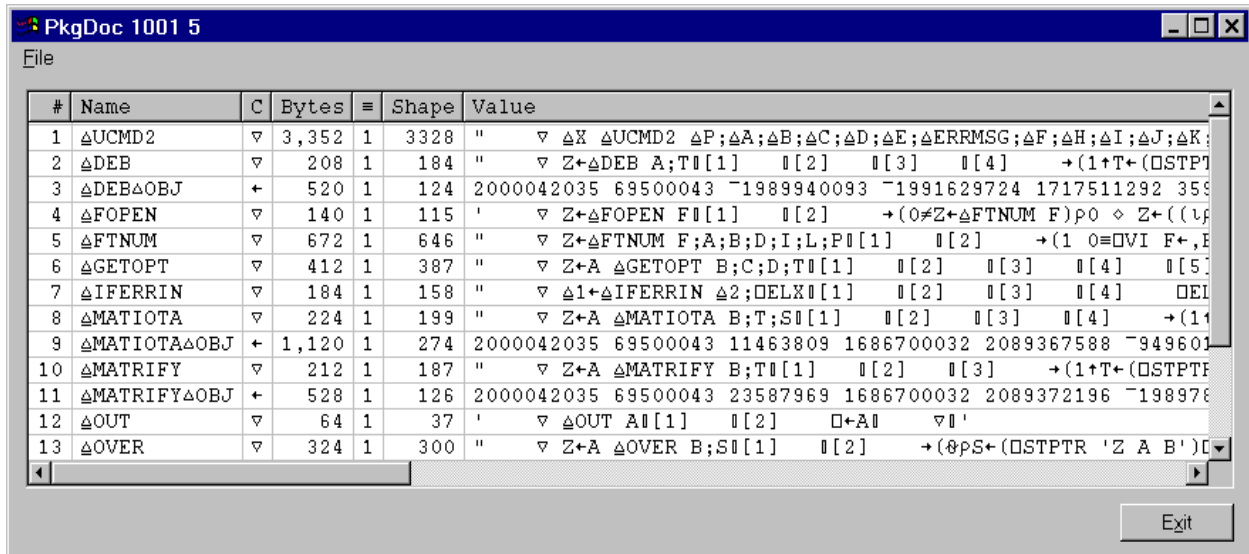
A Double-click component 11



lfiledoc 0 ucmds



A Double-click component 5



]wsloc ΔRemember

```

Ini[18]      :if ΔRemember      A only if option turned on
register[0]   register;ΔRemember;c;q;s
register[3]   ΔRemember←1      A make sure we remember these Ini settings
tkh[616]     A ΔRemember used by ∇Ini
tkh[617]     q←~ΔRemember
tkh[618]     ΔRemember←1      A make sure we remember this one last value
tkh[621]     ΔRemember←q
tkh[625]     q←ΔRemember
tkh[626]     ΔRemember←1      A make sure Ini works
tkh[630]     ΔRemember←q
unregister[0] unregister;s;q;a;ΔRemember
unregister[23] ΔRemember←1

```

]notlocal ΔRemember

Searching 320 functions ...

```

==> Ini (1)
Ini[18]      :if ΔRemember      A only if option turned on
==> tkh (6)
tkh[617]     q←~ΔRemember
tkh[618]     ΔRemember←1      A make sure we remember this one last value
tkh[621]     ΔRemember←q
tkh[625]     q←ΔRemember
tkh[626]     ΔRemember←1      A make sure Ini works
tkh[630]     ΔRemember←q

```

]notlocal □io

Searching 320 functions ...

```

==> AlwaysOnTop (1)
AlwaysOnTop[5] b←(□io+b)▷'hwnd_notopmost' 'hwnd_topmost'
==> AutoResize (4)
AutoResize[15] w[;□io]←w[;□io]×(↑pw)ρs
AutoResize[37] w[;□io]←w[;□io]×(↑pw)ρs
==> FFAW (3)
FFAW[87]      P←' ,'[□IO+P]      A Else space before A 07 Jul 2002
FFAW[110]     F←F-D×□IO          A Adjust first for display origin
FFAW[121]     →(□IO+2↑''''∈A)▷L90,L91,L92,L93
==> GetWinVer (2)
GetWinVer[17] struc[3+□IO]←65536↑struc[3+□IO]
==> goo (1)
goo[2]        □io←1 ◇ foo

```

]notlocal □io←

Searching 320 functions ...

```

==> goo (1)
goo[2]        □io←1 ◇ foo

```

]notlocal Δbdat←

Searching 320 functions ...

```

==> sell (1)
sell[63]      (Δba Δsa Δbdat Δsdat)←(ba sa bdat sdat) A save copies

```

```
lvarss prox  
ΔProxyHost ΔProxyPass ΔProxyPort ΔProxyUse ΔProxyUser
```

```
lvarss dir;path  
AAmixdir ΔAAmixdir ΔExeDir ΔPath ΔSysDir ΔWinDir
```

```
lfnss help  
HtmlHelp help help1 help2 helpg helpw  
lfnss Help /c  
HtmlHelp
```

```
lfnss num  
EditNum EditNumFmt EditNumVifi FMTNUM GETNUMFMT  
lfnss num ;  
EditNum FMTNUM
```

```
lvarfind 'Fixed'
```

```
Searching 880 variables for 'Fixed'
((c1 1)AAmixdir) ← '100% Fixed Income'
griddoc <162> ← 'xFixedCols property:'
griddoc <163> ← " Value@Long ← DWI 'xFixedCols'"
griddoc <164> ← " DWI 'xFixedCols' Value@Long"
griddoc <169> ← 'xFixedRows property:'
griddoc <170> ← " Value@Long ← DWI 'xFixedRows'"
griddoc <171> ← " DWI 'xFixedRows' Value@Long"
```

```
lvarfind 10
```

```
Searching 880 variables for 10
AAptiles ← 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
BBS_RAISED ← 10
BORDER_RAISED ← 10
CHART_Y2LEGFT ← 10
CPI_3DINFO ← 10
CSA_XGAP ← 10
HIT_RIGHTTITLE ← 10
MK_CUBE ← 10
RIFdata[1;] ← 10 1.5
SURFACE ← 10
(14 2 (13 4) 1>VARG0) ← ,10
(15 2 (13 4) 1>VARG2) ← ,10
(63 2 (13 4) 1>VARG2) ← ,10
```

```
lvarfind 1 2
```

```
Searching 880 variables for 1 2
AAmixes ← 1 2 3
EFororder[2;] ← 1 2 'Market Drop'
PPorder[2;] ← 1 2 'Medium Growth'
```

DOMAIN ERROR

```
TextSizePix[4] z←1▷□wi 'Draw' ('Scale' 5-1) ('?Text' t)
                ^
```

```
)si
```

```
TextSizePix[4] ★
```

```
chart3a[219]
```

```
sel3[329]
```

```
⊕
```

```
sel0[102]
```

```
>[fmTK.fMain.sMain.pgHome.fWork.wb;XBeforeNavigate2]
```

```
]si
```

```
TextSizePix[4] ★ z←1▷□wi 'Draw' ('Scale' 5-1) ('?Text' t)
```

```
chart3a[219] (h w)←p TextSizePix (i▷x1) A p is the chart's parent frame
```

```
sel3[329] chart3a 0 A 0=make new chart from scratch
```

```
⊕
```

```
sel0[102] ⊕x
```

```
>[fmTK.fMain.sMain.pgHome.fWork.wb;XBeforeNavigate2]
```

lsymfind clip

The screenshot shows the SymFind clip application window. The main window title is "SymFind clip" and it has a menu bar with "File", "Edit", "View", and "Actions". The main area is a table with columns: Lib, WSID, UFile, C, Name, Shape, and Value. The table lists 96 matching names across 21 workspaces and 2 files. Several context menus are open over the table:

- File menu:** Exit (Ctrl+X)
- Edit menu:** Select All (Ctrl+A), Select None (Ctrl+N), Invert Selection (Ctrl+I)
- View menu:**
 - WSID (checked, Ctrl+W)
 - UFile (checked, Ctrl+U)
 - Timestamp (Ctrl+T)
 - Bytes (Ctrl+B)
 - Depth (checked, Ctrl+E)
 - Shape (checked, Ctrl+S)
 - Value (checked, Ctrl+L)
- Actions menu:**
 - Peek (Ctrl+P)
 - Get Info (Ctrl+G)
 - Get Info for All (Alt+G)
 - Diff (Ctrl+D)
 - Remove from list (Del)
 - Functions only (Ctrl+F)
 - Variables only (Ctrl+V)

At the bottom of the window, there is a status bar with the text: "Found 96 matching names in 21 of 137 workspaces" and "Found 11 matching names in 2 of 5 files". There are also "Get All Info" and "Exit" buttons in the bottom right corner.

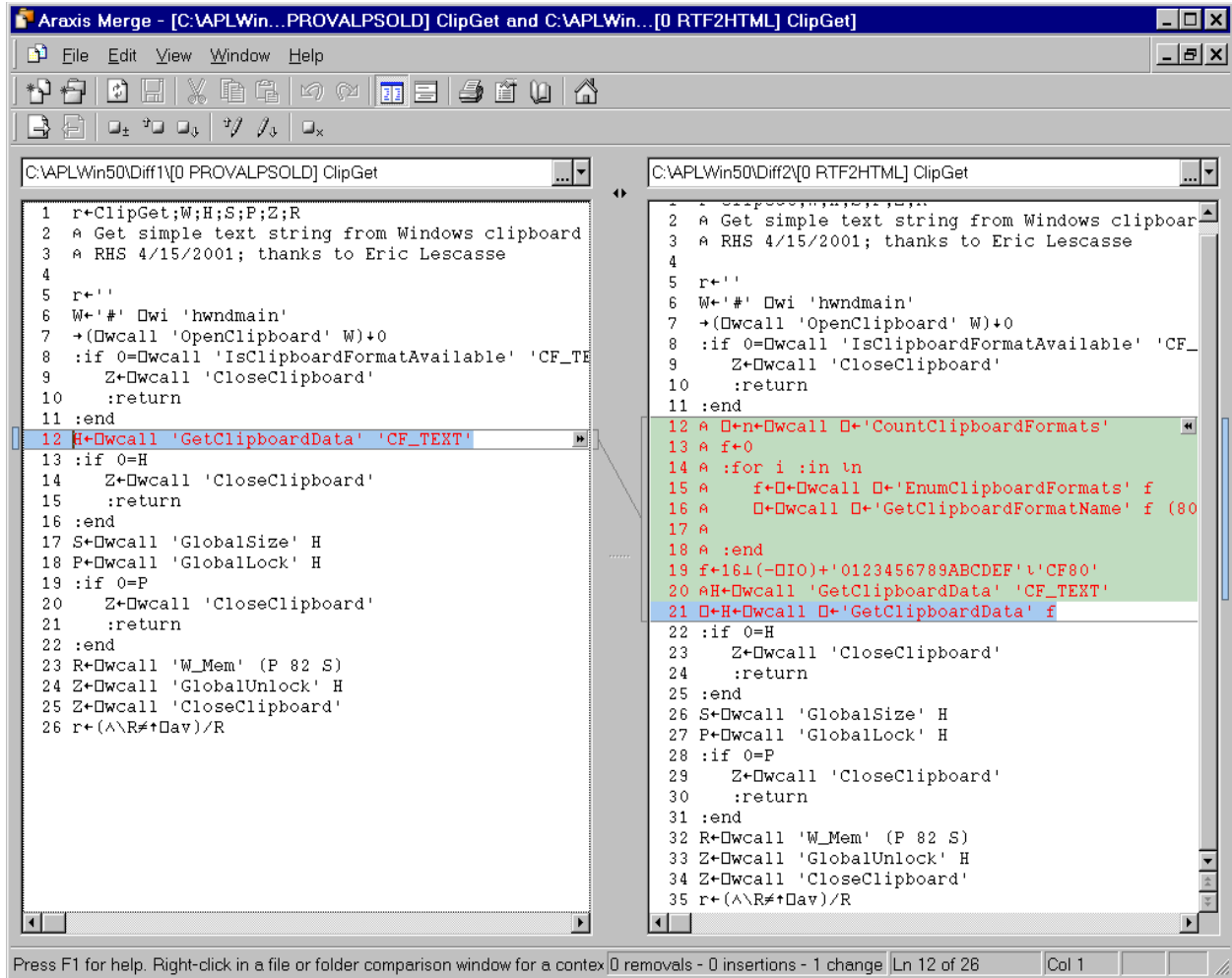
Lib	WSID	UFile	C	Name	Shape	Value
5	PS2		▽	ClipGet2		
5	PS2		▽	ClipGet3		
5	PS2		▽	ClipGetFmt		
5	PSSERVER		▽	CLIPIT		
5	PSServer.old		▽	CLIPIT		
5	PSServer.old		▽	IFCLIPATYPE		
5	PSServer.old		▽	wSETCLIP		
5	PSServer.old		▽	wGETCLIP		
5	UCMDREX		▽	CMDCLIPDOC		
5	UCMDREX		+	GRPCMDCLIPDOC		
5	UCMDREX		+	CLIPDOCΔDOC		
9	EleClipboard		▽	TClipBoard		
22	Source		+	AS_NOTCLIPPED		
22	Source		+	CL_NOTCLIPPED		
22	Source		+	CFX_ID_EXPORTCLIPBOARD		
22	Source		▽	ClipCopy		
22	Source		▽	ClipGet		
22	Source		▽	ClipGet2		
22	Source		▽	ClipGet3		
22	Source		▽	ClipGetFmt		
219	PROVAL		▽	CLIPIT		
219	PROVAL		▽	IFCLIPATYPE		
219	PROVAL		▽	wSETCLIP		
219	PROVAL		▽	wGETCLIP		
5		UCMDREX	+	CLIPDOCΔDOC		
5		UCMDREX	▽	CMDCLIPDOC		
5		UCMDREX	+	GRPCMDCLIPDOC		
C:\APLWIN40		UCMDSW	+	CLIPCOPYΔDOC		
C:\APLWIN40		UCMDSW	▽	CLIPFmt		
C:\APLWIN40		UCMDSW	+	CLIPPASTEΔDOC		
C:\APLWIN40		UCMDSW	▽	CLIPUnfmt		
C:\APLWIN40		UCMDSW	▽	CMDCLIPCOPY		
C:\APLWIN40		UCMDSW	▽	CMDCLIPPASTE		
C:\APLWIN40		UCMDSW	+	GRPCMDCLIPCOPY		
C:\APLWIN40		UCMDSW	+	GRPCMDCLIPPASTE		

