Forthwrite

December
2004
Issue 127
ISSN 0265-5195

news events people reviews projects programming

FIG UK magazine

Forth Interest Group UK AGM 2004

Paths and Brushes

A Brief Introduction to FSharp

EKEY And Events

(inside front cover – blank)

Forthwrite
December
2004
Issue 127
ISSN 0265-5195

news

reviews

programming

people

Graeme Dunbar [g.r.a.dunbar@rgu.ac.uk]

Editorial

I apologise for the long delay in publication of this issue of Forthwrite. Unfortunately my spare time has reduced considerably over the past few months.

This is my last issue as editor. Regrettably there has been no interest expressed in posts of Editor or that of Regular Correspondents advertised in these pages. At the AGM it was decided that we will try a series of "rotating Editorships" with our Chairman, Jeremy Fowell, taking over for the next three issues. All future correspondence on matters relating to Forthwrite should be addressed to him.

Because of the recent problems I am in the unusual situation of publishing both news of the AGM and the report on it in this issue.

Don't forget the monthly IRC session. Our next ones are Saturday 7th August and Saturday 4th September on the IRC server called "IRCNet", channel #FIGUK from 9:00pm BST (that's 2000 UTC for international participants, see: http://www.timeanddate.com/worldclock/).

Signing off,

Graeme Dunbar

Forth Interest Group UK AGM 2004

The AGM Agenda and accounts were posted directly to members but are also included in this issue for consistency with previous years.

Saturday 30th October 2004

2pm

58 Woodland Way

MORDEN

SM4 4DS

If you are planning to attend, please check the details with our Membership Secretary, Douglas Neale, telephone 020 8542 2747, email dneale@w58morden.demon.co.uk

Agenda

1. Welcome.
2. Acceptance of apologies for absence.
3. Approval of the FIG UK 2003 AGM minutes.
4. Matters arising from the minutes of the 2003 AGM.
5. Chairman’s Report.
6. Membership Secretary’s Report.
7. Treasurer's Report.
8. WebMaster's Report.
9. Editor's Report.
   8.1 New FW Editor.
   8.2 Contingency in the case of a new Editor not being found by the end of 2004.
10. Librarian’s Report.
11. Election of Committee.
12. Addition of Special Members to Committee.
13. Projects:
    12.1 Forth CD.
    12.2 Jeremy's Forth Board.
    12.3 Member’s List Server / e-community.
14. AOCB.
15. Date of the next AGM.
16. Meeting close.

FORTH INTEREST GROUP UK

INCOME AND EXPENDITURE ACCOUNT
FOR THE YEAR ENDED 31ST MARCH 2004

	Y/E 31/3/023			
	1,046	Subscriptions		850
		Advertisements		186
	2	Interest (net of tax)		8
	1,048			1,044
633		Printing Forthwrite	362	
10		Other Printing	12	
202		Postage and Telephone	118	
82		Library Expenses	20	
89		Website Expenses	119	
5		Sundry Expenses	22	
	1,021			653
	£ 27	NET SURPLUS FOR THE YEAR		£ 391

BALANCE SHEET AS AT 31ST MARCH 2004

	As at 31/3/03			
	1,174	Accumulated Fund b/f		1,201
	27	Surplus for the Year		391
	£ 1,201	MEMBERS' FUND		£ 1,592
		Represented by:-		
	1,624	Cash at Bank		1,883
	175	Debtors and Unexpired Expenditure		202
	1,799			2,085
596		Unexpired Subscriptions	460	
2		Sundry Creditors	33	
	598			493
	£ 1,201	NET ASSETS		£ 1,592

NOTES TO THE ABOVE ACCOUNTS

The subscription is treated as representing six issues of Forthwrite (four [2003: five] were published in the year). No value has been placed on unsold back numbers, the library or other stocks.

5th June 2004.

NEVILLE A. JOSEPH

Marlowe House, Hale Road,
Wendover, Buckinghamshire.

Chartered Accountant

Douglas Neale [dneale@w58wmorden.demon.co.uk]

AGM Report

Douglas Neale

Well we had the AGM yesterday here at Morden Towers, Jeremy and one non-committee member, John Milner, turned up. I did receive apologies for absence from somebody who isn't a member as far as I know (?Frank Miller?)

John Milner agreed to have a go at translating some of the articles from the German Fig magazine, so we will wait and see the outcome of that. If successful it could give us a lot of extra material. Jeremy hadn't seen their magazine before, and was so impressed that he is going to subscribe to it himself!

Forth CD: we decided that to avoid any more delays we will go with what we have already, which is partially cleaned up from the first release you have all received. Jeremy is going to do an advert to go in the next Issue, and I am going to do little write up to go with the advert. We will be selling it at £5 a copy. Anyone who comes up with help to improve it automatically gets a free upgrade!

Future AGM's: will be totally electronic using the same sort of technology that is currently used for the Forth IRC sessions. If subsequently, we want voice and pictures we can explore those possibilities later. But this will allow all committee members to participate in their AGM without travelling to some central location.

List Server: Jeremy will open up his Fig UK list server for access to all Fig UK members, not just those who buy his kit. He will continue to moderate the list, allowing and denying access as appropriate, but it will be the members task to respond to any questions raised. By prefacing the email subject line with notations such as [ANN], [OT], [FH] members can make announcements about their new products, pass Off-Topic remarks and observations they think other members might like to know about, or make sure the Forth Hardware project member don't miss their emails. Other [xyx] notations can be invented as required.

25th. Anniversary: Nothing has been done about this, so it probably won't happen. Jeremy is very willing to host it in Birmingham but too busy to organise it.

Editor: We all fully appreciated Graeme's problems, and what we have proposed is that we have a team of rotating editors, each of which will do 3 editions. The current editor will garner material with the help of the Jeremy, which will be sent to me in electronic format. I will extend my current duties to encompass the page make-up procedure and producing the PDF files for the web site. This still leaves the task of finding somebody to do the first 3 month stint!

Membership: The current membership has dropped suddenly in the last 12 months by about 20% to the point where we now only have about 80 fee paying members! But as the treasurer has pointed out, we are still viable.

Jim Lawless [jimbo@radiks.net]
http://www.radiks.net/~jimbo

A Brief Introduction to FSharp

Jim Lawless

F# (pronounced "f sharp", also known as weforth) is a compact Forth implementation which includes words to support bindings to the Windows API.

I happened upon F# while on a quest to find a simple Forth which could interact with the Windows API. There are a number of simple Forths out there, but I wanted one that would allow me to experiment with TCP/IP.

