



Forth as Preferred Development Environment

FIGUK magazine:

XML and Forth

F11-UK FIG Hardware Project

euroFORTH 2000 – The Report

Forth Application Techniques

Vierte Dimension 4/00

Letters

January 2001

Issue 110

projects

F11-UK FIG Hardware Project ... 12

Input from users and LCD displays from Tesco.

news

Forth News 3

What's happening and where

Vierte Dimension 4/00 27

The latest from Germany

reviews

Forth Application Techniques ... 30

This issue's new book

euroFORTH 2000 – The Report .. 23

What you missed and why you should read the published papers

programming

Forth Preferred as Development Environment 8

Switching to Forth bring advantages

XML and Forth 16

XML is important and not difficult. Give it a try.

people

Letters 32

Your input is always welcome.



Editorial

Lots of things happening in Forth at present, as you can see by Dave's extra long Forth News. I'm especially pleased to learn about the 2 new books - see Robert Ives' review in this issue.

Our lead article is a report by John Matthews who has been helping a UK manufacturer switch to Forth for a new product. With the high-performance of today's commercial compilers, the main hurdle in switching to Forth is now a shortage of expertise, so congratulations, John, on sharing yours successfully.

Welcome this time to new member Chris Ranklin, a professional programmer from Leeds.

Les Kendall is another professional, whose debut article appears in this issue, showing us how to join in the fast-spreading world of XML.

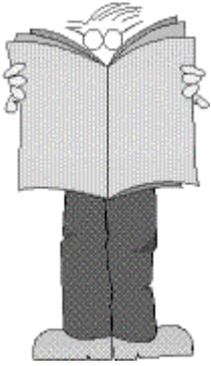
There were some fascinating papers at euroFORTH 2000 and they are already available from the Internet (and FIG UK's Library). Thanks to Howard Oakford for his entertaining report.

There is so much going on in Forth at present that we've provided an extra 2-page session Behind the News to keep you updated.

Don't forget the monthly IRC session. Our next one is Saturday 3rd February on channel #FIGUK from 9:00pm.

Until next time, keep on Forthing,

Chris Jakenan



Dave Abrahams
0161 477 2315
d.j.abrahams@cwcom.net

Forth News

People

Chuck Moore

The Annual Forth Day meeting of the Silicon Valley Chapter of the Forth Interest Group was held on 11/11/00. It ended with Chuck Moore's traditional Fireside Chat and Jeff Fox has posted a transcript at:

<http://www.UltraTechnology.com/fsc2000.htm>

Malcolm Bugler

Malcolm Bugler is MD of Malvatronics Ltd and organiser of the recent euroFORTH 2000. He is off to work in the USA on a four year contract but will continue to operate Malvatronics.

Malcolm says "As I am actually moving to Rochester in New York state, it may be that I am able to resurrect the Rochester Forth conference. However, this would depend on getting sufficient support for possible US delegates."

<http://www.malvatronics.co.uk>

Hans Nordstrom

Hans, one of our Swedish members, has published comprehensive glossaries (in English) for both Pygmy and Quartus Forth. These are on his web site along with some interesting biography at <http://www.abc.se/~m989/Forth/Gena4th.htm>

Commercial Systems

Version PROFORTH 3.22



MPE has released version 3.22 of ProForth VFX for Windows. Two major new features, are user programmable Studio IDE and the new DFX debugger. Both are supplied with source code in all versions.

Full details and a 30 day evaluation version are available at:

<http://www.mpeltd.demon.co.uk>

SWIFTX Version 3.0

FORTH Inc. has announced the release of version 3.0 of the SwiftX line of cross-compilers which now have smaller, faster code and enhanced project management features. The price has dropped too!

See <http://www.forth.com> for details.

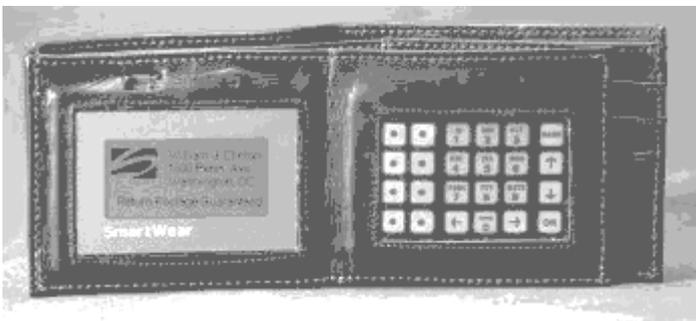
Quartus Forth

New for the Quartus Forth on-board Forth compiler for the Palm OS are MathLib and NewFloatManager functions.

This enhancement allows your apps to go beyond the single-precision floating-point routines built into the Quartus Forth compiler, and make use of full IEEE double-precision calculations.