The screenshot shows the SymFind clip application window. The title bar reads "SymFind clip". Below the title bar is a menu bar with "File", "Edit", "View", and "Actions". The main area contains a table with the following columns: Lib, WSID, UFile, C, Name, Shape, and Value. The table lists various symbols from different libraries, including UCMDREX, UCMSW, PICUTIL, hotkey, PROVALPS, REXPLAY, RTF2HTML, DEMO, REL40, WINDOWS, and Source. A context menu is open over the entry "ClipGet" from the "PROVALPSOLD" library, showing options: Peek, Get Info, Diff, and Remove from list. At the bottom of the window, there is a status bar with the text "Found 96 matching names in 21 of 137 workspaces" and "Found 11 matching names in 2 of 5 files". There are also "Get All Info" and "Exit" buttons.

Lib	WSID	UFile	C	Name	Shape	Value
5		UCMDREX	▽	CMDCLIPDOC	-	2633
C:\APLWIN40		UCMSW	▽	CMDCLIPPASTE	-	1160
0	PICUTIL		▽	CaptureToClip	-	1067
2	picutil		▽	CaptureToClip	-	1067
4	hotkey		▽	Clip		
0	PROVALPS		▽	ClipCopy	-	1114
0	PROVALPSOLD		▽	ClipCopy	-	1114
0	REXPLAY		▽	ClipCopy	-	1120
0	RTF2HTML		▽	ClipCopy	-	1114
5	DEMO		▽	ClipCopy	-	1114
5	PROVALPS		▽	ClipCopy	-	1114
5	PS2		▽	ClipCopy	-	1114
22	Source		▽	ClipCopy	-	1114
1	REL40		▽	ClipFmt	-	803
3	WINDOWS		▽	ClipFmt	-	780
3	WINDOWS		+	ClipFmtΔDOC	1	3518
0	PROVALPS		▽	ClipGet	-	1189
0	PROVALPSOLD		▽	ClipGet	-	1189
0	RTF2HTML		▽	ClipGet	-	1541
5	DEMO		▽	ClipGet	-	1189
5	PROVALPS		▽	ClipGet	-	1189
5	PS2		▽	ClipGet	-	1189
22	Source		▽	ClipGet	-	1189
0	PROVALPS		▽	ClipGet2	-	2013
0	PROVALPSOLD		▽	ClipGet2	-	2013
5	DEMO		▽	ClipGet2	-	2013
5	PROVALPS		▽	ClipGet2	-	2013
5	PS2		▽	ClipGet2	-	2013
22	Source		▽	ClipGet2	-	2013
0	PROVALPS		▽	ClipGet3	-	1223
0	PROVALPSOLD		▽	ClipGet3	-	1223
5	DEMO		▽	ClipGet3	-	1223
5	PROVALPS		▽	ClipGet3	-	1223
5	PS2		▽	ClipGet3	-	1223
22	Source		▽	ClipGet3	-	1223

Found 96 matching names in 21 of 137 workspaces
 Found 11 matching names in 2 of 5 files

Get All Info Exit



]diff ?

Compare (show the DIFFerences between) two functions or variables

Syntax:]DIFF OBJECT1 ; OBJECT2

Where each object may be:

Active ws:

FOO A object name (function or variable)

Saved ws:

UTILS FOO A wsid, object name

9 STUFF FOO A library number, wsid, object name

\EUDORA\ATTACH\UTIL FOO A path, wsid, object name

User command file:

FOO /F=UCMDS A object name and ucmd file to get it from

FOO /F=19 UCMDS A object name and ucmd file with lib number

FOO /F A object name, from first]UFILE file

APL file component:

tn,cn A tie number, component number

tn#cn A tie number, component number

You may use "=" for the second wsid or object name

Example:]DIFF FOO;UTILS =

Alternate syntax (borrowed from]COMP) with no semicolon:

]DIFF object1 object2

Each object may be one of the following types:

fname [/F[=cmdfile]] A function

varname [/F[=cmdfile]] A variable

tn,cn A APL file: tie number, component number

tn#cn A APL file: tie number, component number

Examples:

]DIFF FUN GUN

]DIFF FUN FUN /F=5 UCMDS

]DIFF FUN /F=5 UCMDS GUN

]DIFF FUN /F=5 UCMDS GUN /F=9 MYUCMDS

You may use "=" for the second object name

Note that you may not specify WSIDs when using this no-semicolon syntax

Execute]DIFF /ERASE occasionally to clean up temp files

(Sorry, this feature does not work with Win98)

You may establish a list of user command files with

]DIFF /ULIST=file1;file2;file3

and query it with

]DIFF /ULIST

Then use

]DIFF FOO;= /U

/U triggers a search through the list of files established above for the first object named FOO

Requires: Araxis Merge 2001, Standard Edition, version 6.0

see <<http://www.araxis.com/merge>>

You must set the APL font manually:

View - Options - Fonts and Colours

Then set all four fonts to something like: APLFONT, Regular, 11

Requires: DEB DLB DTB ANFILE

After similar by JVM and Patrick Parks

06 Apr 2003 Rex Swain, Independent Consultant, www.rexswain.com

Tool Talk 2

Rex Swain
Independent Consultant
8 South Street
Washington, CT 06793
Tel 860-868-0131
Fax 860-868-9970
rex@rexswain.com

<http://www.rexswain.com>

Ninth Annual APL2000 User Conference

November 2004

Notes

To add UCMSREX.SF to your user command files path:

```
]UFILE [1.5] UCMSREX
```

If you have UCMSREX.SF from the last year's conference, just replace it with this one; the tools from last year are all here too.

These tools require:

```
)EVLEVEL 2
```

Help:

```
]foo ?
```

Font: You may edit in your APLW.INI file; the defaults are:

```
[UCMSREX]  
FontName=APLFONT  
FontSize=11  
FontStyle=0  
FontCharset=default
```

Before using `fnreplx`, copy file `APL_PCRE.DLL` into your APLW directory.

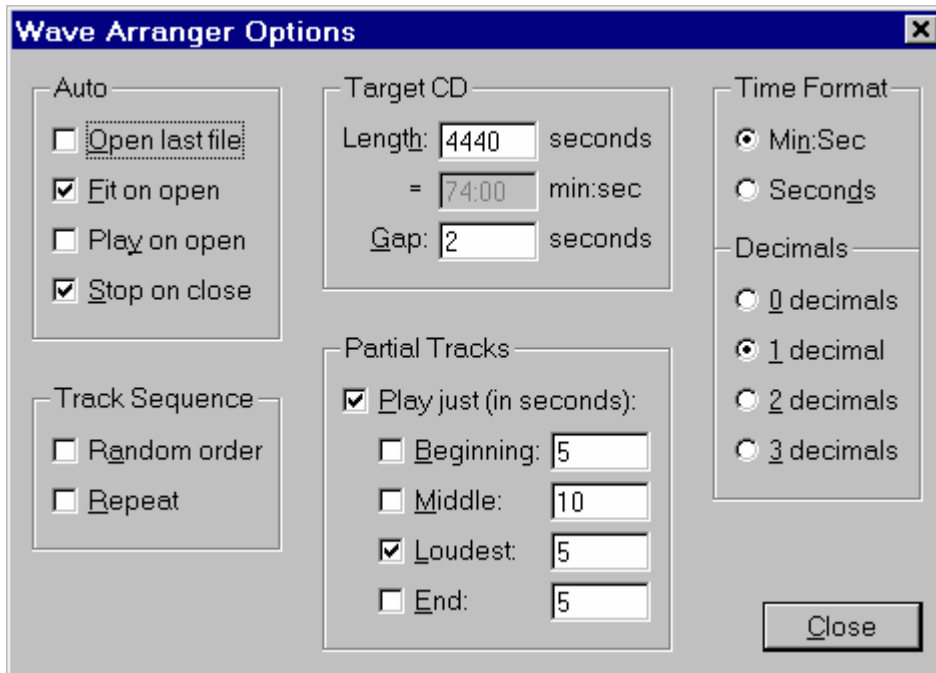
`fnreplx` is not a user command; load it into your active ws:

```
]uload fnreplx /f=ucmsrex
```

An excellent book on regular expressions:

Mastering Regular Expressions (2nd edition) by Jeffrey Friedl (O'Reilly, 2002)

Shortcut Keys:



!shortkeys fmWVopts

Shortcut Keys in: fmWVopts [Form]:

Alt+0	Option	fmWVopts.fDec.op0d	&0 decimals
Alt+1	Option	fmWVopts.fDec.op1d	&1 decimal
Alt+2	Option	fmWVopts.fDec.op2d	&2 decimals
Alt+3	Option	fmWVopts.fDec.op3d	&3 decimals
Alt+A	Check	fmWVopts.fTracks.ckRandom	R&andom order
Alt+B	Check	fmWVopts.fPartial.ckSeg1	&Beginning:
Alt+C	Button	fmWVopts.bnClose	&Close
Alt+D	Option	fmWVopts.fTime.opSec	Secon&ds
Alt+E	Check	fmWVopts.fPartial.ckSeg4	&End:
Alt+F	Check	fmWVopts.fAuto.ckFit	&Fit on open
Alt+L	Check	fmWVopts.fPartial.ckSeg3	&Loudest:
Alt+M	Check	fmWVopts.fPartial.ckSeg2	&Middle:
Alt+N	Option	fmWVopts.fTime.opMin	Mi&n:Sec
Alt+O	Check	fmWVopts.fAuto.ckOpen	&Open last file
Alt+P	Check	fmWVopts.fPartial.ckPartial	&Play just (in seconds):
Alt+R	Check	fmWVopts.fTracks.ckRepeat	&Repeat
Alt+S	Check	fmWVopts.fAuto.ckStop	&Stop on close
Alt+Y	Check	fmWVopts.fAuto.ckPlay	Pla&y on open
Ctrl+K	SubMenu	fmWVopts.mSegs	mSegs
Ctrl+T	SubMenu	fmWVopts.mOpts	mOpts
Available: Ctrl+ A B C D E F G H I J L M N O P Q R S U V W X Y Z			
Available: Alt+ G H I J K Q T U V W X Z			