Although the naming of the product is similar to languages specifically targeting the Microsoft .NET environment, F# is actually a pure Win32 executable file.

The most recent version of F# (v 2.21) may be downloaded at:

http://www.eforth.com.tw/academy/zip/weforth.zip

An older version which includes the Pentium assembly source code can be found here:

http://www.forth.org/svfig/Win32Forth/weforth203.zip

After unzipping F#, ensure that you start the system by clicking on FSharp.exe via the Windows XP Explorer or some other graphical shell. Invocation via the cmd.exe shell can cause various problems to manifest.

F# will initially display a dialog screen requesting the name of an input Forth file. I've been clicking on EXTEND.FEX to load up several of the extended Forth word sets.

After loading the extensions source, you'll be left in a windowed environment that resembles a console screen. You may wish to test F# to ensure that it functions correctly.

```
: tester 10 1 do space i . loop ;
```

If you then invoke the word tester, you'll see output similar to the following:

```
1  2  3  4  5  6  7  8  9  A  B  C  D  E  F ok
```

If you examine the above output, you'll note that F# begins in HEX mode.

Next, one might desire to test one of the interfaces to the Win32 API. Specifically, we'll test the MessageBoxA word which is defined in the file CONSOLEI.F.

```
0 z" Test message" z" Title bar" 0 MessageBoxA drop
```

You should see a message box window appear. Most Win32 API functions return some kind of 32-bit value. The F# words that invoke the Win32 API functions leave this return value on the stack. One must either use the return value or DROP it.

The first 0 pushed onto the stack is the handle of the window that will own the message box window. By specifying 0, the desktop window is the owning window.

The next two parameters are C-style null-terminated string constants depicted by the F# word z". z" is similar to the standard Forth s" word, except that it automatically appends a single byte with a value of zero to the end of the sequence of characters.

The first string we push is the message itself. The second string we push is the string that is to be displayed in the window's caption.

The final parameter we push onto the stack controls the types of buttons that will be displayed. Pushing zero will cause MessageBoxA to display a single "OK" button. Pushing one will cause MessageBoxA will display both "OK" and "Cancel" buttons.

```
0 z" Test message" z" Title bar" 1 MessageBoxA .
```

The MessageBoxA Win32 API function returns a 32-bit integer indicating which button was pressed. The above example will display a "1" if the user clicks on "OK" or "2" if the user clicks "Cancel."

I hope that the very modest samples above will pique the interest of other Win32 Forth users in the community. In the near future, I will provide more adventurous examples which will illustrate more useful interaction with the Win32 API.

David R Pochin [davep@sunterr.demon.co.uk]

Paths and Brushes

David R Pochin

Dave continues his tutorial on Windows graphics programming in Win32Forth.

In the Win32Forth file '**dc.f**' there are a number of methods relating to 'paths'; beginning with '**BeginPath**' and ending with '**EndPath**'. But I know nothing of 'Paths' or how they are used. Even more puzzling, '**FillPath**' is listed twice and two of the methods are commented 'Needs Brushes ?' . My first study of 'paths' is summarised in the following notes and listing.

A 'path' is created by a series of instructions between the '**BeginPath'** and '**EndPath**' statements. Often the first instruction in Win32Forth is '**MoveTo**' which places a 'pen' at a set position, subsequent instructions may be '**LineTo**', or '**PolylineTo**', or '**TextOut**' or any of the eleven instructions available in Windows 98 or twenty in NT. Not all these have been defined as Forth words in the '**dc.f**' file, so may have to be written for the occasion.

For example '**PolyBezierTo** (ptr cnt --)' is in the class WinDC, but '**Polyline'** is not. A suitable Win32Forth method close to the style in use, suggests that

```
:M PolyLine: ( ptr cnt -- )
         swap rel>abs GetHandle: dc Call Polyline ?win-error
;M
```

will do the job. Another example of this used in the listing is the method '**CloseFigure'**.

In general, the instructions with the suffix '**To**', take only one pair of co-ordinates as their parameter, assuming the present position of the 'pen' to be the beginning of the line or other operation. However '**PolyBezierTo'** takes any number of pairs of co-ordinates as control points before the end point of the curve.

It is not always necessary to make the last pair of co-ordinates the same as the first pair to close a particular path, the '**CloseFigure'** method will do this for us.

It is not even necessary that any figure is closed, it is permitted to use '**MoveTo'** and start another figure before finally using the '**EndPath'** method.

I took a little time to realise that '**CloseFigure'** closes a figure but '**EndPath**' ends a path !! Too many late nights !!? But see later.

Just ending a path is not sufficient to show the path on the display device. The path does exist but needs to be 'rendered'. Three basic rendering operations are available and written as methods in Win32Forth.