<http://quartus.net/products/forth/>

WearLogic



After discussion on comp.lang.forth about Linux in a wrist-watch, (<http://www.research.ibm.com/WearableComputing/factsheet.html>), Elizabeth

Rather pointed out one of the products from WearLogic at

<http://www.wearlogic.com>

“...their "wallet" is based on an AVR 8515 (soon to be upgraded to a Mega103) programmed in Forth using SwiftX.”

Free Systems

DELTA Forth

The full Java source code for DELTA Forth - known for its platform independence - has been released. It is now free of charge and you may download your copy from:

<http://www.dataman.ro/dforth>

Included is an algebraic to RPN converter that is able to generate Forth code for solving expressions.

Ficl release 2.04

Ficl (Forth inspired command language) version 2.04 is now available for web download from:

<http://www.taygeta.com/ficl.html>

Ficl is a lightweight, efficient language designed to be incorporated into other programs, including (especially) firmware based systems.

StrongForth version 0.03

Stephan Becher has made version 0.03 of Strong Forth available at:

<http://home.t-online.de/home/s.becher/forth/>

StrongForth uses “static type-checking and object overloading.”

KForth

kForth for Windows (95/98/NT) has been rebuilt using the free Cygwin development tools for Windows (see <http://cygwin.com>).

The new version will allow anyone to modify and rebuild the executable using the Cygwin tools.

<http://ourworld.compuserve.com/homepages/krishnamyneni/ccre/ccre.htm>

GForth version 0.5.0

Anton Ertl has released Gforth 0.5.0 (source distribution) at

<http://www.complang.tuwien.ac.at/forth/gforth/>

Gforth is a fast and portable implementation of the ANS Forth language and runs under Unix, Win95, OS/2, and DOS and should not be hard to port to other systems supported by GCC.

Examples of good documentation encouraging people to explore Forth are rare, so it is especially good to hear that the manual has been greatly improved, thanks mostly to FIG UK member Neal Crook.

LIB4TH Forth as a DLL

Peter Recktenwald has announced the first working version (0.0.51) of "lib4th". Documentation and source code are available at:

<http://home.snafu.de/phpr/lhpf8.html>

Lib4th is an ANS compliant Forth kernel supplied as a linux/ELF dll for 2.2.14+ (2.2 versions only) kernel and i586+/k6+ cpu-s.

Peter would welcome any comments or suggestions.

Camel Forth for the Z80, Z180 & Rabbit 2000

Douglas Beattie has ported Camel Forth to the Rabbit 2000 processor. "The Rabbit_2000 is a highly-integrated microcontroller based on the instruction set and CPU model of the Zilog Z180. It is a new architecture with superior execution times, optimized for embedded applications, and rich in peripheral functions."

For a free download visit:

<http://www2.whidbey.net/~beattidp/>

System Extensions and Utilities

CTRAN Formula to code translator

Julian Noble announced on comp.lang.forth that the next version of his 4-function formula to code translator is now available on:
<http://Landau1.phys.virginia.edu/classes/551/>

"The chief virtue is readability and ease of creation".

Win32Forth toolset



Jos van der Ven offers a toolset for Win32Forth users at: <http://home.planet.nl/~josv>

Included are over 2000 error-codes for Windows.

Forth Hardware

ForthChip.com

Richard Westmoreland has added a discussion board to his web site devoted to Forth hardware.

<http://www.forthchip.com/discussions.htm>

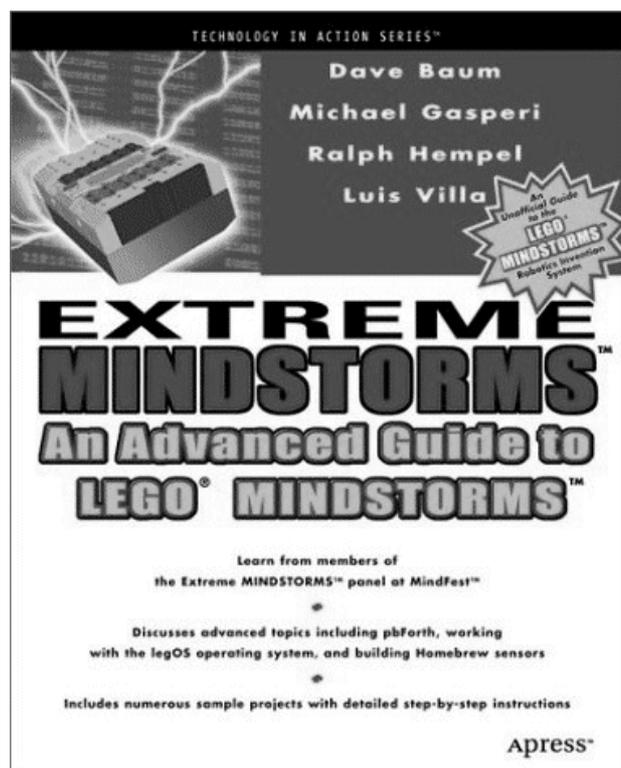
Books

Forth Application Techniques

Forth Inc. has announced the publication of a new book on Forth Application Techniques. This is an introductory tutorial which can be ordered from: <http://www.forth.com> or sales@forth.com

See the review later in this issue.