Properties, Events, and Methods:

lpem fmWVopts.fAuto.ckFit

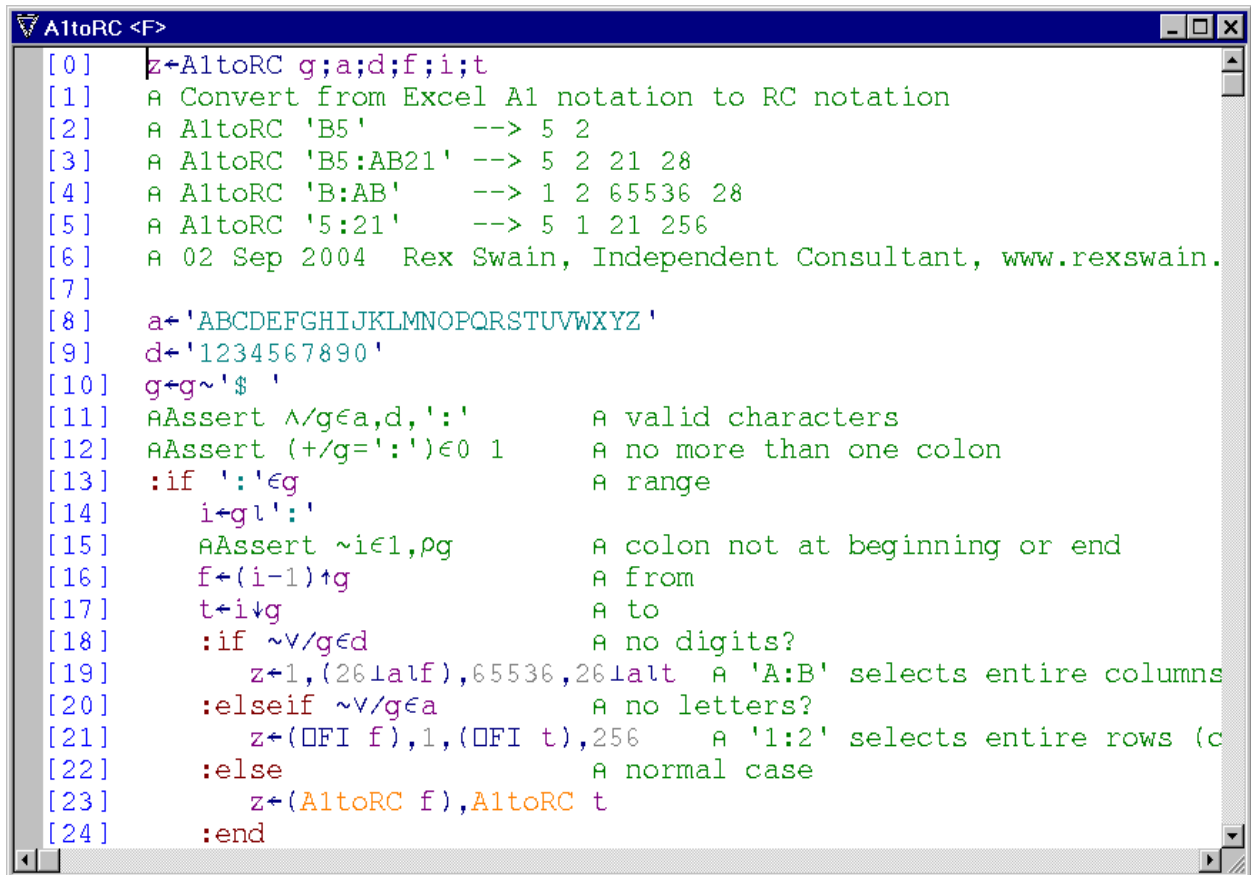
Object: fmWVopts.fAuto.ckFit
 Class: Check
 Properties: border caption children class color data def deferexit
 edge enabled events extent font help helpcontext hwnd
 instance keys links methods mode modified modifystop name
 noredraw opened order place pointer prompt properties
 scale scrollaccel scrollmargin self size state style
 suppress tabgroup tabparent tabstop targetformats tooltip
 tooltipwidth translate value visible where ++ Hwnd Self
 _data dragimage limitwhere nograychildren onAction
 siblings sourceformats transparent wedProps xdef
 Events: Action Click Close Delete Destroy DragEnter DragLeave
 DragOver Drop Exit ExitError Focus Help Hide KeyDown KeyUp
 Modified MouseDouble MouseDown MouseDrag MouseEnter
 MouseLeave MouseMove MouseUp Move Open Paint Reopen Resize
 Send Show Unfocus ++ onDragContinue onDragEnd
 onDragFeedback onDragStart onPreCreate onWindowPosChanged
 onWindowPosChanging
 Methods: Close Create Defer Delete Draw Event Exec Focus Help Hide
 Info Modify New Open Paint Popup Ref Resize Send Set
 SetLinks Show ++ Layout Msg

lpem Button /p

Class: Button
 Properties: border caption children class color data def deferexit
 edge enabled events extent font help helpcontext hwnd
 imageindex imagelist imagespace instance keys links
 methods mode modified modifystop name noredraw opened
 order place pointer prompt properties scale scrollaccel
 scrollmargin self size state style suppress tabgroup
 tabparent tabstop targetformats tooltip tooltipwidth
 translate value visible where ++ Hwnd Self _data dragimage
 limitwhere nograychildren onAction siblings sourceformats
 transparent wedProps xdef

Peek Into Workspace:

```
lpeek 5 xl AltoRC
```



```

AltoRC <F>
[0] z←AltoRC g;a;d;f;i;t
[1] A Convert from Excel A1 notation to RC notation
[2] A AltoRC 'B5' --> 5 2
[3] A AltoRC 'B5:AB21' --> 5 2 21 28
[4] A AltoRC 'B:AB' --> 1 2 65536 28
[5] A AltoRC '5:21' --> 5 1 21 256
[6] A 02 Sep 2004 Rex Swain, Independent Consultant, www.rexswain.
[7]
[8] a←'ABCDEFGHJKLMNOPQRSTUVWXYZ'
[9] d←'1234567890'
[10] g←g~'$ '
[11] AAssert ^/g∈a,d,':' A valid characters
[12] AAssert (+/g=':')∈0 1 A no more than one colon
[13] :if ':'∈g A range
[14] i←gl':'
[15] AAssert ~i∈1,pg A colon not at beginning or end
[16] f←(i-1)↑g A from
[17] t←i↓g A to
[18] :if ~v/g∈d A no digits?
[19] z←1,(26↓a|f),65536,26↓alt A 'A:B' selects entire columns
[20] :elseif ~v/g∈a A no letters?
[21] z←(⊞f),1,(⊞t),256 A '1:2' selects entire rows (c
[22] :else A normal case
[23] z←(AltoRC f),AltoRC t
[24] :end

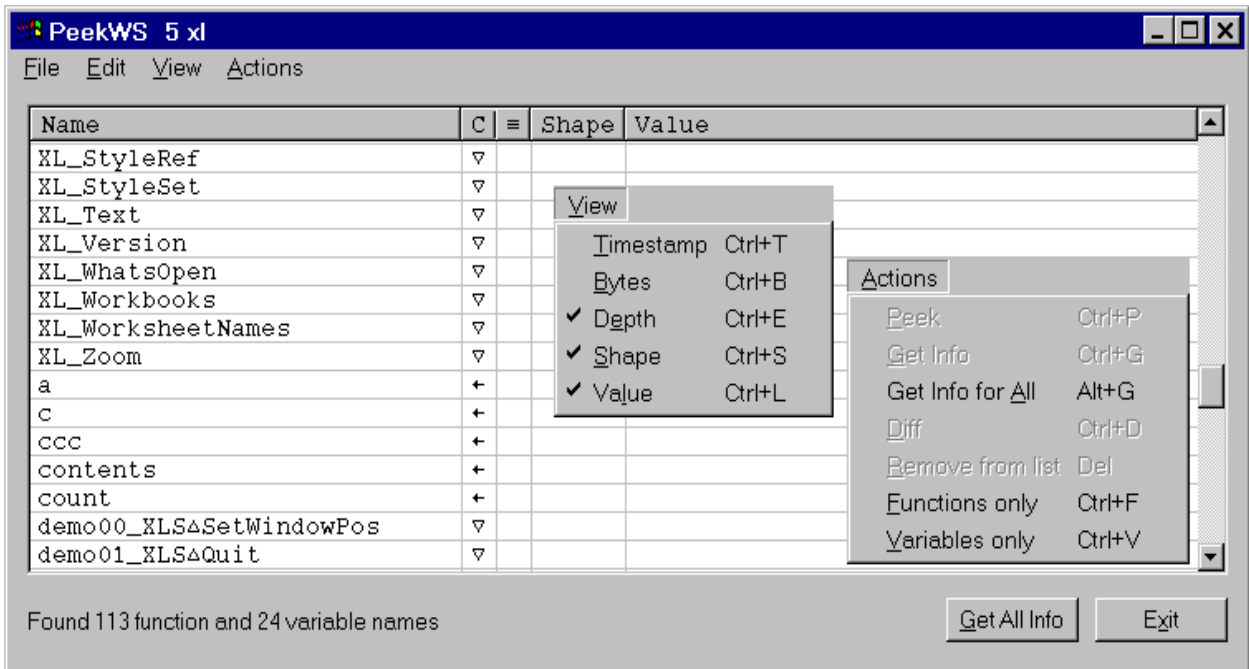
```

Ctrl+Q to quit -- function was localized and will disappear from active ws

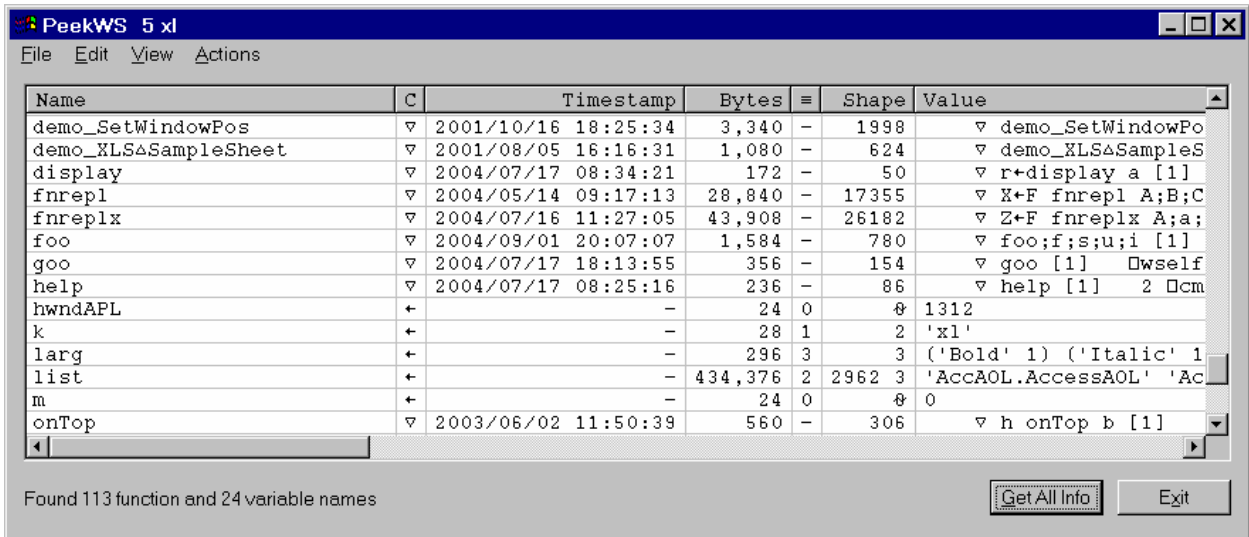
Ctrl+E to define in active ws

Peek At Workspace:

lpeekws 5 xl



Click Get All Info button; view Timestamps and Bytes



What's In My Clipboard?