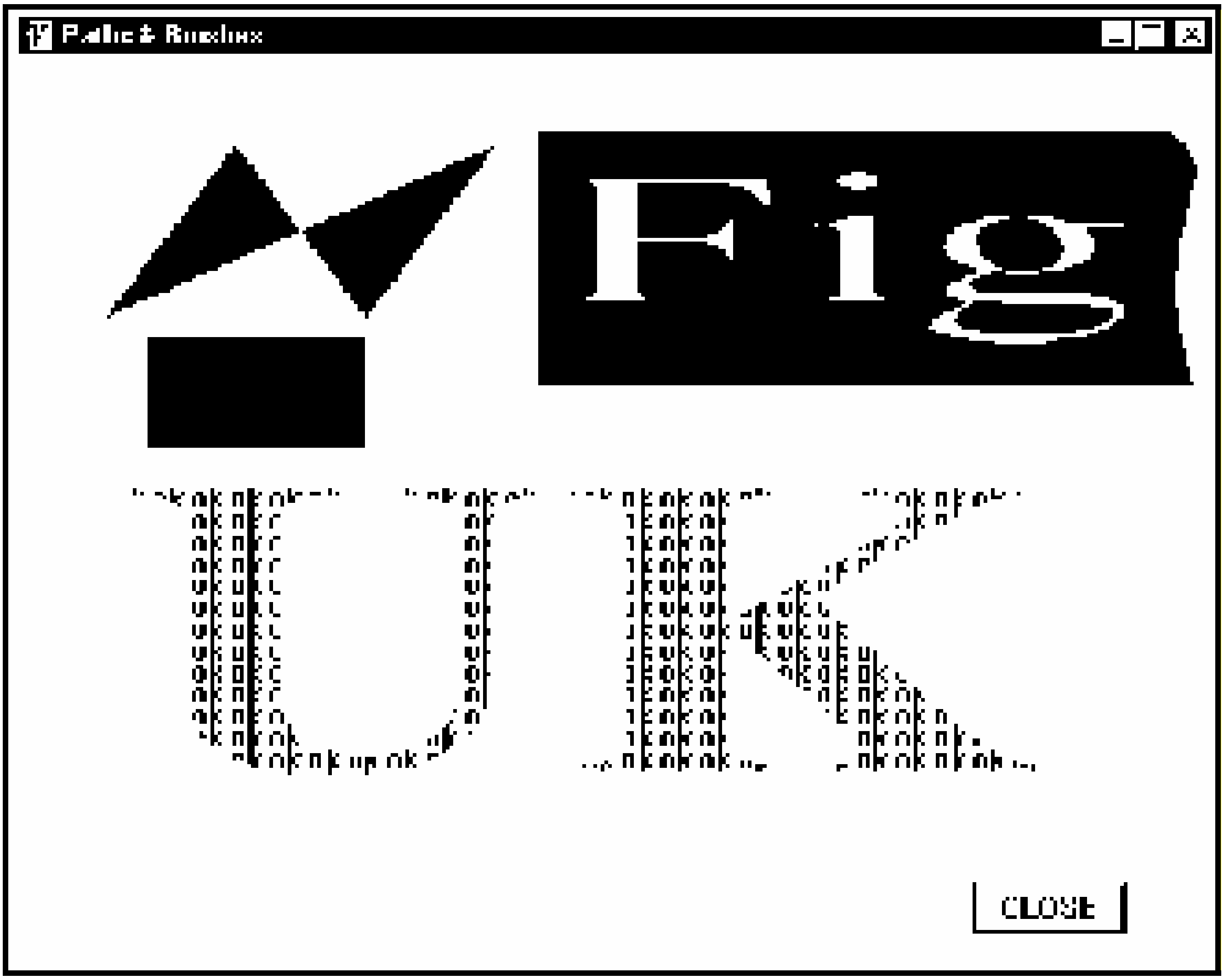
1. **StrokePath**. Uses the current pen to outline the figure.

2. **FillPath**. Uses the current brush to flood fill the interior of the figure.

   (NOTE. Make sure the figure is closed.)

3. **StrokeAndFillPath**. Combines 1 and 2 above.

There are many other functions for defining and handling paths, but my favourite so far is the Windows function '**SelectClipPath**', which converts the current path into a clipping region. It takes two parameters, the handle of the DC and a Combination Mode, which can be RGN_AND or any of the other modes used with regions.



Notes on the Listing.

The early part of the listing sets up arrays for use with the polyline and bezier curve examples and the two large fonts.

Path 1. Shows the use of the '**Polyline**', '**CloseFigure**' and '**FillPath**' functions.

Path 2. Sets up a red filled rectangle below the blue butterfly, using the '**MoveTo**', '**LineTo**', and '**CloseFigure**' functions, but it does not show on the display until the path is rendered. Uncomment the '**FillPath: dc**' line to see the rectangle. I used this path to experiment with different combinations of the lines before '**BeginPath**' and after '**EndPath**'. The path always uses the current pen colour with stroke instructions and the current brush with fill instructions. If neither pen nor brush are specified the default values are used.

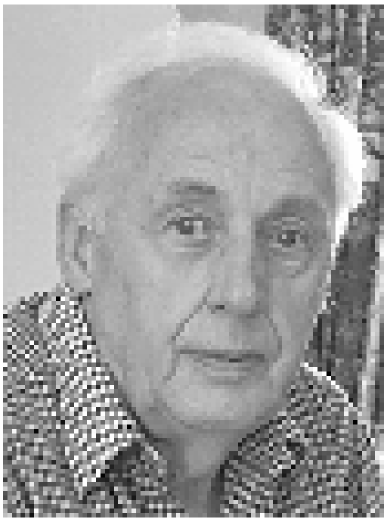
Path 3. After resetting the pen colour to black, shows the use of the '**BezierTo**' and the '**TextOut**' functions. To see the text and the background rectangle separately, comment out the line setting the background mode to 'Transparent'.

Path 4. This shows a use of the '**SelectClipPath**' function by converting the path to a clipping region. Comment out the line marked '\ CLIP' to see how this works. I spent some time experimenting with different values of the 'RGN_ ---' modes, pen and text colours to see some of the possible variations.

After spending a little time 'hands on', I think I know more about paths than before. There are other path related functions I haven't yet used, included the curious '**FlattenPath**' which converts a curve to a series of straight line segments and can be used to fit text the shape of the curve. At least, that is what the text book says! Maybe some winter evening !!

PS. If I get involved with different line widths, I think I will need to learn a little more about the different ways '**CloseFigure**' and '**LineTo**' work and explore the Windows function '**SetMitreLimit**'.

David R Pochin



Dave is now retired and living in Worcester. In an earlier life he spent some years at sea before becoming an instructor in the use of electronic navigational aids.

Back then 'radar assisted collisions' at sea were all too common due to the misuse of the new technologies by operators and visa versa.

He claims there is still work to be done and suggests you watch the humans interface with the latest 'service for customers' console at your local bank, any town, any day.

He first met Forth as a graphics application called 'White Lightning' for an early Sinclair. For more information try 'white lightning forth' on a web search. There are still sites out there.