Extreme Mindstorms: an Advanced Guide to Lego Mindstorms



This is a new book on Lego Mindstorms with two chapters by Ralph Hempel on pbForth, his Forth for the LEGO Robotics Discovery Kit. This book is now available from Amazon:

<http://www.amazon.com/exec/obidos/ASIN/1893115844/hempeldesigngroup>

ANS Forth

Web site for discussion of ANS proposals

Anton Ertl has a web page at:

<http://www.complang.tuwien.ac.at/forth/ansforth/proposals.html>

which discusses proposals for the next ANS Forth.

He is seeking comments, new proposals, agreement (and disagreement too.)

Solutions proposed include:

- taking input from a file
- i/o buffering and re-direction
- improvements for THROW
- enabling optimisations for DOES> and EXECUTE

Miscellaneous

Forth in schools

Martin Bitter (German FIG) provides tuition in Forth-controlled robots at a local Hauptschule (i.e. for least able 20%). His team won 4th prize out of about 30 in a LEGO-RCX competition.

PbForth for the Lego MindStorm

Hempel Design Group	/ Home / lego / pbFORTH / Forth for Mindstorms
---------------------	---

Ralph Hempel's pbForth appeared in a November issue of Wired magazine on the dramatic success of Lego's MINDSTORM.

<http://www.hempeldesigngroup.com/Lego/pbFORTH/index.html>

Doug Neale, our Membership Secretary, tells me he was surprised to find that members were responding so quickly to the renewal notices he sent out. Then he noticed that he was using an old form by mistake, with the £10 for 6 issues price. Could this be just a coincidence, we wonder?

He hopes members will realise that this was an oversight, not a special offer, and that £10 will be sufficient only for 5 issues of the magazine.

Forth Preferred as Development Environment

John Matthews

Now retired, John still finds ways to promote Forth.

Maurice Yates, MD of Advanced Energy Monitoring Systems Ltd., has revealed that his next generation of 'Yatesmeters' is being developed using Forth. The Yatesmeter is the world's leading instrument for the on-line monitoring of pumps, especially pump efficiency and flow. It is not well known that 20% of the total generation of world electricity is used in pumping water or that improvements in pump operation could save half of this.

The use of Forth was first suggested to Maurice by NEL many years ago but it was the availability of FIG UK member John Matthews after early retirement from Exeter University as an advisor that made the development practical. John continues working for 2 days/month even though he is now an OAP!

The Yatesmeter is based on the 'thermodynamic method' whereby the small rise in temperature of the fluid pumped is related to pump losses. Owing to its being only a few milliK, previous workers had rejected the method as not viable but AEMS have devised a reliable way not only of measuring it, but also a method of deriving fluid flow through the pump that is more accurate than most conventional flowmeters.

The new instrument comprises a set of intelligent sensor cards linked to a master processor by a communications bus. This gives flexibility of configuration for multiple pump installations and for the development of new applications.



The Yatesmeter control module

Forth allows the communications link to be tested thoroughly. The processors at both ends of the link can be monitored simultaneously under real-time conditions while the application is running. MPE Ltd supplied the development system, and their help and support are acknowledged.

For more information visit www.yatesmeter.co.uk

Forthwrite reaches further

Now that older issues of Forthwrite can be downloaded from our web site, our influence is growing.

In response to a newsgroup thread in which Rick van Norman mentioned the Canon Cat, non-member Felix Franz fefr@gmx.net posted the following message. He has clearly downloaded and read Forthwrite with care.

The June 2000 issue of Forthwrite contains an article about the Canon Cat ("The Canon Cat -- The most promising forth machine to date?" written by Neal Crook.)

The pdf-version is available at: <http://forth.org.uk/articles.htm>

Behind the News

Chris Jakeman

The Forth world seems to have been especially busy of late, so here is an extended Editorial to report some of the many things that have been happening.

Books

Did you notice that two books have been published? These are “Forth Application Techniques” from Forth Inc. and “Extreme Mindstorms: an Advanced Guide to Lego Mindstorms” from Ralph Hempel. It seems that the warnings of a couple of years back, that no more Forth books would be published in paper form, have proved pessimistic. There’s plenty of scope for more to come.

Downloads

I have said before that there is much more Forth activity taking place than is apparent from following comp.lang.forth or attending euroFORTH. This was discussed briefly on the newsgroup recently.

For example, Jenny Brien has recently added a monitor to our web-site which counts the downloads. This is far more accurate than the visible page counter reporting the number of visits to the download page. The total downloads for the August issue of Forthwrite (during the 2 months after posting) were 166, so the electronic version of Forthwrite has more than doubled our readership.