Copy 8 characters 'cliptest' from APL session into clipboard

!clipboard

Format	Hex	Name	Bytes	Value
1	1	A CF_TEXT	12	'cliptest',␣TCNUL,␣TCNUL,␣TCNUL,␣TCNUL
16	10		4	␣TCHT,'␣',␣TCNUL,␣TCNUL
7	7	A CF_OEMTEXT	12	'cliptest',␣TCNUL,␣TCNUL,␣TCNUL,␣TCNUL
13	D	A CF_UNICODETEXT	24	'c',␣TCNUL,'l',␣TCNUL,'i',␣TCNUL,'p',␣TCNUL

Copy 8 characters 'cliptest' from Microsoft Word into clipboard

!clipboard

Format	Hex	Name	Bytes	Value
49161	C009	DataObject	4	'ó€',␣TCLF,␣TCNUL
49166	C00E	Object Descriptor	120	'x',␣TCNUL,␣TCNUL,␣TCNUL,␣TCNUL,'←',␣TCNUL
49303	C097	Rich Text Format	3,434	{\rtf1\ansi\ansicpg1252\uc1\de
49279	C07F	HTML Format	2,379	'Version:1.0',␣TCNL,␣TCLF,'Star
1	1	A CF_TEXT	9	'cliptest',␣TCNUL
13	D	A CF_UNICODETEXT	18	'c',␣TCNUL,'l',␣TCNUL,'i',␣TCNUL
14	E	A CF_ENHMETAFILE	0	(Could not get GlobalLock)
3	3	A CF_METAFILEPICT	16	␣TCBS,␣TCNUL,␣TCNUL,␣TCNUL,'`O'
49163	C00B	Embed Source	19,968	'-ÿæí␣→ß',16ρ␣TCNUL,'>',␣TCNUL
49156	C004	Native	19,968	'-ÿæí␣→ß',16ρ␣TCNUL,'>',␣TCNUL
49155	C003	OwnerLink	28	'Word.Document.8',␣TCNUL,'Docum
49165	C00D	Link Source	132	␣TCHT,'ε',6ρ␣TCNUL,'À',6ρ␣TCNUL
49167	C00F	Link Source Descriptor	140	'i',␣TCNUL,␣TCNUL,␣TCNUL,'←',␣TCNUL
49154	C002	ObjectLink	37	'Word.Document.8',␣TCNUL,'Docum
49171	C013	Ole Private Data	408	␣TCNUL,␣TCNUL,␣TCNUL,␣TCNUL,'␣§
16	10		4	␣TCHT,'␣',␣TCNUL,␣TCNUL
7	7	A CF_OEMTEXT	9	'cliptest',␣TCNUL

Press PrintScreen key

!clipboard

Format	Hex	Name	Bytes	Value
2	2	A CF_BITMAP	0	(Could not get GlobalLock)
8	8	A CF_DIB	9,437,236	'(',␣TCNUL,␣TCNUL,␣TCNUL,␣TCNUL,␣TCHT,␣TCNUL
17	11		9,437,320	'!',␣TCNUL,␣TCNUL,␣TCNUL,␣TCNUL,␣TCHT,␣TCNUL

Monitor

```
]mf /on
...
]mf
```

MF summary for wsid: 5 REXPLAY

File View Resolution (Milliseconds)

Function	Total	Here	Pct	r/It	Calls	Header
wavefile	1604.550	1603.667	34.03	42	42	r+m wavefile f;DELX
wvh	12082.189	1154.167	24.49	1733	1733	r+opt wvh event;a;b;c;d;
FFAW	1106.679	820.397	17.41	5284	5284	R+W FFAW A;D;E;F;K;N;P;S
TraceMe	1917.115	679.415	14.42	1742	1742	TraceMe;d;e;f;h;r;s;t;v;
wv	339.079	184.786	3.92	8	2	wv;b;d;g;h;i;lw;n;p;q;t;
wvfmtclock	124.662	83.412	1.77	1393	1393	r+wvfmtclock m;s;f;p
wvo	81.176	54.307	1.15	4	4	opt wvo event;f;fh;fw;fl
fexist	52.478	51.617	1.10	42	42	r+fexist f;t;DELX
Ini	48.269	45.802	0.97	126	126	r+v Ini a;d;e;f;s
fread	16.841	16.788	0.36	2	2	r+fread f;t;DELX
Parse	10.214	4.204	0.09	76	76	r+Parse a;q;i;w
workarea	4.125	2.962	0.06	7	7	a+workarea p;e;h;i;n;r;s
mrufiles	15.289	2.346	0.05	36	4	w mrufiles a;f;s;i;m;c;n
dlb	3.555	2.170	0.05	152	152	r+dlb v
TextWidth	1.552	1.362	0.03	16	16	w+o TextWidth t
AlwaysOnTop	1.397	1.312	0.03	2	2	owself AlwaysOnTop b;h;v
trypath	1.281	1.204	0.03	4	4	r+trypath p;DELX
TextHeight	0.677	0.617	0.01	4	4	h+o TextHeight t
KeepOnScreen	15.796	0.538	0.01	5	5	KeepOnScreen;a;c;g;p;u;w
Assert	0.753	0.512	0.01	21	21	w Assert a
upper	0.820	0.450	0.01	38	38	r+upper a
Sink	1.739	0.430	0.01	172	172	Sink a

MF detail for function: TraceMe

File View Resolution (Milliseconds)

Line	Total	Here	Iter	Instruction
TraceMe	1917.115	679.415	1742	▽ TraceMe;d;e;f;h;r;s;t;v;w;x
TraceMe[1]	0.000	0.000	0	[1] A Trace callback function
TraceMe[2]	0.000	0.000	0	[2] A 11 Feb 2003 Rex Swain, Independ
TraceMe[3]	0.000	0.000	0	[3]
TraceMe[4]	11.213	11.213	1742	[4] +ΔRuntimep0 A only
TraceMe[5]	8.894	8.894	1742	[5] w+OPW
TraceMe[6]	797.545	47.452	1742	[6] r+'[' ,owself, ' ' , (w FFAW owevent)
TraceMe[7]	31.421	31.421	1742	[7] s+OSI A si s
TraceMe[8]	9.892	9.892	1742	[8] +(1=+ps)p0 A done
TraceMe[9]	5.008	5.008	1742	[9] t+''
TraceMe[10]	6.810	6.810	1742	[10] :if 1<+ps A if a
TraceMe[11]	27.970	27.970	1742	[11] f+s[2;] ◇ f+(^f≠[']/f A call
TraceMe[12]	20.176	20.176	1742	[12] d++/s[;u1+pf]^.=f, '[' A recu
TraceMe[13]	4.723	4.723	1742	[13] e+f
TraceMe[14]	15.174	15.174	1742	[14] e+e, (d>1)/' [depth ' , (±d), ']'
TraceMe[15]	5.690	5.690	1742	[15] e+e, ':'
TraceMe[16]	14.506	14.506	1742	[16] (x v)+2p1 DAT f A expl
TraceMe[17]	5.968	5.968	1742	[17] :if v>0 A if a
TraceMe[18]	12.576	12.576	1737	[18] h+OCRL f, '[0]' A head
TraceMe[19]	15.661	15.661	1737	[19] h+(-1+h';')h A igno

Workspace Search/Replace with *Regular Expressions*

... fnreplx ...

Right argument:

```
... fnreplx 'target'           A Search (if target begins with a letter)
... fnreplx 'target/options'  A Ditto, with options
Any non-alpha delimiter character may be used in place of the slashes shown below
... fnreplx '/target'         A Search (exactly 1 delimiter)
... fnreplx '/target/'       A Search (exactly 2 delimiters)
... fnreplx '/target/options' A Ditto, with options
... fnreplx '/target/replacement/' A Replace (exactly 3 delimiters)
... fnreplx '/target/replacement/options' A Ditto, with options
```

Regex options:

```
... fnreplx '.../i' A Case-insensitive matching
... fnreplx '.../m' A Multi-line
... fnreplx '.../x' A Extended syntax: un-escaped white space is ignored
... fnreplx '.../s' A Dot meta-character matches all characters including new-lines
```

Left argument:

```
fnreplx ... A Look in all functions
'F1 F2' fnreplx ... A Only look in named function(s)
'F*' fnreplx ... A Function names with wildcards (foo* *old fm*def etc.)
'~F1' fnreplx ... A Look in all but named function(s)
'?' fnreplx ... A Prompt for confirmation before each replacement
'-' fnreplx ... A Underline the hits as they are displayed ('-' okay too)
'+-' fnreplx ... A Suppress display; return fn names
'A' fnreplx ... A Match only in comments
'"' fnreplx ... A Match only in character constants ('"' okay too)
'α' fnreplx ... A Match only in APL (not comments/quotes)
```

Function line numbers in brackets are not considered part of the function
(I display the line numbers so you can double-click on "foo[6]" to start editor)

Function lines may have leading blanks, but never have trailing blanks
Use /^FOO/m to find FOO at the (very) beginning of a line
Use /\ *FOO/m to find FOO at the beginning of a possibly-indented line
Use /FOO\$/m to find FOO at the end of a line

Special regex characters that must be escaped in target: \ | ^ \$. ? * + [] () { }

PCRE comes from a Perl/Unix heritage, where linefeeds are used between lines of text
So when they say "newline" (\n) they are talking about □TCLF
So when they say "return" (\r) they are talking about □TCNL
To make the regex engine work, I put LF's (not NL's) between function lines

Regex cheat sheet

Special characters, outside of square brackets:

```
\  general escape character with several uses
^  assert start of string (or line, in multiline mode)
$  assert end of string (or line, in multiline mode)
.  match any character except newline (by default)
[  start character class definition
|  start of alternative branch
(  start subpattern
)  end subpattern
?  extends the meaning of (
    also 0 or 1 quantifier
    also quantifier minimizer
*  0 or more quantifier
+  1 or more quantifier
    also "possessive quantifier"
{  start min/max quantifier
```

Special characters, inside square brackets:

```
\  general escape character
^  negate the class, but only if the first character
-  indicates character range
]  terminates the character class
```

Non-printing characters:

```
\a  alarm (dec 7, hex 07) \TCBEL
\cx  "control-x", where x is any character
\e  escape (dec 27, hex 1B) \TCESC
\f  formfeed (dec 12, hex 0C) \TCFF
\n  newline (dec 10, hex 0A) \TCLF  a note Unix nomenclature!
\r  return (dec 13, hex 0D) \TCNL
\t  tab (dec 9, hex 09) \TCHT
\ddd character with octal code ddd (or backreference)
\xhh character with hex code hh
```

Generic character classes (upper-case for "not")

```
\d  digit [0123456789]
\s  whitespace character (blank, tab, LF, CR, formfeed)
    (caution: this also matches LFs between lines! use \x20 for blank)
\w  word character [1-9a-zA-Z_]
```

Assertions:

```
^  matches beginning of line (with /m option)
$  matches end of line      (with /m option)
\b  matches at word boundary (\B for not)
```

Quantifiers (greedy by default; add ? suffix to make non-greedy):

```
?  0 or 1 (same as {0,1})
*  0 or more (same as {0,} )
+  1 or more (same as {1,} )
{n} exactly n
{min,} at least min
{min,max} between min and max inclusive
```

Back references:

```
\2 refers to second parenthesized subexpression
```

Examples

Find fns that contain aa followed by bb on the same line

```
'←' fnreplx '/aa.*bb/'
```

Use the /s regex option to have "." match even newline characters; e.g.,

Find fns that contain aa followed by bb, not necessarily on the same line

```
'←' fnreplx '/aa.*bb/s'
```

Find fns that contain aa and bb, in either order, anywhere in the function

```
'←' fnreplx '/aa.*bb|bb.*aa/s'
```

Use the /i regex option for case-insensitive matching; e.g.,

```
fnreplx '/πio/i'
```

For "syntactic" search, use the regex "word boundary" assertion \b

```
'-foo' fnreplx '/\bfoo\b/'
```

This works pretty well for APL, but... Note that the regex engine does not consider Δ and Δ to be letters. So that would match the "foo" in "Δfoo".

Match all APL identifier names:

```
'-foo' fnreplx '/[⊖A-Za-zΔΔ][A-Za-zΔΔ0-9_]*/'
```

Match integers: basically it's

```
fnreplx '/-?\d+/'
```

but we need to pay attention to what comes before and after; this is close

```
fnreplx '/(?![\.\w])-\d+(?![\.\w])/'
```

But since \w does not include the deltas, we should really use

```
fnreplx '/(?![\.\a-zA-Z0-9_ΔΔ])-\d+(?![\.\a-zA-Z0-9_ΔΔ])/'
```

The (?. . .) things are lookbehind and lookahead assertions

Delete the leading blank from lines that are indented by exactly one space

```
fnreplx '/^ (?! )//m'
```

Parenthesized subpatterns in the target string may be referred to with

\$1 \$2 \$3 in the replacement string; e.g.,

```
fnreplx ';\b(able|baker)\b;<I>$1</I>;'
```

wraps <I> and </I> html tags around words able and baker

Change a ⊖ss b to b⊖a where a and b are variable names

```
fnreplx '/([⊖A-Za-zΔΔ][A-Za-zΔΔ0-9_]*) *⊖ss +([⊖A-Za-zΔΔ][A-Za-zΔΔ0-9_]*)/$2⊖$1/i'
```

Break lines at diamonds (remember, newline to the Unix crowd is ⊖TCLF)

```
'foo' fnreplx '/\s◇\s/',⊖tclf,'/'
```

Find bare branches

```
fnreplx '/->[ ]*(R|◇|$)/m'
```

\1 \2 \3 . . . \9 may be used for "back references" --

to refer to strings matched in parenthesized subexpressions earlier in the target

Find repeated words

```
'-' fnreplx '/\b([A-Za-z]+) +(\1)\b/'
```

Tool Talk 3

Rex Swain
Independent Consultant
8 South Street
Washington, CT 06793
Tel 860-868-0131
Fax 860-868-9970
rex@rexswain.com

<http://www.rexswain.com>

APL2000 User Conference

November 2005

Notes

To add UCMSDREX.SF to your user command files path:

```
]UFILE [1.5] UCMSDREX
```

If you already have UCMSDREX.SF from a previous conference, just replace it with this one; the tools from 2003 and 2004 are all here too.

See UCMSDREX03.PDF and UCMSDREX04.PDF to learn about tools from the 2003 and 2004 conferences.