Listing: Paths and Brushes.

```
\ Paths.F    Experiments with paths and brushes

     anew program

    :OBJECT Pathdemo <SUPER WINDOW

     ButtonControl Button_1   \ a button

     Create DOUBLECURVE       \ coordinates for use with PolyBezierTo
     ( x1 , y1 , x2 , y2 etc )
     560 , 55 , 515 , 75 , 540 , 150 ,

    \ Set Up Two Fonts.
      Font aFont
      Font bFont



    :M SetScreenFont:
      s" Times" SetFaceName: aFont
      s" Times" SetFaceName: bFont
    ;M


    :M CloseFigure:  ( -- )
             GetHandle: dc  Call CloseFigure  ?win-error
    ;M


    :M Curve1:  ( -- )
           DOUBLECURVE 3 PolyBezierTo: dc
    ;M


    :M PolyLine:  ( ptr cnt -- )
       swap rel>abs GetHandle: dc Call Polyline ?win-error
    ;M


    \ Set up an Array of Data Points for use with PolyLine
     Create POLYDATA
      ( x1 , y1 , x2 , y2 etc )
      40 , 120 , 100 , 40 , 160 , 120 , 220 , 40 ,
:M Path1:
  \ Path 1. The Blue Butterfly
               LtBlue BrushColor: dc
               BeginPath: dc
                polydata 4 PolyLine: self
                CloseFigure: Self
               EndPath: dc
               FillPath: dc
;M
```

```
:M Path2:
  \ Path2. The red rectangle
              White PenColor: dc
              LtRed BrushColor: dc
              BeginPath: dc
                 60 130 MoveTo: dc
                160 130 LineTo: dc
                160 180 LineTo: dc
                 60 180 LineTo: dc
                CloseFigure: Self
                EndPath: dc
   \ Use none or only one of the lines below in any one trial
           \   FillPath: dc
           \  StrokePath: dc
           \  StrokeAndFillPath: dc
;M
```

```
:M Path3:
  \ Path3. The 'Fig' Box
              Black PenColor: dc
              LtRed BrushColor: dc
              BeginPath: dc
                 240 35 MoveTo: dc
                 530 35 LineTo: dc
                 Curve1: self
                 240 150 LineTo: dc
                 Handle: aFont SetFont: dc
                 CloseFigure: Self
                 TRANSPARENT SetBkMode: dc
                 ALTERNATE GetHandle: dc Call SetPolyFillMode
                 245 35 s" Fig" TextOut: dc
              EndPath: dc
              StrokeAndFillPath: dc
;M
```

```
:M Path4:
 \ Path4. Embedded Text
            Green PenColor: dc
         \  LtRed SetTextColor: dc
            LtBlue SetBkColor: dc
            BeginPath: dc
              Handle: bFont SetFont: dc
              TRANSPARENT SetBkMode: dc
              45 155 s" UK" TextOut: dc
              CloseFigure: self
            EndPath: dc
          \  StrokePath: dc
          \  Use any of RGN_DIFF, RGN_OR, RGN_XOR, RGN_COPY
          \ in place of RGN_AND in the next line
            RGN_AND GetHandle: dc Call SelectClipPath drop   \ CLIP
            SYSTEM_FONT SelectStockObject: dc drop

            16 1 DO
              26 1 DO
                25 i 18 * +  175 j 10 * +  s" ok" TextOut: dc
                  LOOP
                LOOP
;M




:M Paths:
   Path1: self Path2: self Path3: self Path4: self
;M



:M ClassInit:   ( -- )
             ClassInit: super
             SetScreenFont: self
             100 Height: aFont
             80 Width: aFont
             220 Height: bFont
             120 Width: bFont
             FW_BOLD Weight: bFont
             ;M




:M ExWindowStyle: ( -- style )
             ExWindowStyle: SUPER
             ;M
```

```
:M WindowStyle: ( -- style )
                WindowStyle: SUPER
                WS_BORDER OR
                WS_OVERLAPPED OR
                ;M


:M WindowTitle: ( -- title )
                z" Paths & Brushes"
                ;M



:M StartSize:   ( -- width height )
                550 420 ;M



:M StartPos:    ( -- x y )
                100 100
                ;M



:M Close:       ( -- )
                Delete: aFont
                Delete: bFont
                Close: SUPER
                ;M



:M On_Init:     ( -- )

                IDOK            SetID:   Button_1
                self            Start:   Button_1
                440 380 70 25     Move:    Button_1
                s" CLOSE"         SetText: Button_1
                                  GetStyle: Button_1
                BS_DEFPUSHBUTTON OR
                                  SetStyle: Button_1

               Create: aFont
               Create: bFont
;M



:M On_Paint:  ( -- )          \ screen redraw procedure
            Paths: self
          ;M
```

```
:M WM_COMMAND   ( hwnd msg wparam lparam -- res )
        OVER LOWORD ( Id )
        CASE
                IDOK OF
                        Close: self
                     ENDOF
        ENDCASE
        0 ;M

;OBJECT



: DEMO          ( -- )
                Start: Pathdemo
                 ;
  cr cr .( Type DEMO to run )



  Demo


\ END OF LISTING
```

What Languages Fix

Back in Issue 122 an observation on programming languages was taken from the web page of Paul Graham, author and designer of the Arc language (http://www.paulgraham.com/fix.html) and the question "what problem does Forth fix?" was posed.

Here is a contribution from James Boyd:

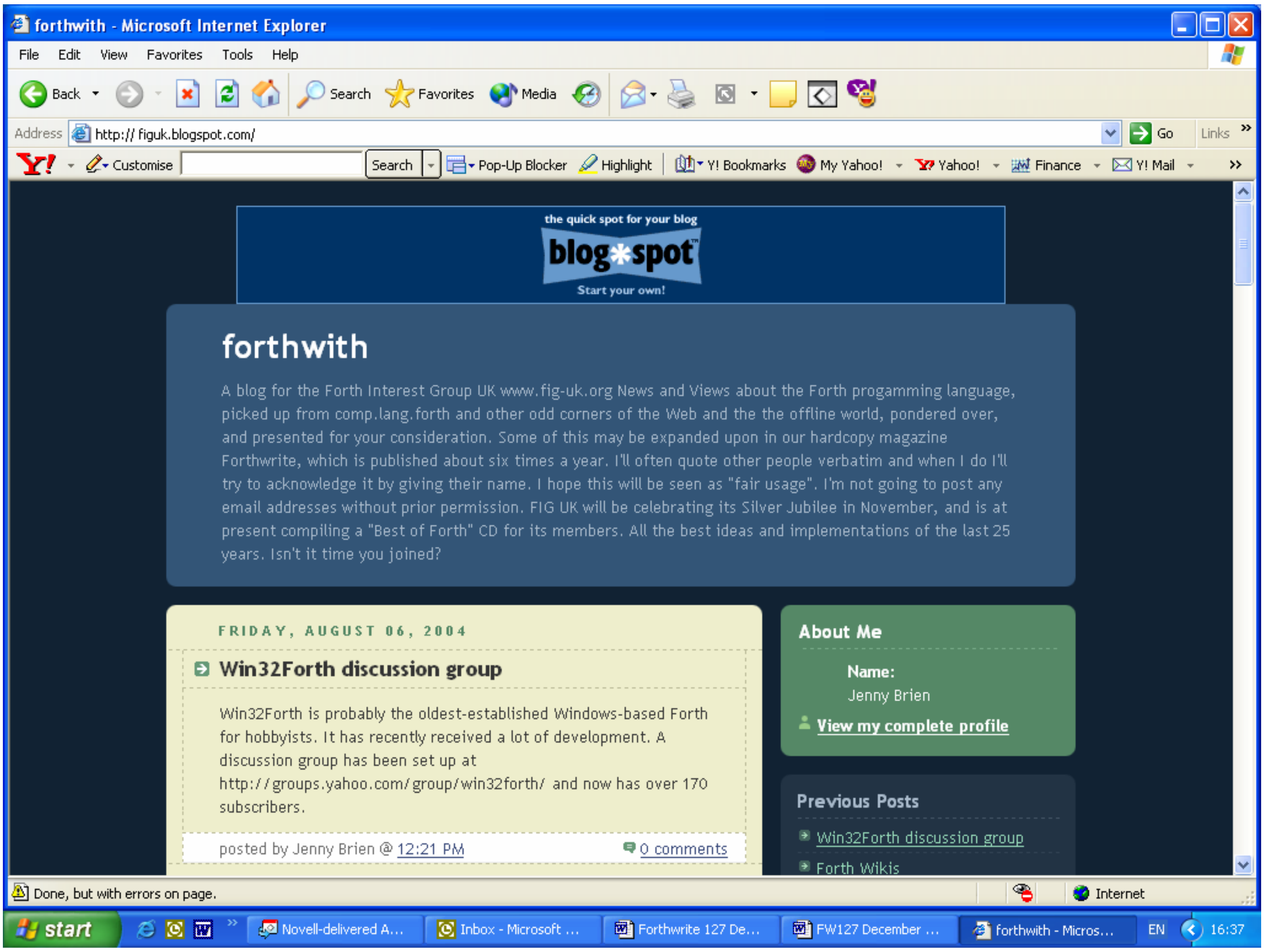
Some of my Forth books include Starting Forth second edition and Thinking Forth. Maybe one of the comments from Thinking Forth would be a good example of what Forth Fixes (what Languages Fix). To paraphrase Leo Brodie, Forth allows prototyping, a more refined way to plan.