Tom Zimmer, author of Win32Forth, reported, “I haven't mentioned it here before, but at least twice last year my internet provider threatened to shut down my FTP site for Win32Forth because it was near the 100MB/day download limit. I am not complaining mind you, since my provider promised to put me back on line at the start of the following month.”

The author of Quartus Forth, Neal Bridges, wrote, “Another data point: The evaluation version of Quartus Forth (for the Palm PDA) gets approximately 30 downloads per day from the Quartus.net site (plus an unknown number from other repositories).” This is a very high number.

Let’s give the last word to Philip Koopman of Carnegie Mellon University and author of the book “Stack Machines: The New Wave” which is available on-line at http://www.cs.cmu.edu/~koopman/stack_computers/index.html “I get about 25-30 downloads of the complete zip file of my stack computers book every week. That number of accesses has been fairly steady for quite a long time. While that's not strictly a Forth metric (many of them are really after Java information I think), it gives an idea that things aren't completely dead, even for a 10-year-old book.”

Based on e-mail I get, my perception is that the majority of the downloads are from Asia and the former Eastern Bloc countries, although certainly not all of them are from those areas.”

Applications

I am sometimes asked why we can't have more items on applications in Forthwrite. The shortage of such items is also suggested as evidence for a decline in the use of Forth. In practice, applications do not often appear in print because the authors are too busy earning a living by writing Forth to write about it. There is also the additional hurdle of clearing an article with your customer before publication.

If you read this issue with care, you should find references to no fewer than 10 applications. To save you time in searching, I list them here:

- on page 3, the electronic wallet from WearLogic
- on page 8, the Yatesmeter, from AEMS
- on page 12, the F11-UK used for motor control
- on page 16, two applications undertaken by Les Kendall
- on page 23, three applications reported in detail at euroFORTH 2000
(the security package, the servo controller and the PPP comms)
- on page 27, applications published in Vierte Dimension
(the Lego robot and the digital signature)

This is probably a record for recent issues of Forthwrite and suggests that the Forth world is thriving.

Robots – USA and Germany

The last issue of Forthwrite carried a letter by Ralph Hempel, the author of pbForth for Lego robots, and mentioned his efforts at 2 kid's camps last summer teaching youngsters to program Lego robots in Forth.

Did you spot the item on Forth News on Forth-controlled robots? FIG UK member Fred Behringer has been helping Martin Bitter do something very similar at Vaterstetten near Munich last July. He writes about the experience in Vierte Dimension 4/00.

He now expects to be presenting a twice-a-month course on pbForth for robots for youngsters aged 11 to 13 years old. So congratulations to another retired FIG UK member spreading the word!



0121 440 1809
jeremy.fowell@btinternet.com

F11-UK FIG Hardware Project

News of this project continues to arrive. Here are some extracts to show what is going on. We eagerly await details of the first applications carried out by FIG UK members with the new board.

Dave Abrahams reported on the F11-UK news list that he had (at last) got around to completing it. Here is Jeremy's reply:

On 18 December 2000 at 16:08 David Abrahams wrote:

```
> I'm still here and by coincidence I have just built  
> my F11-UK kit (better late than never). It seemed  
> to work first time but I cannot re-establish the  
> comms link after hitting escape, I have to repeat  
> the whole download process each time. I will go  
> over the previous emails in the list to see if  
> there are any clues.
```

I just ran a quick test on my F11-UK board to confirm that after you press escape to return to 80x86 Pygmy, you then need to type `TERM` to reconnect to PygmyHC11 running on the F11-UK board. (Apologies, this should have been in the F11-UK User Guide).

Just typing the word `DUMB` in 80x86 Pygmy to run the simple terminal program fails to work it would seem.

I hope the Christmas break will at last give me a chance to test some of the code I have written for incremental compiling and also a single stepper for colon definitions. When done, I will make these available since the improved compiling should reduce the download time by a large amount.

I have been using an F11-UK board to program the

on-chip EPROM in the E9 version of the HC11 with great success. The one thing you have to watch is that the CONFIG register is set correctly first. (PCBUG11 manual page 4-8). PygmyHC11 has been in use for a month or two now running an industrial 3-phase motor control. I have been surprised how stable both the hardware and software have been. (There is a cable around 5 cm away from the HC11 carrying up to 80 amps pulsed DC !)

“new code ... adds incremental compiling”

Recently I have been thinking about what could be the cheapest usable 16-key keypad to connect to the F11-UK board, and what might be the lowest cost display device.

Well I think I may have found the answer. Yesterday I went to our local Tesco store and noticed a pocket calculator for £3.00 ! It has a large 85 x 62mm 23 key keypad and the LCD has eight 15mm high digits (model EZ 2 GRIP-D as far as I can tell).

It looks like the case is held together with 5 screws and the keypad and display just clip into place. It also has some sort of solar cell. I bought one with a translucent green case (sort of Apple Mac style) so you can see the internals, which I thought was really cool. However my wife informed me this was completely the wrong colour and that I had no fashion sense at all, so you never can tell.