These tools *no longer* require)EVLEVEL 2

Help:

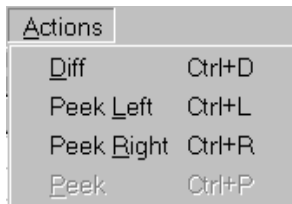
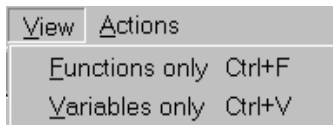
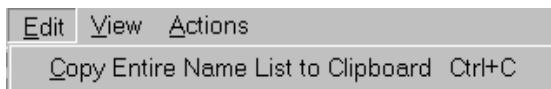
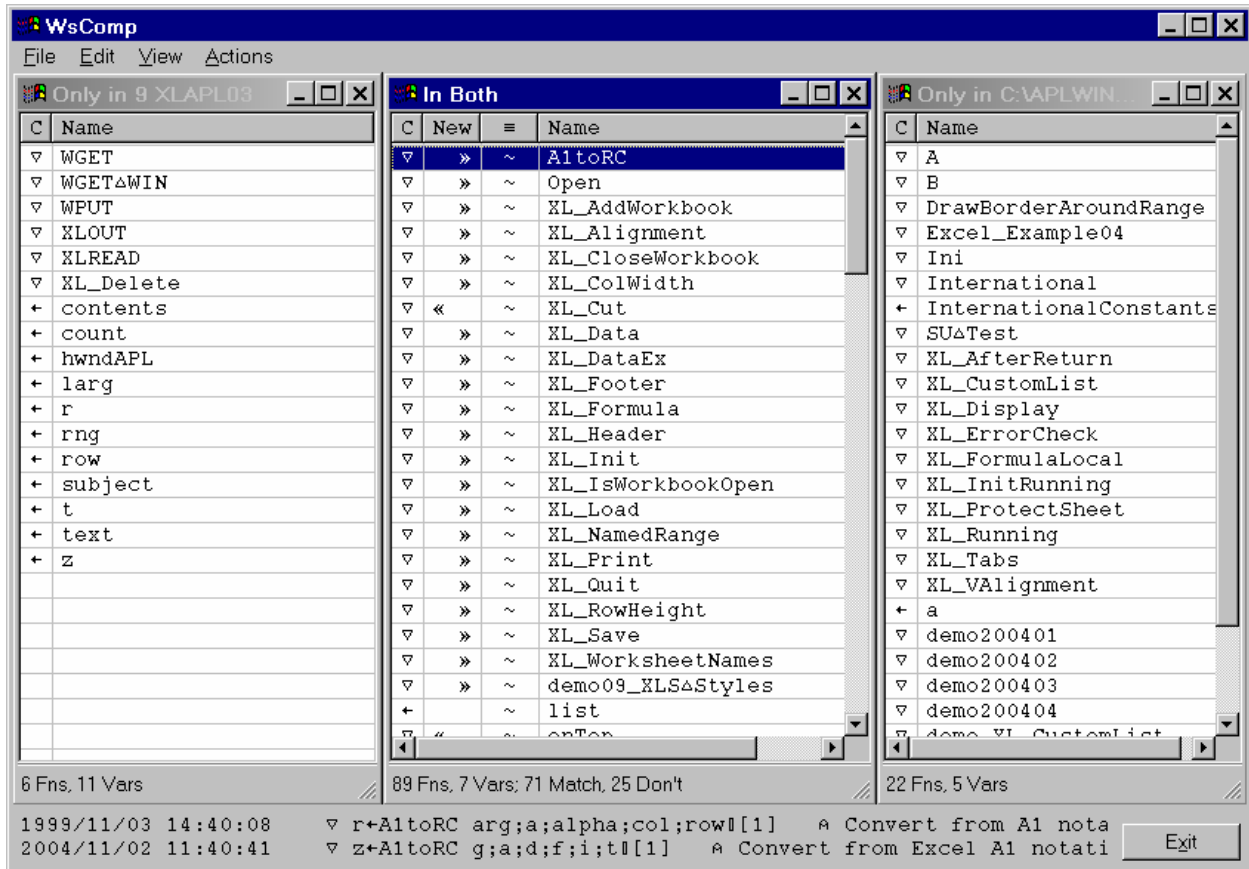
```
]foo ?
```

Font: You may edit in your APLW.INI file; the defaults are:

```
[UCMSDREX]  
FontName=APLFONT  
FontSize=11  
FontStyle=0  
FontCharset=default
```

Workspace Compare

lwscomp 9 xlapl03 ; C:\APLWin50\2004 Conf Disk\xlapl04



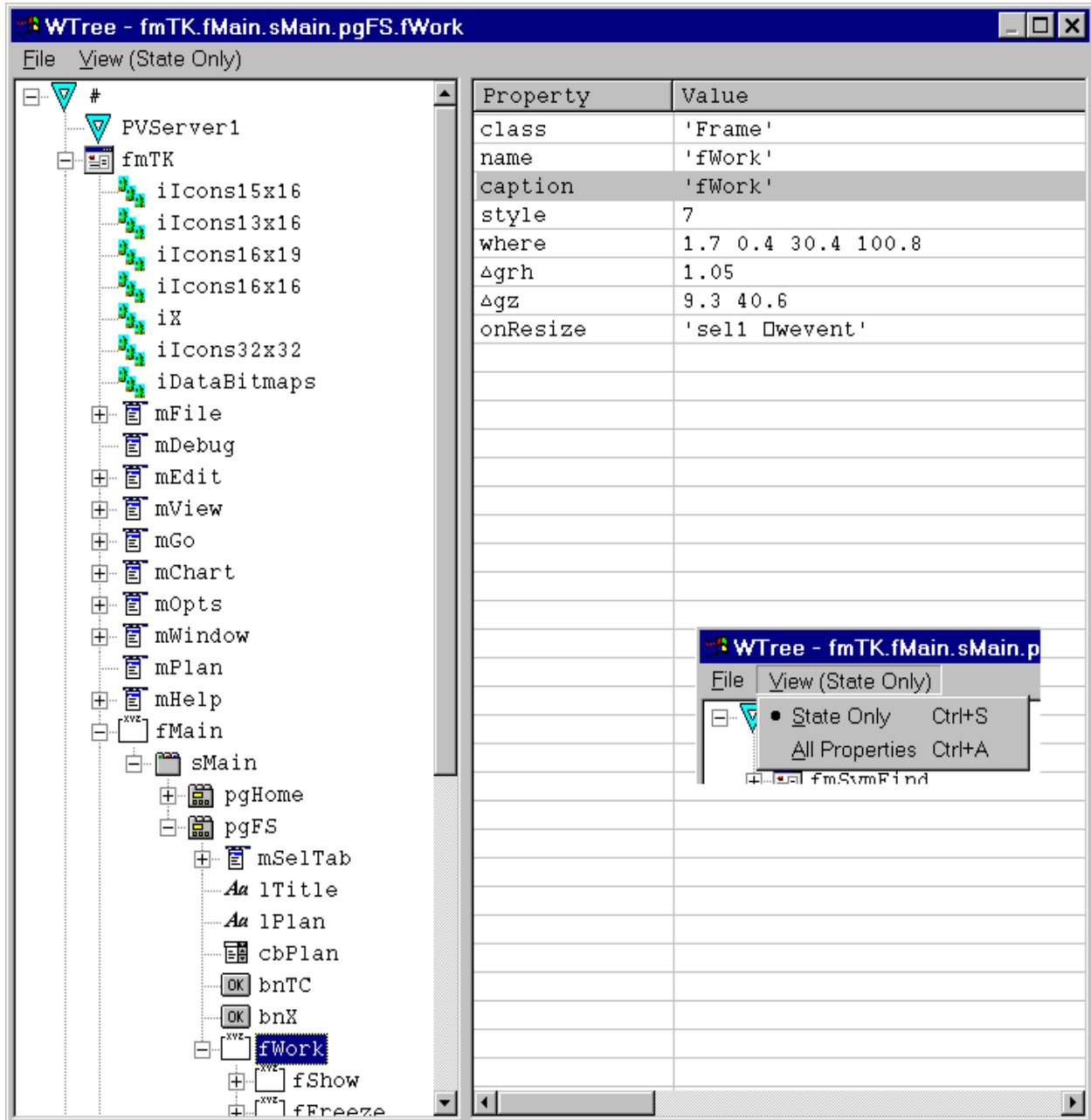
Try `lwscomp /regserver` if you have not already registered aplw as a server.

Shadowing can be avoided by using a saved ws rather than the active ws.

JWTREE with argument

- A Enhanced to allow argument specifying object to start with
- A]WTREE fmFoo.fData.bnCancel A start with specified object
- A]WTREE . A start with []wself
- A Added View menu (All Properties/State Only)
- A Remembers window size and position

`]wtree fmTK.fMain.sMain.pgFS.fWork`



APL+Win Keys

]keys

Keys					
File Edit					
A	C	S	Shifts	Key	Default
				F1	Help
				F2	Jump to next bookmark
	C		Ctrl+	F2	Bookmark toggle
		S	Shift+	F2	Jump to previous bookmark
	C	S	Ctrl+Shift+	F2	Remove all bookmarks
				F3	Repeat find
		S	Shift+	F3	Repeat find backward
	C		Ctrl+	F3	Find current selection
	C	S	Ctrl+Shift+	F3	Find current selection backward
	A		Alt+	F3	Find dialog (same as Ctrl+F)
				F4	
	A		Alt+	F4	End APL session
				F5	Run [CW]
	C		Ctrl+	F5	Set next statement [CW]
				F6	
	C		Ctrl+	F6	Next window
	C	S	Ctrl+Shift+	F6	Previous window
				F7	
	C		Ctrl+	F7	Run to cursor or below [CW]
	C	S	Ctrl+Shift+	F7	Run to cursor [CW]
				F8	
				F9	Recall previous line backward
		S	Shift+	F9	Recall previous line forward
	C		Ctrl+	F9	Recall line dialog
	A		Alt+	F9	Jump to previous executed line
	A	S	Alt+Shift+	F9	Jump to next executed line
				F10	Step to next statement [CW]
		S	Shift+	F10	Step to next line [CW]
	C		Ctrl+	F10	Step over by primitive [CW]
				F11	Step into function [CW]
		S	Shift+	F11	Step out of function [CW]
	C		Ctrl+	F11	Step into by primitive [CW]
				F12	Copy code to APL session [CW]
	C		Ctrl+	F12	Edit pending function
	C		Ctrl+	A	Select all
	C	S	Ctrl+Shift+	A)SAVE As dialog
	C		Ctrl+	C	Copy
	C	S	Ctrl+Shift+	C)COPY dialog
	C	S	Ctrl+Shift+	D)DROP dialog
	C		Ctrl+	E	End and save edit
	C		Ctrl+	F	Find dialog
	C		Ctrl+	G	Get (Fetch dialog)
	C	S	Ctrl+Shift+	G	Gather selected executed lines to clipboard
	C		Ctrl+	J	Jump dialog
	C	S	Ctrl+Shift+	J	Jump to column dialog
	C		Ctrl+	K	Show/hide Code Walker
	A	C	Alt+Ctrl+	K	Show/hide code window [CW]
	C	S	Ctrl+Shift+	K	Show/hide state indicator window [CW]
	C		Ctrl+	L	Localize toggle
	C	S	Ctrl+Shift+	L)LOAD dialog