Jenny Brien [webmaster@figuk.plus.com]

FIG UK Blog

I've put up a blog called "forthwith" at http:// figuk.blogspot.com/ , the idea being that I can post anything that is of interest either on clf or the website as it happens and try to generate some traffic. (Logs show we are getting a fairly steady 150 hits per day at present). Blogs seem to attract a lot of web searches because they change so quickly, and can be tracked easily by subscribing with an RSS newsreader (now included in both Firefox and Opera browsers.) Not much up there yet apart from a quite interesting 'What is Forth' perspective from Jeff Fox. I'm planning to post every one or two weeks so that you will have stuff that can be followed up or expanded for Forthwrite i.e. I can "more on this in the next Forthwrite issue." I can add co-authors if anyone is interested.

Jenny Brien



forthwith

A blog for the Forth Interest Group UK www.fig-uk.org News and Views about the Forth progamming language, picked up from comp.lang.forth and other odd corners of the Web and the the offline world, pondered over, and presented for your consideration. Some of this may be expanded upon in our hardcopy magazine Forthwrite, which is published about six times a year. I'll often quote other people verbatim and when I do I'll try to acknowledge it by giving their name. I hope this will be seen as "fair usage". I'm not going to post any email addresses without prior permission. FIG UK will be celebrating its Silver Jubilee in November, and is at present compiling a "Best of Forth" CD for its members. All the best ideas and implementations of the last 25 years. Isn't it time you joined?

FRIDAY, AUGUST 06, 2004

Win32Forth discussion group

Win32Forth is probably the oldest-established Windows-based Forth for hobbyists. It has recently received a lot of development. A discussion group has been set up at http://groups.yahoo.com/group/win32forth/ and now has over 170 subscribers.

posted by Jenny Brien @ 12:21 PM   0 comments

About Me

Name:
Jenny Brien

View my complete profile

Previous Posts

- Win32Forth discussion group
- Forth Wikis

Jenny Brien [webmaster@figuk.plus.com]

EKEY And Events

Jenny Brien

Jenny takes a look at what the ANS standard has to say about EKEY and events.

From dpansa10.htm#A.10.6.2.1305 (http://www.taygeta.com/forth/dpansa10.htm):

> *"EKEY provides a standard word to access a system-dependent set of raw keyboard events, including events corresponding to members of the standard character set, events corresponding to other members of the implementation-defined character set, and keystrokes that do not correspond to members of the character set.*
>
> *EKEY assumes no particular numerical correspondence between particular event code values and the values representing standard characters. On some systems, this may allow two separate keys that correspond to the same standard character to be distinguished from one another."*

(A long discussion of possible MSDOS implementations follows.)

That seems reasonable. Raw keyboard events are not portable between different types of keyboard. In practice, though, non-standard keyboards emulate standard keyboards and it becomes an OS issue, though different Forth systems could still implement EKEY differently. When porting a program from one system to another you must either:

1. Write a new EKEY to return the correct values or
2. Go through the source and change any values returned which are incompatible with the destination EKEY.

But the quote continues:

> *"In systems that combine both keyboard and mouse events into a single event stream, the single number returned by EKEY may be inadequate to represent the full range of input possibilities. In such systems, a single event record may include a time stamp, the x,y coordinates of the mouse position, the keyboard state, and the state of the mouse buttons. In such systems, it might be appropriate for EKEY to return the address of an event record from which the other information could be extracted."*

Which implies that it is even possible for EKEY to return the same value in every case - the address of an "event record". In that case, you would need to re-write any word that decodes the event record. (Does anyone actually implement EKEY in that third way? Surely it would be simpler to include the decoding within EKEY so that each keyboard event yielded a different value?) It is a reminder, though, that a program running under a GUI must respond to more event than simple keystrokes, which makes things even more complicated.

A simple keyboard-driven program might have the form:

```
BEGIN EKEY? IF
        EKEY
        decode-and-handle
        THEN
        exit-condition UNTIL
```

The “decode-and-handle” is most often some form of CASE statement, but that means that if you want to change the keys that the program responds to, you need to reload the source.

Long ago, when I was writing block editors, I decided that I would not use any control key that returned a value greater than ASCII 31. The decode-and-handle became a simple case of indexing an array of execution tokens. Since I could read and write the array, I could redefine the control key actions whenever I needed to. That becomes impractical however when the number of events that a program might respond to becomes greater.