After a bit of reverse-engineering I will report back.

Jeff Penn was one of the first members to assemble the kit and reports his findings here.

I was asked by Chris to write about my experiences building the F11-UK kit. My description is brief, a testament to the professionalism that went into producing the kit.

The F11-UK kit supplies the PCB, components and instructions for assembling the controller board. A copy of the PygmyHC11 software is forwarded via email (although my copy needed to be posted on a floppy).

All that's missing from the kit is a power adaptor, a serial cable (details given), and two files only available with a registered copy of Pygmy Forth.

Registering Pygmy from Frank Sergeant costs \$20, on receipt of which you are sent a "bonus disk" via email. The "disk" includes a 68HC11 assembler, and serial I/O routines required to compile the F11-UK code and communicate with the board. For the power supply, I make use of a personal CD player mains adaptor.

The instructions cover 5 sides of A4. They start off with advice on precautions against electrostatic discharge, with a suggestion on how to improvise. Also included are a table to identify parts, parts list, PCB layout and circuit diagrams.

The written instructions fill around a page, and are straightforward if followed with care and frequent reference to the tables and layout diagram.

The PygmyHC11 software is supplied in two zip files, one of which includes a readme with instructions for minor updates to source files. A modified Pygmy metacompiler compiles the target image from several text files. Tools are provided to output a listing of compiled words, with default PygmyHC11 words explained in a PDF document.

Two Motorola reference books covering full details of the processor instructions and onboard peripherals are recommended, available free of charge from the Motorola web site.

It took me several hours to assemble the controller components. Once complete, the code loaded onto the controller first time without any of the problems mentioned in the User Guide supplied with the software.

I am now ready for a stab at the Clock Challenge.

F11-UK

provides everything needed in a professional-quality low-cost Forth controller board.

Use it in industrial or hobby projects to control a wide range of devices using the well-known multi-tasking Pygmy Forth.

Designed for hosting from a DOS or Windows PC, you can test your application as it runs on the F11-UK board itself. The board was developed by FIG UK members to provide an easy way to explore the world of controlled devices – a niche where Forth excels.

The kit includes both hardware and software and is supported and sold to members at a nominal profit through a private company.

Software

PC-based PygmyHC11 Forth compiler running under DOS produces code for Motorola HC11 micro-controller.

Code is downloaded via standard serial link from the PC to the FLASH memory (or RAM) on the F11-UK single board computer (SBC).

No dongle or programming adaptor of any kind is required.

Forth running on the SBC is interactive which makes debugging and testing much easier.

Multitasking and Assembly included.

The serial link can be disconnected to enable the SBC to function as a stand alone unit.

All source code provided - 78 pages or so (unlike many commercial systems).

Around 30 pages of additional documentation is supplied including a full glossary of the 300 or so Forth words in the system.

Email mailing list for discussion and limited support.

Hardware:

Processor: Motorola HC11 version E1 – 8 MHz (2 MHz E-Clock).

Memory: 32k x 8 FLASH
32k x 8 battery backed SRAM
512 x 8 EEPROM onboard HC11.

I/O: 20 lines plus 2 interrupts (IRQ and XIRQ).

Analogue in: up to 8 lines using onboard 8-bit A/D.

Serial: 1) RS232, UART onboard HC11
2) Motorola SPI bus onboard HC11.

Expansion: Via HC11 SPI serial bus using 2 or more of 20 available lines.

Timer system:

Inputs: 3 x 16-bit capture channels
Outputs: 4 x 16-bit compare channels.

PCB size: 103 x 100 mm.

Price to FIG UK members: £47.0 plus postage and packing (£2 UK, £4 overseas) plus \$25.0 (US Dollars) for registration of 80x86 Pygmy Forth with the author Frank Sergeant.

Delivery: ex-stock.
More information: jeremy.fowell@btinternet.com and 0121 440 1809

XML and Forth

Les Kendall

Les Kendall makes his first appearance in Forthwrite on the hot topic of XML, a convenient way to store and exchange information which is rapidly replacing ad-hoc data formats everywhere.

XML (extensible Markup Language) is one of the most important technologies that have emerged in recent times. It is supported by the likes of Microsoft¹, Oracle, IBM, Sun and just about any company that can write software.

XML is a cousin of HTML and is widely used by Web applications. So what has this to do with Forth? Well, XML is far more than just a web technology. Where HTML is about the presentation of the data, XML describes the data itself, and so it is an important data interchange medium.

Example

Let's take an example when you use Forth in an embedded application, say a data logging application. You store the data on-board and at some stage you need to get the data to a PC for analysis, so you dump your data over the serial line and it comes in like this:

```
164558067211652622714116586897230
```

Although all the information is there, it's unreadable. We're going to have to write a custom app. to analyse it or at least get some other code to translate it.