Keys					
File Edit					
A	C	S	Shifts	Key	Default
	C		Ctrl+	M	Match paren/quote/bracket
	C	S	Ctrl+Shift+	M	Match paren/quote/bracket and select
	C		Ctrl+	N	New edit
	C		Ctrl+	O	Open edit dialog
	C	S	Ctrl+Shift+	O	Open edit session on object at caret
	C		Ctrl+	P	Print dialog
	C	S	Ctrl+Shift+	P)PCOPY dialog
	C		Ctrl+	Q	Quit edit
	C		Ctrl+	R	Replace dialog
	C	S	Ctrl+Shift+	R)CLEAR dialog
	C		Ctrl+	S	Save edit
	C		Ctrl+	T	Toggle edit and session
	C		Ctrl+	V	Paste
	C		Ctrl+	W	Watchpoints dialog
	C		Ctrl+	X	Cut
	C		Ctrl+	Z	Undo
	C		Ctrl+	↓	Scroll down without moving caret
	C	S	Ctrl+Shift+	↓	Move the cursor line to the bottom of the page
	C		Ctrl+	↑	Scroll up without moving caret
	C	S	Ctrl+Shift+	↑	Move the cursor line to the top of the page
	C	S	Ctrl+Shift+	End	Select from caret to end
A		S	Alt+Shift+	End	Erase from caret to end of session log
	C	S	Ctrl+Shift+	Home	Select from caret to beginning
	C		Ctrl+	PgDn	Scroll left
	C		Ctrl+	PgUp	Scroll right
				Tab	Insert tab indent
		S	Shift+	Tab	Remove tab indent
	C		Ctrl+	Tab	Next window
	C	S	Ctrl+Shift+	Tab	Previous window
A			Alt+	Tab	Next application [XX]
	C		Ctrl+	Space	Insert space indent
	C	S	Ctrl+Shift+	Space	Remove space indent
A			Alt+	Bksp	Undo (same as Ctrl+Z)
		S	Shift+	Del	Cut (same as Ctrl+X)
		S	Shift+	Ins	Paste (same as Ctrl+V)
	C		Ctrl+	Ins	Copy (same as Ctrl+C)
	C		Ctrl+	Enter	Insert line in session (without execution)
	C	S	Ctrl+Shift+	Enter	Move the cursor line to the middle of the page
	C		Ctrl+	/	Insert lamp
	C	S	Ctrl+Shift+	/	Remove lamp
	C		Ctrl+	+	Increase font size
	C	S	Ctrl+Shift+	+	Decrease font size
	C		Ctrl+	.	Stop toggle
	C		Ctrl+	,	Trace toggle
	C	S	Ctrl+Shift+	.	Stop remove
	C	S	Ctrl+Shift+	,	Trace remove
	C		Ctrl+	Num *	Top of stack [CW]
	C		Ctrl+	Num -	Up one stack level [CW]
	C		Ctrl+	Num +	Down one stack level [CW]
	C		Ctrl+	Num /	Bottom of stack [CW]

Trace Variable Changes

```

▽ foo;a;i
[1]   :for i :in 13
[2]     a←i
[3]   :end
▽

```

```

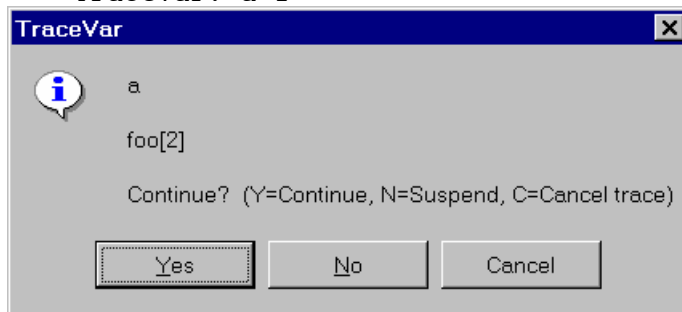
]tracevar a
foo
→→→ TraceVar: foo[2]
←←← TraceVar: a+1
→→→ TraceVar: foo[2]
←←← TraceVar: a+2
→→→ TraceVar: foo[2]
←←← TraceVar: a+3

```

```

]tracevar a /alert
foo
→→→ TraceVar: foo[2]
←←← TraceVar: a+1

```



```

]tracevar /off

```

```

]tracevar ?
]TraceVar foo ...           A trace changes to the named variable(s)
]TraceVar foo:2⇒foo         A display 2⇒foo when foo changes; just one var
]TraceVar ... /STOP         A sets stop on first change; suspends before next
]TraceVar ... /ALERT        A display MessageBox after change report

]TraceVar foo ... /OFF      A turn off tracing of the named variable(s)
]TraceVar /OFF              A turn off all tracing

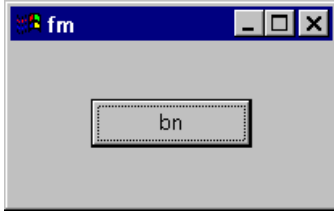
```

Trace Object Events and Actions

```

[]wself-'fm' []wi 'Create' 'Form' ('size' 5 20)
[]wself-'bn' []wi 'Create' 'Button'

```



]traceobj

TraceObj now tracing 31 events on fm.bn

TraceObj now tracing actions on fm.bn

```

  A cause an Action
  []wi 'where' 1 1
... TraceObj: fm.bn Action ('where' 1 1) <[fm.bn;where]
  A Alt+Tab to form
... TraceObj: fm.bn Focus (⊕)
... TraceObj: fm.bn Paint (⊕)
  A use mouse to click button and then click X to close form
  A trace display is: ... TraceObj: []wself []wevent ([]warg) []si
... TraceObj: fm.bn MouseEnter (1.1 9 0 0 0)
... TraceObj: fm.bn MouseMove (1.1 9 0 0 0)
... TraceObj: fm.bn MouseMove (0.8 8.8 0 0 0)
... TraceObj: fm.bnMouseDown (0.4 8.4 1 1 0)
... TraceObj: fm.bn MouseUp (0.4 8.4 1 0 0)
... TraceObj: fm.bn Click (⊕)
... TraceObj: fm.bn MouseMove (0.4 8.4 0 0 0)
... TraceObj: fm.bn MouseMove (0.4 8.2 0 0 0)
... TraceObj: fm.bn MouseMove (0.4 9 0 0 0)
... TraceObj: fm.bn MouseLeave (⊕)
... TraceObj: fm.bn Unfocus (⊕)
... TraceObj: fm.bn Destroy (⊕)
  A delete form (which causes button to be deleted)
  'fm' []wi 'Delete'
... TraceObj: fm.bn Delete (⊕) <[fm;Delete]

```

]traceobj ?

Trace activity on an APL GUI object

Syntax:]TraceObj [obj] [/A] [/E] [/OFF] [/X=...] [/I=...]

If object not specified, []wself is used

Use /A to trace just actions or /E to trace just events; default is both
 /X excludes (all but) actions/events matching substring -- e.g., /X=mouse
 /I includes (only) actions and events matching substring

Any trace on an object replaces any previous trace

Trace display is: ... TraceObj: []wself []wevent ([]warg) []si

For example, after: 'fm' []wi 'Create' 'Form' ('onResize' 'foo')

/A traces property changes: 'fm' []wi 'size' 10 20
 and methods: 'fm' []wi 'Show'

but it does NOT trace events (Resize is not traced if user resizes fm)

/E traces events such as Resize triggered if user resizes form

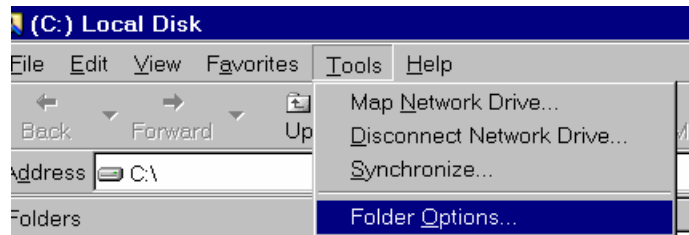
Associating JFileDoc with .SF Files

The following is for Windows 2000, and assumes your APL+Win executables are in directory C:\APLWIN52 ; adjust accordingly.

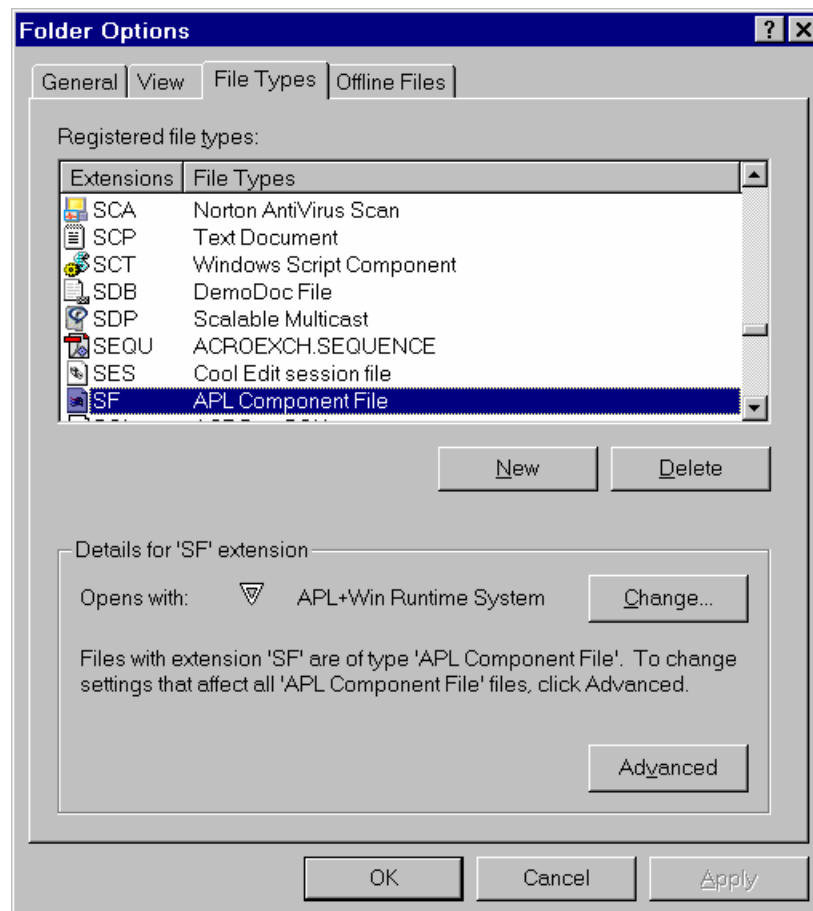
)LOAD the workspace FILEDOC and then)RSAVE it into C:\APLWIN52

Note!)RSAVE not)SAVE

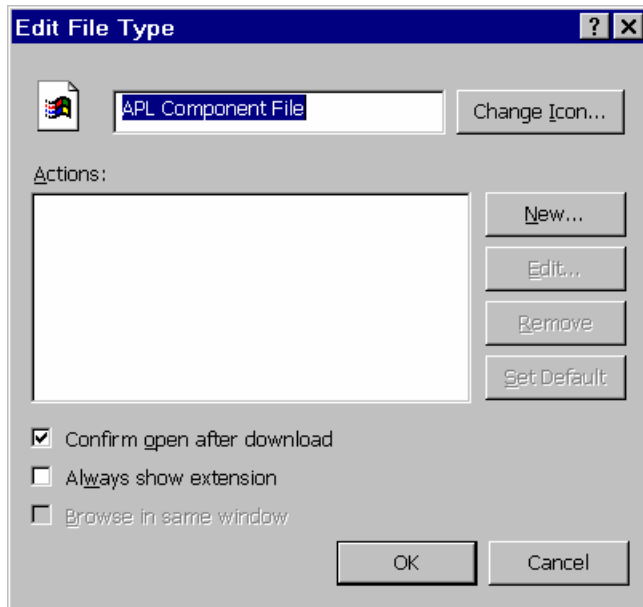
Start Windows Explorer, Tools menu, Folder Options