On a GUI system, the event handler becomes something like:

```
get-gui-event
CASE
....
keydown OF
        EKEY decode-and-handle ENDOF
( handle other messages)
hand-it-back-to-OS
ENDCASE
```

No loop here - it gets called by the OS whenever there is a message for that window. The number of events that a particular window/program needs to handle is usually small but even so, writing a different case statement up front for each seems somehow un-Forthlike.

This got me thinking of how to change the list of events that a program or window responds to while it is still running. It then becomes like a little Forth system, with each event being equivalent to a Forth word. You might start begin by opening a window which understood only the default OS events and interactively add capabilities to it.

I don't know if this is even possible on any particular OS: it requires the each window, instead of having a specific event-handling routine, has a generic event-handler and a pointer to a list of events it will respond to and the words that handle them. The interface is now in the list, which may be manipulated independently. The program only supplies capabilities.

A possible implementation

An event list is a circular linked list of 3-cell elements with the format:

```
| &next | event# | xt of action |
```

And is constructed thus:

```
: EVENT-LIST              \ xt ++ ;
  CREATE HERE ,           \ point to itself
  0 ,                     \ dummy event number
  ,     ;                 \ default action
```

And events are added to the start of the list so that they can override old handlers of the same event

```
: DOES    xt   event# event-list  -- ; add to start of list
   HERE OVER @ , SWAP !  ,  ,   ;
```

Each window has its own pointer to an event-list, and a standard event handler that does whatever the OS needs and then calls

```
message# mylist @ reply
: REPLY  \  event#  event-list -- ;
\ associate default with current message
        2DUP CELL+ !
\ try each handler in turn until match found
        BEGIN
        @
        2DUP CELL+ @ = IF
        NIP 2 CELLS + @ EXECUTE EXIT THEN
        AGAIN ;
```

Because this is a circular list any unspecified event eventually matches with itself and so triggers the default action and exits.

The action for a particular event might itself use an event list to distinguish cases. For example:

```
: KEYLIST  event-list  DOES> EKEY SWAP REPLY ;
: ADDKEY   ' >BODY DOES ;
KEYLIST  mykeys
'  anaction      somekey  addkey mykeys
'  otheraction  otherkey       addkey  mykeys
.....
'  mykeys keydown some-window-event-handler DOES
```

Henry Vinerts [Volvovid@aol.com]

Across the Big Teich

Henry Vinerts

This material was prepared for Vierte Dimension by Henry Vinerts, and printed by kind permission of Forth Gesellschaft (German FIG).

Hello, Friederich, Fred and Graeme,

As fate would have it, because of Cogswell College schedules, our SVFIG meeting came one week early this month, and your shipment of the SWAPs, which I received yesterday, missed the meeting by two days. At this point, I don't know for sure which day is scheduled for the meeting in September, but I'll try to be there.

Silicon Valley FIG Meeting – August 2004

The attendance in last Saturday's meeting peaked at 15. It seemed to me that Dr. Ting's talk, which was a continuation on the subject of his current project, had attracted a handful of non-regular or new attendees. As usual, there was no break in the morning session as we listened to details of porting e-Forth to Analog Devices' ARM7 microprocessors. Actually, there are two phases to Dr. Ting's project. The first one involves what he calls "Firmware Engineering Workshop" -- a course with a text book for Chinese computer science students, and for this he has successfully downloaded e-Forth to the ARM7 in Game Boy Advance. The second has to do with development of an inexpensive digital storage oscilloscope, DSO, for short. This is still unfinished, since the UART on the ADuC7024 microcontroller has stymied the loading process of e-Forth to its ARM7. (I don't know whether I got it all correctly here. You have to forgive a foreign correspondent for trying to describe a strange culture.) The point is that the development kit from Analog Devices at $260 is inexpensive, compared to the cost of thousands of dollars for commercially available DSOs. Not counting the lunch break and ample time for the always welcomed informal information interchange ("III" for short?), the rest of the day was spent listening to an interesting historical review of the Nike missile system by Ed Thelen. You might remember from my January report that Ed's son Randy is the "father" of a homemade TTL-based Forth computer that he calls MIPPY. We are glad to have the "grandfather" also now in the SVFIG.

It is a real privilege for SVFIG to be able to meet at the college. This time we were assigned to a room that had about a dozen computers on line and we were able to follow the outline of Ed's presentation on individual screens and not have to strain the old eyes on overhead projections. I think that you might be interested in visiting http://ed-thelen.org/ to see for yourselves. It boggles my mind, as they say, to see the magnitude of the anti-aircraft defense system as it existed in the days of vacuum tubes, half a century ago. And the amount of money that is spent these days by people to defend against (or to attack) other people is simply incomprehensible to me. Here is hoping that I can send you more news about "live Forth" in another month or so.

Silicon Valley FIG Meeting – November 2004

Two days ago, on November 20th, 2004, separated by almost 10,000 airline kilometers, two different groups of Forth programming language aficionados met at their respective yearly conventions, which by now have been held for about a quarter of a century. The EuroForth conference at Dagstuhl Castle, Germany, and the Forth Day at Cogswell College in Sunnyvale, California, marched forth together for almost a full day, albeit separated by long distance. This occasion made me think of a clipping that I had saved from the Sunday, April 11, 1993, issue of the San Francisco Examiner. It is from "Ask Dvorak" question- and answer column in the paper's "Computers & Technology" section. At that time Dvorak was the editor of the PC Magazine; later on I heard his keynote address at one of the Embedded Systems Conferences. Anyway, in the aforementioned article Dvorak responded to a reader's question about programming languages and referred to Forth as "a peculiar development language, which has a cult following." Back then, as a third-year novice in the SVFIG, I didn't appreciate his comment, but today, upon rechecking the definitions of "peculiar" and "cult" in my dictionaries and having witnessed the long-term devotion of the forthers, I cannot really take offense with Dvorak's semantics. What is your opinion?

To support my "quarter century" statement, I am reading from page 79 in Forth Dimensions I/6 (March/April, 1980): "The first meeting of the FORTH MODIFICATION LABORATORY (FORML) was held at Imperial College, London, January 8-10, 1980. Representatives of both the European FORTH Users' Group and FIG attended." From later sources, I see that the second FORML conference was held in November 1980 at Asilomar in California, which saw its last FORML in 1999, the year the chief organizer, Rob Reiling, passed away.