If we sent the *same* data as XML, it would look like this:

```
<?xml version="1.0"?>
<DataLog>
  <Log>
    <LogNum> 1</LogNum>
    <Time> 1645</Time>
    <Speed> 580</Speed>
    <Temp> 67.2</Temp>
```

¹ To see some XML, save a modern Excel spreadsheet as an HTML file and examine it with a text editor. Interleaved with the HTML data needed to present the data on a web page, is sufficient XML data to re-construct the original spreadsheet, including its formulas, when loaded back into Excel – Ed.

```

    <Valve> OPEN</Valve>
  </Log>
<Log>
  <LogNum> 2</LogNum>
  <Time> 1652</Time>
  <Speed> 622</Speed>
  <Temp> 71.4</Temp>
  <Valve> OPEN</Valve>
</Log>
<Log>
  <LogNum> 3</LogNum>
  <Time> 1658</Time>
  <Speed> 689</Speed>
  <Temp> 72.3</Temp>
  <Valve>CLOSED</Valve>
</Log>
</DataLog>

```

Wow! Even if you have not seen or heard of XML before, you can follow this. Can you say at what time the highest speed occurred? It's so readable, it's almost self-explanatory.

Tags

XML uses 'Tags', which are the names in the "<" ">" brackets. We have a *Start Tag* and an *End Tag*, and in between the two we have data. The correct terminology is that Tags are *markup* and the data between them is *content*.

The end tag must have the same name as the Start Tag but is preceded by a '/'. For example, to send a surname of 'Jones' you may use

```
<surname>Jones</surname>
```

You can choose any tag name you want and the more descriptive the better. If you think that sounds like making things up as we go along, well, that's the power and flexibility of XML; all you need to do is to keep to the syntax.

Markup may be hierarchical. Between a Start Tag and End Tag, we can add other tags. The rule here is that you must nest Start and End Tags in the correct order. We can use

```

  <tag1><tag2></tag2></tag1>
but
  <tag1><tag2></tag1></tag2>

```

is illegal. As long as you obey that simple rule, you can nest as deeply as you want.

Empty tags indicate that there is no data. For example, if there is no email address, you can write empty tags as

```
<email></email> or as <email/>.
```

Say you wanted to describe a business contact list in XML. Think of a *contact list* that will have a number of *contacts* and that each contact has *Name*, *email* and *company*; each Company has a *Name*, *Address* and *Phone* number. We call the layout the *XML Schema*. An example with just 2 records may look like this:

```
<?xml version="1.0"?>
<contactlist>
  <contact>
    <name>          Les Kendall</name>
    <email>         Les@cybsoft.co.uk</email>
    <company>
      <name>          LKSoft</name>
      <address>       Birmingham</address>
      <phone>        123 4567</phone>
    </company>
  </contact>
  <contact>
    <name>          Mall Salesman</name>
    <email>         sales@cyberforth.com</email>
    <company>
      <name>          Cyberforth Ltd</name>
      <address>       UK</address>
      <phone>        987 6543</phone>
    </company>
  </contact>
</contactlist>
```

Notice how contact name and company name both have the tag `<name>` but due to the inherent nature of XML, its use is both legal and obvious to the user.

'Well Formed' XML

There are only a few more things you need to know before you can use XML.

You *must* include the first line:

```
<?xml version="1.0"?>
```

That tells the parser that this is intended as a XML document and what version of XML we are using.

The above examples are “*well formed*” because they follow XML syntax and will pass through a XML Parser (the name of a XML reader) without the parser objecting.

Valid XML

If you want the parser to validate the data then the structure of the file can be defined at the start of the XML. This is called the DTD (Document Type Declaration) and it describes what markup is to be expected in the XML file.

```
<?XML version = "1.0" ? >
<!DOCTYPE ContactFile [
  <!ELEMENT ContactFile (Contact)* >
  <!ELEMENT Contact (Name, Phone, Fax) >
  <!ELEMENT Name (#PCDATA) >
  <!ELEMENT Phone (#PCDATA) >
  <!ELEMENT Fax (#PCDATA) >
]>
<ContactFile>
  <Contact>
    <Name> Fred</Name>
    <Phone> 123</Phone>
    <Fax> 456</Fax>
  </Contact>
  <Contact>
    <Name> Jack</Name>
    <Phone> 111</Phone>
    <Fax> 222</Fax>
  </Contact>
</ContactFile >
```

Here, the DTD at the start of the document explains to the parser what to expect. This is how it works:

<?XML version = "1.0" ? >	Required.
<!DOCTYPE ContactFile [The document is a ContactFile schema. It contains a number of Elements.
<!ELEMENT ContactFile (Contact)* >	The * shows there can be more than one set of contacts.
<!ELEMENT Contact (Name, Phone, Fax) >	Each contact is made up of Name, Phone and Fax.
<!ELEMENT Name (#PCDATA) >	The base elements are Name,

```
<!ELEMENT Phone (#PCDATA) >           Phone and Fax.
<!ELEMENT Fax   (#PCDATA) >
```

There are a few more bits and pieces to a DTD but this shows the fundamentals. Armed with this information, the parser can check the integrity of the document.