Click the File Types tab, and scroll down to SF files



Click the Advanced button

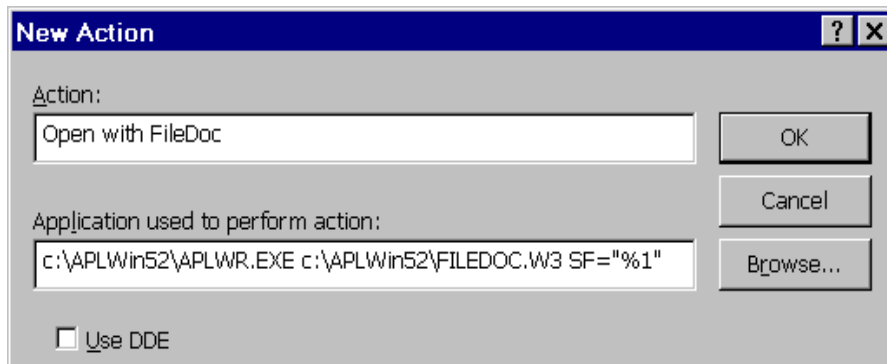


Click the **New** button

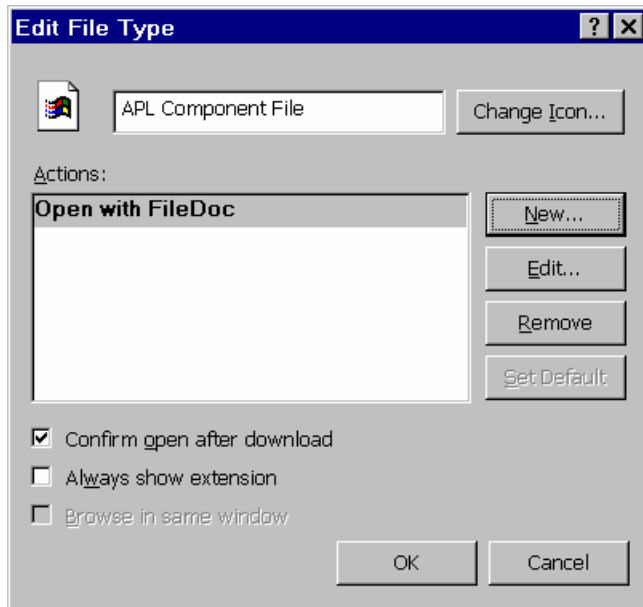
Type `Open with FileDoc`

and `C:\APLWIN52\APLWR.EXE C:\APLWIN52\FILEDOC.W3 SF="%1"`

Note! APLWR not APLW

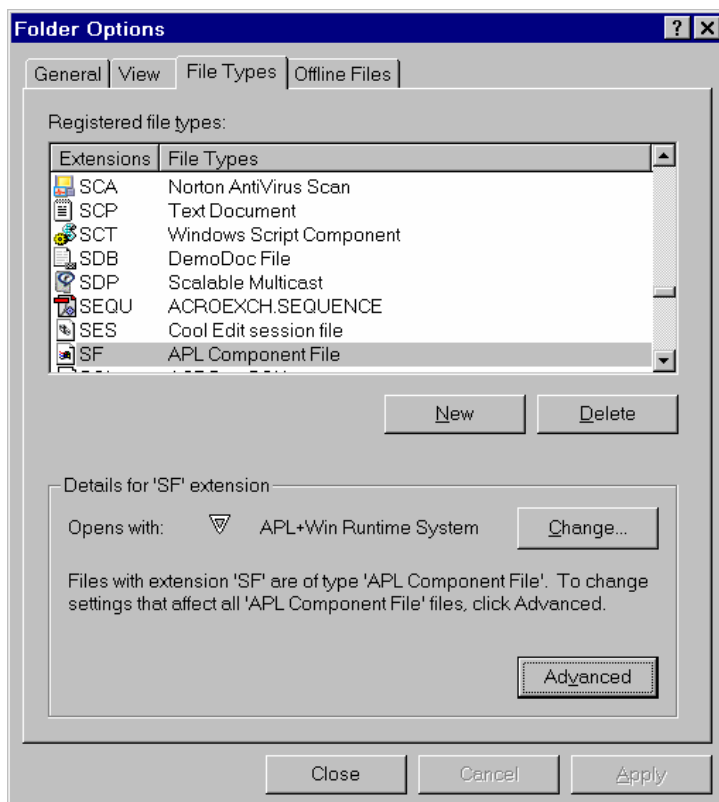


Click **OK**



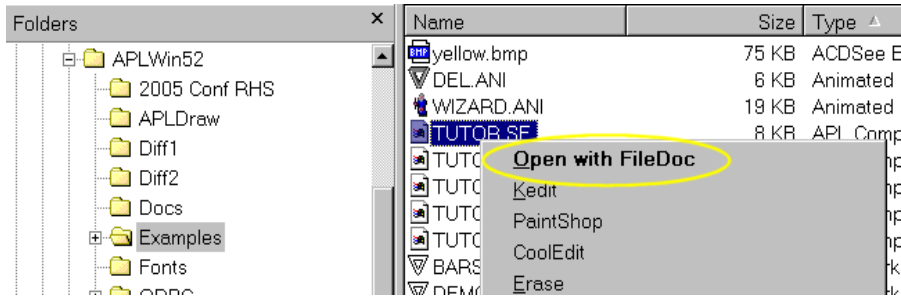
Note that **Open with FileDoc** is in bold face. That means it's the default action for the .SF file type.

Click OK.

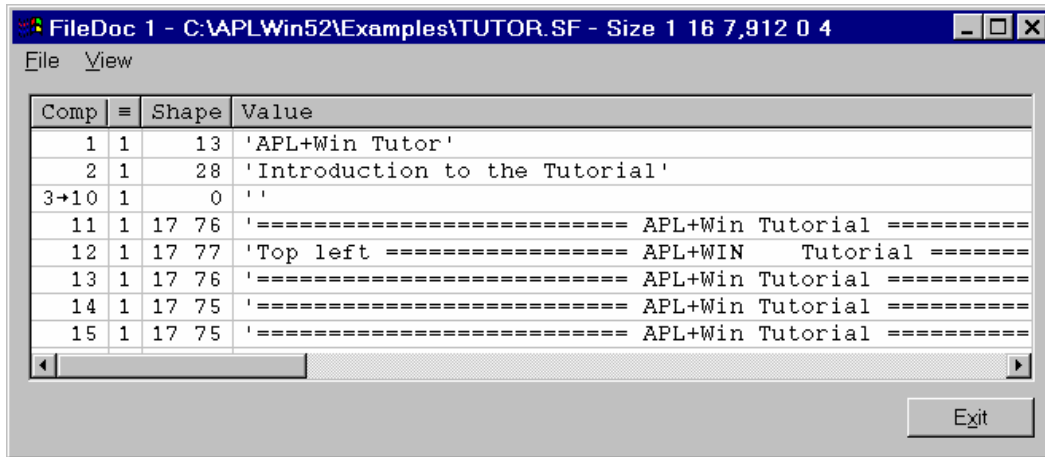


Click Close.

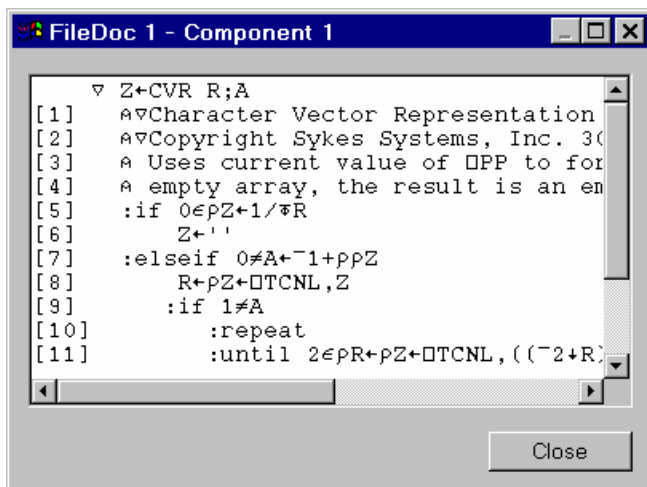
Now if you right-click on an .SF file, you should see **Open with FileDoc** on the context menu.



If you choose that menu item, or just double-click on the file, FileDoc should come up. And if someone emails an SF file to you, you should be able to take a quick look at it just by double-clicking the attachment.



Note that since this runs the APL+Win runtime engine,)EDIT is not possible. So when you peek at a file component, an Edit box is used.



Tool Talk 4

Rex Swain
Independent Consultant
8 South Street
Washington, CT 06793
Tel 860-868-0131
Fax 860-868-9970
rex@rexswain.com

<http://www.rexswain.com>

APL2000 User Conference

November 2006

Notes

- To add UCMSDREX.SF to your user command files path:

```
]UFILE [1.5] UCMSDREX
```

If you already have UCMSDREX.SF from a previous conference, just replace it with this one; the tools from 2003, 2004, and 2005 are all here too.

- To learn about tools from previous conferences, see:

```
UCMSDREX03.PDF (2003)
```

```
UCMSDREX04.PDF (2004)
```

```
UCMSDREX05.PDF (2005)
```

- These tools work with both)EVLEVEL 1 and)EVLEVEL 2
- Help:

```
]foo ?
```

Enhancements to Previous Tools

-]DIFF -- Automatically prompts to save changes made in Araxis Merge
-]PEM Edit?style -- Displays help for property

Acknowledgements

Many thanks to all those who have helped me to hurdle countless Windows obstacles -- notably Bob Smith, Patrick Parks, J. Merrill, Eric Lescasse, Pierre Gilbert, and Brent Hildebrand. I have enormous respect for these gurus who have explored the world outside of the APL, returned to tell the tale, and saved me countless hours of flailing around.

Function Search and Replace

lfnrepl ?

Find (and optionally replace) strings in functions

Syntax:

```
lfnrepl target           A Search (if target begins with a letter)
lfnrepl target/options  A Ditto, with options
```

Any non-alpha delimiter character may be used in place of the slashes shown below

```
lfnrepl /target         A Search (exactly 1 delimiter)
lfnrepl /target/       A Search (exactly 2 delimiters)
lfnrepl /target/options A Ditto, with options
lfnrepl /target/replacement/ A Replace (exactly 3 delimiters)
lfnrepl /target/rep/options A Ditto, with options
```

Multiple targets (not allowed in replace mode):

```
lfnrepl /t1/^/t2/options A Find only lines with both matches
lfnrepl /t1/v/t2/       A Find lines with either match
lfnrepl /t1/v\t2\      A Delimiters may be different
lfnrepl /t1/&~/t2/     A Logical not; & and | are also permitted
lfnrepl /t1/>/t2/     A Can use any of <=>≠v^v^* & | and ~
lfnrepl /t1/^~/t2/v/t3/ A Evaluated as APL would ("right to left")
```

Default is to perform case-sensitive, syntactic (name) search, on all global functions

Options:

To loosen match criteria:

```
/Γ      A Non-syntactic search (Alt+S for non-Syntactic)
/∅      A Ignore case (upper/lower/underscore) (Alt+C for Case)
```

To limit functions searched:

```
/f1 f2  A Only look in named functions
/f*     A Fn names with wildcards (foo* *old fm*def etc.)
/~f1 f2 A Look in all but named functions
```

To limit search scope (using all three is equivalent to using none):

```
/A      A Find only in comments
/'      A Find only in character constants (or /")
/α      A Find only in APL (not comments/quotes)
```

To expand search scope:

```
/∇      A Find strings anywhere in a function -- not necessarily on
        A the same line. Only useful with multiple targets. Doesn't
        A change behavior of /t1/v/t2/. Display from /t1/^/t2/∇ is
        A similar to /t1/v/t2/ but only shows functions that contain
        A at least one hit on both strings.
```

Other:

```
/p      A Auto reply Yes to the replacement confirmation prompt
/?      A Prompt before each replacement
/←      A Suppress display; return fn names
```

Nice features:

Both single- and double-quotes are handled (thanks, Zark!)
Searches for system names (like `□io`) are automatically case-insensitive

Outstanding issues:

```
Search for '-2' should not match '1E-2'
Search for 'if' should not match ':if'
Search for ':if' should be automatically case-insensitive
```

I recently implemented the multiple target feature, and I like it. My `fnreplx` which uses regular expressions (see UCMSDREX04.PDF from the 2004 conference) is much more powerful, but I have trouble remembering the regex syntax. This `/t1/^/t2/` syntax solves about 95% of my problems -- and I can remember it!

I also like the `/t1/^/t2/∇` feature. I often look for a string and get a flood of hits. Generally I only care about the hits in functions where a second string is present -- though not necessarily on the same line.

There is also `]scan` which is the same as `]fnrepl` except:

- Search only (no replace)
- Case-insensitive
- Non-syntactic

Align and WordWrap Comments

Would you like to go from this...

```

[0]  R←FACT N
[1]  :if N≤1  A 0 or 1
[2]    R←1  A by definition
[3]  :else  A all other args
[4]    R←N×FACT N-1  A recurse
[5]  :end

```

... to this:

```

[0]  R←FACT N
[1]  :if N≤1          A 0 or 1
[2]    R←1          A by definition
[3]  :else          A all other args
[4]    R←N×FACT N-1  A recurse
[5]  :end

```

Or, from this...

```

[154]
[155] A 403 Forbidden: The server understood the request,
[156] A but is refusing to fulfill it.
[157] A Authorization will not help and the request should not be repeated.
[158] A If the request method
[159] A was not HEAD and the server wishes to make public why the request
[160] A has not been fulfilled,
[161] A it should describe the reason for the refusal in the entity body.
[162] A This status code is
[163] A commonly used when the server does not wish to reveal exactly why
[164] A the request has been
[165] A refused, or when no other response is applicable.
[166]

```

... to this:

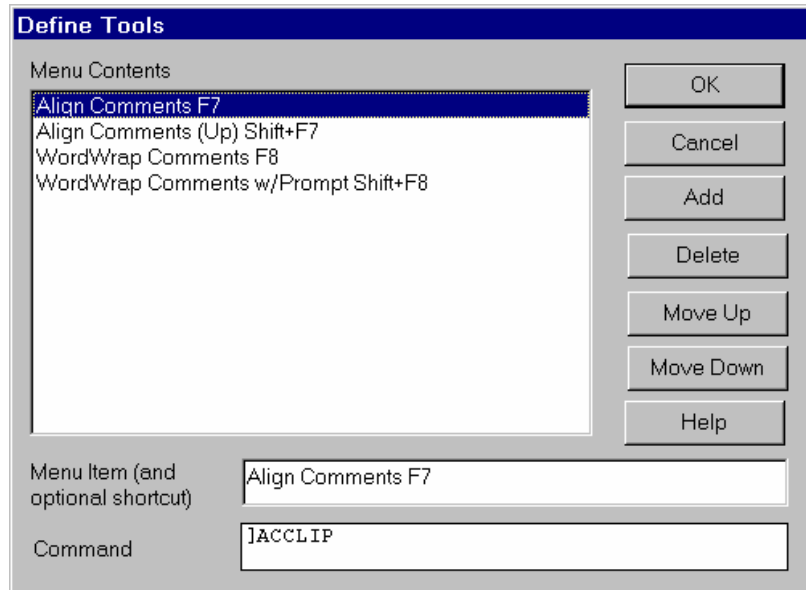
```

[154]
[155] A 403 Forbidden: The server understood the request, but is refusing to fulfill
[156] A it. Authorization will not help and the request should not be repeated. If
[157] A the request method was not HEAD and the server wishes to make public why the
[158] A request has not been fulfilled, it should describe the reason for the refusal
[159] A in the entity body. This status code is commonly used when the server does
[160] A not wish to reveal exactly why the request has been refused, or when no other
[161] A response is applicable.
[162]

```

Define Tools

- Select Options|Tools from the APL+Win session menu...