To support my "long-term devotion" statement, I am going through old volumes of Forth Dimensions and looking the list of authors of articles from FORML, EuroFORML, and EuroForths from 1980 through 1993. The following, who were present at the SVFIG Forth Day last Saturday, are either the original forthers or long-time devotees:

Chuck Moore, Glen Haydon, Bill Ragsdale, John Cassady, Andy Korsak, Robert Patten, John Rible, Robert Smith, C.H. Ting, John Carpenter, Jeff Fox, John Hall, Dave Jaffe, George Perry, Jay McKnight, Kevin Appert. There were others at the meeting, whom I haven't listed, just because I am in a hurry to mention a few prominent names from the other side of the "Big Pond," who also appear in the above list: Klaus Schleisiek, Anton Ertl, Ulrich Hoffmann, Peter Knaggs, Wolf Wejgaard. There are many more whom I know who did not make it to our Forth Day and also those I don't know, who, I am sure, were present at the EuroForth conference. Judge for yourselves, my friends. It is devotion to follow the same flag for so many years!

With around 35 attendees this time our Forth Day showed no reduction in number of "believers." Throughout the day nine people came to the podium and all of them talked about what they have been doing with Forth. Dave Jaffe has already listed their names and subjects on http://www.forth.org/svfig/, therefore I shall not repeat them in detail. Personally, I most enjoyed Jeff Fox's lecture on using one's whole brain to do Forth, rather than being limited to one side of it. Citing Marshall McLuhan, Jeff explained why forthers, who are inclined to be right-brained, are not understood by the majority of programmers who are brought up to work in a rigid left-brained society.

Chuck loves life in Sierra City, above the snow line in the California foothills, where one can still see stars at night and walk on the state highway to dine at a restaurant in town (of 500 people) without worrying about being run over. New version of ColorForth will probably appear in about six months, and OKAD 2 has been busy designing new Forth chips.

I have already written briefly to Friederich about the previous SVFIG meetings, but did'nt have much to report. In October Dr. Ting described his Digital Storage Oscilloscope project and Dave Jaffe talked about an automobile-driving simulator that he is enhancing for purposes of testing brain-injured or older persons before their licenses may be renewed. In September also only about a dozen people showed up to listen to Ting's earlier experiences in the above mentioned project and to LaFarr Stewart's presentation of some less-known or forgotten aspects of number theory, such as modular arithmetic.

I forgot to mention some important items from the Forth Day:

1. Dr. Ting unfailingly fed all of the troops with his traditional and delicious barbecue lunch.
2. All of the German Forth society's 20-year commemorative swap-dragon pins were handed out to happy recipients.
3. Peculiarly enough, no women showed up at the meeting (as the group picture which Kevin Appert took after lunch will indicate, provided that it finds its way across the ocean).

4. We did receive pictures and a video clip by e-mail from Dagstuhl at about the same time, and you may tell Klaus Schleisiek that there still are quite a number of old forthers who recognize him.

Oh, one more note from Silicon Valley: On October 20th three of us went to hear Niklaus Wirth talk about his career (or should I say "read" his autobiography?) at the Computer History Museum. Interesting, but not very entertaining, yet I doubt that Donald Knuth (he was in the audience) could have done better if he had had to do the same thing in German at the ETH.

The year may expire before I write to you again, therefore let me now wish you all a pleasant and peaceful holiday season and a happy New Year.

Henry

Hello, again!

After I sent my e-mail yesterday, the group picture and thumbnail shots of our Forth Day attendees appeared on http://www.forth.org/svfig/fd2004/photos.html

I hope that the group picture found its way to Klaus's designated e-mail address at Dagstuhl. We are still working on matching the 31 persons in the picture with their correct names, and while at that, I have a request for the editors of your magazines to add a few "oldtimers" to my list in yesterday's e-mail, lest I have to apologize to them for forgetting:

Please, add Alan Furman and John Peters in the list between George Perry and Jay McKnight. As far as I know, they both qualify as "oldtimers," for they were there when I first started at SVFIG.

Tschuess,

Henry

P.S. Tell me whether I correctly see the golden swap-dragon pin on Chuck Moore's shirt? He told me that he had received it, and as far as my eyes can tell, it appears on the left side of his shirt. (I wore mine on the right side.)

Deutsche Forth-Gesellschaft

Would you like to brush up on your German and at the same time get first-hand information about the activities of fellow Forth-ers in Germany?

Become a member of the German Forth Society for 32 Euro (£22) per year (16 Euro (11 Pound) for students and retirees). Read about programs, projects, vendors and our annual conventions in the quarterly issues of Vierte Dimension.

For more information, please contact the German Forth Society at the e-mail address:

SECRETARY@FORTH-EV.DE

or visit:

http://www.forth-ev.de/

or write to:

Forth-Gesellschaft e.V.
Postfach 19 02 25
80602 München
Germany

Telephone:

(089) 1 23 47 84

Joe Anderson [jia@jia.abel.co.uk]
phone: 0131 662 4007

Vierte Dimension 2/2004

Joe Anderson

Joe provides a look at the latest issue of the German FIG magazine.

Editorial. 4

Friederich Prinz
[vd@forth-ev.de]

Friederich promises for the next edition of Vierte Dimension reports on the really successful Jubilee Forth Conference in April on Fehmarn. One continuous party! After that he welcomes three new members. In addition he emphasizes the uniquely good cooperation with the British Forth adherents. He mentions the lively interest aroused in Vierte Dimension readers by both articles by James Boyd on non-deterministic machines. And he praises Graeme Dunbar's readiness to provide the necessary documents for translation into German.

Readers' Letters. 5

Readers letters from: Ulrich Paul (on non-deterministic machines and sub-optimal solutions),Ulrich Paul (interchanging variables without an intermediary > store),Fred Behringer (correction to remarks on Euler paths),Carsten Strotman (more on USB developments in Forth),Martin Bitter (an error in ZF's MARK).

A Virtual Non-deterministic Machine in Forth. 8

James A. Boyd
[JimBoyd@techemail.com]

Second part of the noteworthy article by James A. Boyd (see Forthwrite 124) in the translation by Friederich Prinz. (James has indeed also received the FIG UK Forthwrite award for it).

Advertisements. 13

Notices on membership adverts for FIG UK and the Dutch Forth-gebruikersgroep.

Non-deterministic machines? 13

Friederich Prinz
http://home.abel.co.uk/webmail/src/compose.php?send_to=VD%40FORTH-EV.DE

Questions (from the Editor of Vierte Dimension) to specialists in IT and Mathematics (Bernd Paysan and Fred Behringer).