Entities

Some of you may have noticed that we could really mess up the syntax if the content (data) contained a '<' or a '>'. The angle brackets are reserved characters in XML, together with '&', '"' and "'".

These reserved characters are called *Entities* and have to be translated as shown in this table:

Description	Character	Entity
less than	<	<
greater than	>	>
apostrophe	'	'
quotation mark	"	"
ampersand	&	&

So if we have a company called "FRED'S DISKS & DRIVES" then it would be output as

```
<company>FRED&apos;S DISKS &amp; DRIVES</company>
```

The parser will decode these entities when the file is read.

XML Parsers

To read the XML back you need a *XML Parser*. There are many free parsers around, and all the large software companies have at least one. Many software applications are starting to have their own built-in parsers and, as XML gains pace, these will become standard and you won't even realise that all data will be in XML format.

One of these applications is MS Internet Explorer v5 or later. As a quick check to test if your XML is well formed, just try to open it in IE5. If it displays without error then you're in business.

Editor's Note

I asked on the newsgroup for an XML parser in Forth. No-one has done one yet, so if you are keen to make a contribution to the Forth community, here is a golden opportunity.

Forth code to generate XML

Just to show how easy it is to output data in XML, here is a little Forth program that will produce an output on the screen of the data we used in the initial example.

```
\ XML Test output file (ignores entities)
\ Change INTSIZE for the number of bytes in an integer on your Forth
\ e.g. 4 for WinForth, 2 for F-PC
4 CONSTANT INTSIZE

\ Assume our datalogger stored data; we'll just stuff some in here...
Create iTime 1645 , 1652 , 1658 ,
Create iSpeed 580 , 622 , 689 ,
Create iTemp 672 , 714 , 723 ,
Create bValve 1 C, 1 C, 0 C,
3 CONSTANT NUMRECS

\ output a number without any spaces
: WriteNumber ( n -- )
  S>D <# #S #> TYPE
;

\ output a number as a temperature as xx.x
: WriteTemperature ( n -- )
  10 /MOD WriteNumber ASCII . EMIT WriteNumber
;

: WriteXML ( -- )
  CR
  ." <Datalog>" CR
  NUMRECS 0 DO
    ." <Log>" CR
    ." <LogNum>" I 1+ WriteNumber ." </LogNum>" CR
    ." <Time>" iTime I INTSIZE * + @ WriteNumber
      ." </Time>" CR
    ." <Speed>" iSpeed I INTSIZE * + @ WriteNumber
      ." </Speed>" CR
    ." <Temp>" iTemp I INTSIZE * + @ WriteTemperature
      ." </Temp>" CR
    ." <Valve>"
    bValve I + c@ 0= IF ." OPEN" ELSE ." CLOSED" THEN
      ." </Valve>" CR
    ." </Log>" CR
  LOOP
  ." </Datalog>" CR
;

;
```

Note: You don't need to format the XML with tabs and spaces; I've done it in the examples to make the XML more readable.

Advantages of XML

- XML is non-proprietary.
- XML is fairly simple and very readable.
- XML is platform-independent and you can use XML to pass data back and forth between Windows, Linux, Unix, Sun, IBM and virtually anything else.
- XML is already supported by every major software vendor.
- XML is easy to search.
- XML is easy to sort.
- XML can be placed directly on Web Pages.
- XML is plain ASCII and will support almost any transmission method.
- XML is already the basis of many other exciting new developments, e.g. SOAP (Simple OA Protocol)
- Within a year all spreadsheets, databases and word processors will be able to read and write XML.

You can find out more about XML at <http://www.xml.org> .
For free XML tools, see list at <http://www.garshol.priv.no/download/xmltools>.

Conclusion

Microsoft has already delayed Windows.NET so that they can ensure that XML is the basis of all their applications. Other manufacturers are planning the same. XML has enormous and limitless potential for everyone in the IT business, no matter what sector or allegiance.

XML gives Forth an equal footing with all other languages and systems. Forth can now (or soon will) interchange data with any other application. Think about it. Straight from a Forth app. into a Microsoft Excel spreadsheet for analysis, or straight into an Oracle database for storing. Forth is no longer handicapped by not having native database drivers, object brokers or a wad of file conversion routines.

So next time you write a Forth app, either for the PC or running on a micro board, tell your customer that you can output data directly to almost any device they can think of; because with XML you are capable of doing just that.

Les Kendall is a freelance software developer who programs in Visual Basic, C++ and Forth. He was introduced to Forth by a friend in 1978 and “has been a follower ever since”. He delivered two embedded Forth applications during 2000, which I hope we will hear about in detail in due course.

euroFORTH 2000 – The Report

Howerd Oakford

The November conference extended beyond Europe bringing in visitors from USA and Russia. We are indebted to Howerd who not only presented an interesting paper, but also provided us with this entertaining report.