- ...and define four tools:

Menu Item	Command
Align Comments F7]ACCLIP
Align Comments (Up) Shift+F7]ACCLIP /UP
WordWrap Comments F8]WWCLIP
WordWrap Comments w/Prompt Shift+F8]WWCLIP /W=?

- Select Options|Save Settings Now from the APL+Win session menu.

Digression

Another way to assign these commands to function keys is:

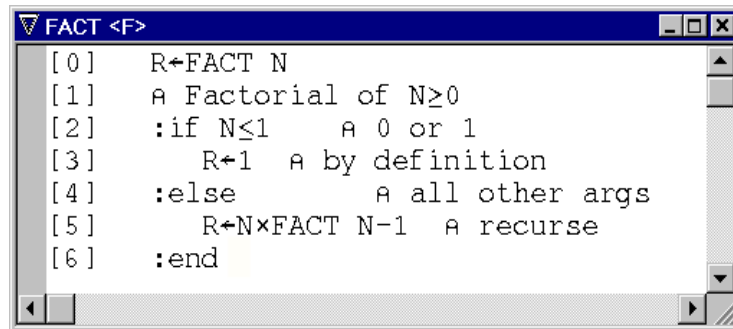
```
'#' □WI 'PF' 118 ']ACCLIP' A F7
'#' □WI 'PF' 100118 ']ACCLIP /UP' A Shift+F7
'#' □WI 'PF' 119 ']WWCLIP' A F8
'#' □WI 'PF' 100119 ']WWCLIP /W=?' A Shift+F8
```

See APL+Win Windows Interface help for "Keyboard Events" for Virtual Key Codes. Function keys F1-F24 are 112-135; add Shift=100000, Ctrl=200000, Alt=400000.

But these are not saved in your APLW.INI file.

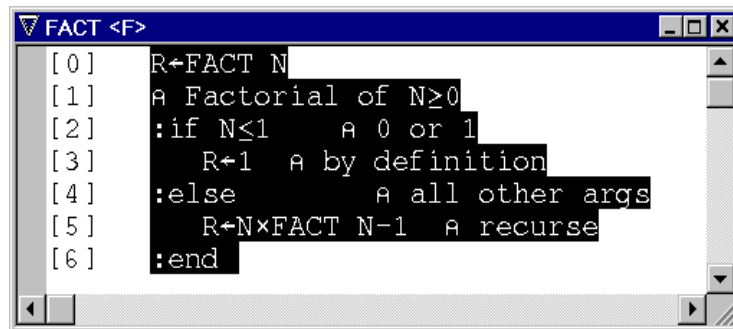
Align Comments

-)edit FACT



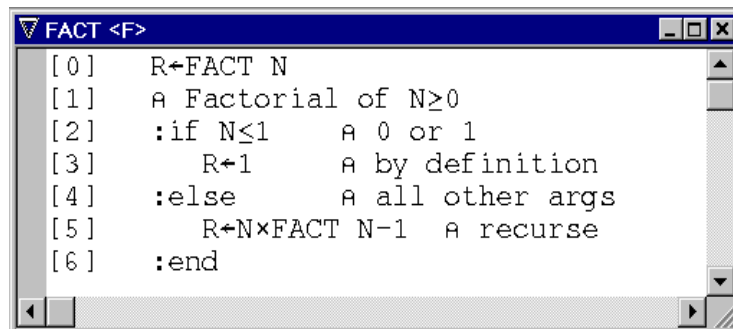
```
[0] R←FACT N
[1] A Factorial of N≥0
[2] :if N≤1 A 0 or 1
[3] R←1 A by definition
[4] :else A all other args
[5] R←N×FACT N-1 A recurse
[6] :end
```

- Ctrl+A



```
[0] R←FACT N
[1] A Factorial of N≥0
[2] :if N≤1 A 0 or 1
[3] R←1 A by definition
[4] :else A all other args
[5] R←N×FACT N-1 A recurse
[6] :end
```

- F7



```
[0] R←FACT N
[1] A Factorial of N≥0
[2] :if N≤1 A 0 or 1
[3] R←1 A by definition
[4] :else A all other args
[5] R←N×FACT N-1 A recurse
[6] :end
```

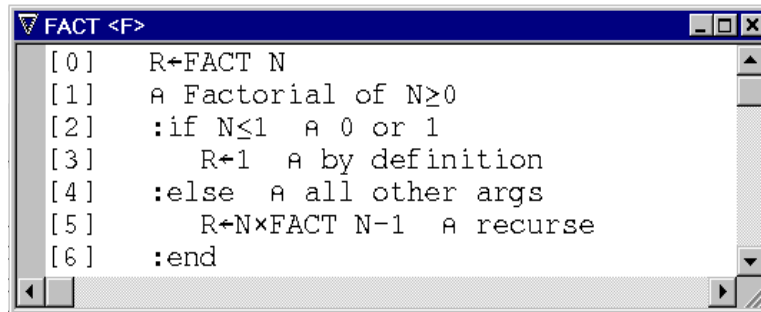
Whole-line comments are ignored.

Working from the top down, the first partial-line comment establishes the desired lamp position.

If you don't like what happened, just use Edit|Undo (Ctrl+Z).

Align Comments (Up)

-)edit FACT

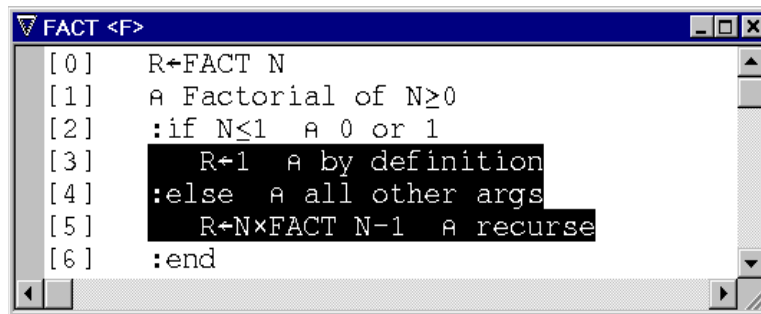


```

[0] R←FACT N
[1] A Factorial of N≥0
[2] :if N≤1 A 0 or 1
[3]   R←1 A by definition
[4] :else A all other args
[5]   R←N×FACT N-1 A recurse
[6] :end

```

- Select just three lines

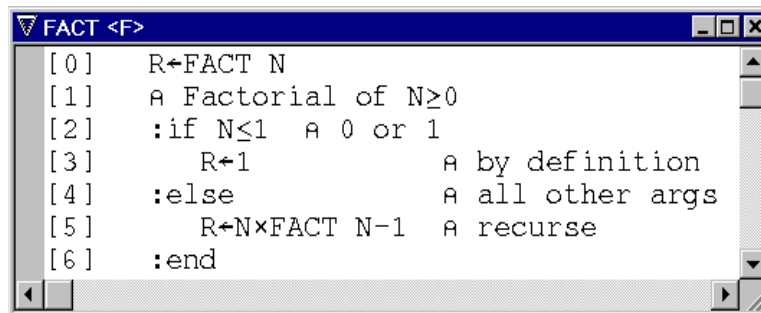


```

[0] R←FACT N
[1] A Factorial of N≥0
[2] :if N≤1 A 0 or 1
[3]   R←1 A by definition
[4] :else A all other args
[5]   R←N×FACT N-1 A recurse
[6] :end

```

- Shift+F7



```

[0] R←FACT N
[1] A Factorial of N≥0
[2] :if N≤1 A 0 or 1
[3]   R←1 A by definition
[4] :else A all other args
[5]   R←N×FACT N-1 A recurse
[6] :end

```

Only the selected lines are considered.

Working from the bottom up, the first partial-line comment establishes the desired lamp position.

If you don't like what happened, just use Edit|Undo (Ctrl+Z).

How Does This Work?

The way that APL+Win "Tools" work is perfect for this. When you press F7, the user command `]ACCLIP` is executed. Even though your focus is on the function editor window, the user command is executed in the session window -- not typed into the function as a keyboard macro program (such as Macro Express) would do. It's also nice that what's executed does not *appear* in the session (you may have seen `⌘INBUF` used when you *want* something to appear in the session).

Once we get our hands on the selected text, the rest is easy. The trick is getting (and replacing) the selected text.

When the user command executes, it copies the selected text into the Windows clipboard. It does this by using the Windows `keybd_event` function to simulate typing Ctrl+C. This has to be done at a low level:

1. press Ctrl
2. press C
3. release C
4. release Ctrl

`]ACCLIP` then retrieves the clipboard text using a series of Windows functions such as `GetClipboardData`, processes the text, and writes it back into the clipboard with `SetClipboardData`. (See utility functions `ClipGet` and `ClipCopy`.)

Finally we paste the modified text from the clipboard back into the function with more simulated keystrokes that perform Ctrl+V.

A nice aspect of using Paste from the clipboard: it's undoable.

Digression

There is an alternative to using the Windows `keybd_event` function to trigger things like Copy and Paste: you can send `WM_COMMAND` messages to the APL session. For example:

```
h←'#' ⌘wi 'hwndmain'  A session handle
c←57634  A ID_EDIT_COPY = 0xE122 = 57634
⌘wcall 'SendMessage' h 'WM_COMMAND' c 0
```

This uses a standard Windows menu item which would work for many Windows applications. If you search MSDN for `ID_EDIT_COPY` you will find lots of others like File New/Open/Close/Save and Edit Clear/Cut/Copy/Paste.

WordWrap Comments

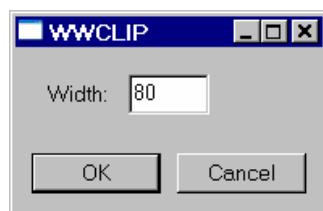
-)edit foo

```
[154]
[155] A 403 Forbidden: The server understood the request,
[156] A but is refusing to fulfill it.
[157] A Authorization will not help and the request should not be repeated.
[158] A If the request method
[159] A was not HEAD and the server wishes to make public why the request
[160] A has not been fulfilled,
[161] A it should describe the reason for the refusal in the entity body.
[162] A This status code is
[163] A commonly used when the server does not wish to reveal exactly why
[164] A the request has been
[165] A refused, or when no other response is applicable.
[166]
```

- Select lines [154] or [155] through [165] or [166]
- Press F8

```
[154]
[155] A 403 Forbidden: The server understood the request, but is refusing to fulfill
[156] A it. Authorization will not help and the request should not be repeated. If
[157] A the request method was not HEAD and the server wishes to make public why the
[158] A request has not been fulfilled, it should describe the reason for the refusal
[159] A in the entity body. This status code is commonly used when the server does
[160] A not wish to reveal exactly why the request has been refused, or when no other
[161] A response is applicable.
[162]
```

- Press Shift+F8 if you want to specify a different wordwrap right margin.



This value is remembered in your APLW.INI file.

Works With and Without Lamps

If you're editing a character vector or matrix of text, you can use]WWCLIP for a simple paragraph. (Actually, this works in functions too -- if the first non-blank character in the selection is a lamp, then lamps are used as a prefix.)

Preserves First-Line and Hanging Indents

Indentation of the first line (that is, blanks between the lamp and the text) is preserved.

Indentation on the second line is used to set a hanging indent, which will be used on all subsequent lines.

Preserves Indented Lamps

If the selected block of comments is consistently indented (that is, if the lamps are indented), WWCLIP will preserve that indentation.

How Does This Work?

]WWCLIP uses the same clipboard techniques as]ACCLIP.

There is nothing particularly fancy about the paragraph flow logic. This tool does just one paragraph at a time, and it will preserve indents. That is adequate for my purposes. I'm sure you have some more elaborate code that you could incorporate.

The idea of this presentation is to give you something that works, and more importantly, to spark ideas for more elaborate or completely different tools of your own.