Forth Programmes for the Articles by James Boyd. 15

No articles without Forth programmes! Here are the programmes for both articles (Parts 1 and 2) by James Boyd. The articles have appeared in their original version in Forthwrite as well.

dah8300 - Hitachi Disassembler. *20*

Risto A. Karola

The translator of this article, Michael Kalus, has been involved for the past four years with the Hitachi micro-controller H8/300 in the construction of Lego-robots. In order to be able to check out the many reports on the Internet on machine-programming with the H8/300 he was on the hunt for a disassembler. And so there came to his assistance this already nine-year-old contribution by Risto A. Karola: http://www.pcuf.fi/~rak/index-eng.html

Reviews. *21*

Fred Behringer

Fred Behringer reviews Forthwrite 124 and Vijgeblaadje 41

Greetings from California. *22*

Henry Vinerts

Henry’s report on the SVFig meetings in December 2003 and January 2004, in the translation by Thomas Beierlein. The original reports also appeared in Forthwrite. Henry draws the reader's attention to http://www.forth.org, where the Webmaster David Jaffe also reports on the SVFIG meetings.

How to put Forth on Knoppix-Linux. *23*

Carsten Strotman

Every year Linux-day takes place in Karlsruhe. The Forth-Gesellschaft has made a point of being represented there with a stand. For interested visitors a Knoppix-Linux CD with inbuilt Forth systems was developed. Carsten describes the production process for "interested puzzlers". (Remastering of the Knoppix-CD).

S-Record Data-packets for the RCX. *25*

Friederich Prinz

Binary code to be transmitted between the PC and the Lego-robot component RCX over the infra-red link is sent in predefined "packets", the S-records. The format of the S-records originated with Motorola, and has been established for such transfer of data. Friederich sees redundancies and superfluous control encumbrances that can be eliminated. In addition the implemented S-records are not "long" enough for him. In his considerations Holon from Wolf Wejgaard plays a large part.

Congratulations to the Dragon-prizewinner. *32*

Friederich Prinz

The lucky winner this time is Ewald Rieger (for many years work with Forth and for his initiative on behalf of the Forth-Gesellschaft).

AI and Forth. *33*

Ulrich Paul

The author ponders the question, "What is intelligence?" A nomad who cannot count up to three is lost in Europe, a teacher from an advanced institute in Europe cannot cope in the desert. "IQ tests are pure nonsense. It depends on the cultural needs. But a measuring system for intelligence cannot and must not be made dependent the cultural background." Further critical expressions: neural network, fuzzy logic, induction, abstraction, deduction, abilities, knowledge, wisdom. "What has that to do with Forth?" There follow thoughts about DEFER, CREATE-DOES, IMMEDIATE, FORGET - "A CREATE-DOES construction can represent a neuron, but not its varying connections to other neurons. For that you need nested CREATE-DOES constructions."

WANTED:

Computer artefacts to help in teaching the history of computers

For example:

- floppy disks of 5.25" and 8" size
- punched paper tape
- Hollerith cards
- ferrite core memory
- transistor-based processor cards

Postage refunded.

Please raid your attic/loft/shed in a good cause.

Many thanks,

Chris Jakeman [cjakeman@bigfoot.com]

(trying to improve his teaching at Peterborough Regional College)

FIG UK

FIG UK Contacts and Information

Chairman **Jeremy Fowell,**
11 Hitches Lane, EDGEBASTON B15 2LS
0121 440 1809 jeremy.fowell@btinternet.com

Secretary **Douglas Neale,**
58 Woodland Way, MORDEN SM4 4DS
020 8542 2747 dneale@w58wmorden.demon.co.uk

Editor **Jeremy Fowell,**

Treasurer **Neville Joseph,**
Marlowe House, Hale Road, WENDOVER HP22 6NE
01296 62 3167 naj@najoseph.demon.co.uk

Webmaster **Jenny Brien,**
Windy Hill, Drumkeen, BALLINAMALLARD,
Co. Fermanagh BT94 2HJ
02866 388 253 webmaster@figuk.plus.com

Librarian **Graeme Dunbar,**
School of Engineering, The Robert Gordon
University, Schoolhill, ABERDEEN AB10 1FR
01224 262415 g.r.a.dunbar@rgu.ac.uk

Membership enquiries, renewals and changes of address to Douglas.
Technical enquiries and anything for publication to Jeremy.
Borrowing requests for books, magazines and proceedings to Graeme.

FIG UK Web Site

For indexes to Forthwrite, the FIG UK Library and much more, see http://www.fig-uk.org

FIG UK Membership

Payment entitles you to 6 issues of Forthwrite magazine and our membership services for that period (about a year). Fees are:

National and international	£12
International served by airmail	£22
Corporate	£36 (3 copies of each issue)

Forthwrite Deliveries

Your membership number appears on your envelope label. Please quote it in correspondence to us. Look out for the message "SUBS NOW DUE" on your sixth and last issue and please complete the renewal form enclosed. Overseas members can opt to pay the higher price for airmail delivery.

Copyright

Copyright of each individual article rests with its author. Publication implies permission for FIG UK to reproduce the material in a variety of forms and media including through the Internet.

FIG UK Services to Members

Magazine

Forthwrite is our regular magazine, which has been in publication for over 100 issues. Most of the contributions come from our own members. The Editor is always ready to assist new authors wishing to share their experiences of the Forth world.

Library

Our library provides a service unmatched by any other FIG chapter. Not only are all the major books available, but also conference proceedings, back-issues of Forthwrite and also of the magazine of International FIG, Forth Dimensions. The price of a loan is simply the cost of postage out and back.

Web Site

Jenny Brien maintains our web site at http://www.fig-uk.org. She publishes details of FIG UK projects, a regularly-updated Forth News report, indexes to the Forthwrite magazine and the library as well as specialist contributions such as "Build Your Own Forth" and links to other sites. Don't forget to check out the "FIG UK Hall of Fame".

IRC

Software for accessing Internet Relay Chat is free and easy to use. FIG UK members (and a few others too) get together on the #FIG UK channel every month. Check Forthwrite for details.

Members

The members are our greatest asset. If you have a problem, don't struggle in silence - someone will always be able to help. Do consider joining one of our joint projects. Undertaken by informal groups of members, these are very successful and an excellent way to gain both experience and good friends.

Beyond the UK

FIG UK has links with International FIG, the German Forth-Gesellschaft and the Dutch Forth Users Group. Some of our members have multiple memberships and we report progress and special events. FIG UK has attracted a core of overseas members; please ask if you want an accelerated postal delivery for your Forthwrite.

Back cover (Advert)