The material presented is very varied and of high standard. As well as the impressive industrial applications, I am always fascinated by papers that extend the capability of Forth in just a handful of new words.

The “web-enabled light switch” is not as frivolous as it sounds. As the full paper reveals, this is a major security application which has successfully moved from Unix to Forth to reduce cost and size.

Twenty-three people, fourteen papers, four workshops, one swimming pool, and a partridge in cranberry sauce.... Actually it was turkey, but I claim poetic licence!

The venue this year was the De Vere Mottram Hall hotel, near Manchester, UK. Of course we weren't only there for the excellent food, attentive service and sauna/health spa - there

were also the papers, presenting the latest developments in and around Forth :



Chris Bailey

“Inter Boundary Scheduling of Stack Operands”
described a new technique for eliminating local variables by keeping them on the stack.

Anton Ertl

“CONST-DOES>” put forward an extension to ANS to allow compilers to optimise words which would otherwise be created by CONSTANT ... DOES> .

Chris Bailey
and
Mike Weeks

“Single and Multiple Issue ILP speedup for stack-based code” - possible compiler extensions to allow optimisation of Instruction-Level Parallelism in stack-based processors.

Stephen Pelc

“Inside the VFX optimiser” describes MPE's important new optimising Forth compiler, which can speed up Forth programs by three to five times compared to conventional compilers. This usually gives a performance within 20% of hand-crafted assembler code. If anyone says that Forth is slow, point them at VFX!

“The DOCCEN documentation generator” is another MPE development : a set of words are defined which allow program documentation to be put *in the source code*, and extracted as finished documentation. This ties the source and documentation together in a delightfully elegant way.

Eugenia Lyakina
and
Sergei Sidorov

“Bilingual Programming System DSSP + Forth” is about a Russian Forth-like language - DSSP - and how it can be re-united with Forth by a new bilingual system. I am always fascinated by software from Eastern European countries because it is not subject to the same commercial pressures as in the West, and gives us a different perspective.

Bill Stoddart

“A Virtual Machine Architecture for Constraint Based Programming” gives us two new words ?CONTINUE and CHOICE, and three code fields which allow a program to run backwards.

Each Forth word has three ways in which it can be executed (a variable holds the offset to one of its three code fields). In Normal mode each Forth word behaves normally, in Conservative mode any data which is required to undo the effect of the word is put on a special History Stack, and in Reverse mode each word undoes its own actions. The History Stack is required by words such as + to save one of its inputs, otherwise its action could not be undone.

Apart from the sheer "wow" of a program that can run forward then "change into reverse gear" and run backwards, this is a real implementation in the academic field of reversible computing.

Federico de
Celballos

“A Machine Forth Specification based on the PSC1000 capabilities” explores the implementation of Forth given two additional hardware registers (address and index) and corresponding instructions of the PSC1000. Conventional Forths are tailored to the processors that they run on, but now that Chuck Moore has found that these features are useful in hardware it is obviously important to see if they give any advantages in software too.

This demonstrates very well the dynamic, fluid nature of Forth - improvements in software influence hardware, which affects the software again - there are no artificially imposed hierarchical layers, just real ones, such as what can economically be put on silicon and what cannot.

Nick Nelson

“Industrial control languages: Forth vs IEC61131” is about using Forth in the real world. Flavour of the month for Programmable Logic Controllers is IEC61131, which defines three graphical and two text-based languages for programming them - unfortunately none of them is Forth. So how do you convince your customer to let you program his application in half the time (and at half the cost) in Forth, instead of following the herd?

Two answers spring to mind : get Forth included in IEC61131 (how do you do this?), or develop the application in Forth then translate it into one of the approved languages.

Michael
Milendorf

“The Firmware Architecting Process” describes Sun Microsystems, Inc. procedures for controlling the generation of ANS Forth code used in their OpenBoot firmware.

It's nice to see a major company using Forth, and it's always good to remind people in the non-Forth world that pressing "StopA" on a Sun workstation keyboard gives you a Forth "ok" prompt ...

Jeremy Moller,
Desmond Ward,
George Redpath

“Porting Complex Unix Applications to a deeply embedded Forth Environment” - CEM Systems Ltd have crammed an HTTP server, SMPT client, BOOTP, TELNET and a TCP/IP "stack" into a Web-enabled security package the size of a light-switch.

Forth still excels where there is limited hardware, and here is a classic example of how Forth can be used to tame complexity. Bravo!

A.C.Wheatley,
N.J.Nelson

“A hydraulic servo controller using the IX1 microprocessor”

- Micros Electronics Ltd gave us another example of Forth being used in the real world, this time to control "stretch forming" machines used to produce car body parts.

Nick Nelson's practical demonstration of the complex multi-axis control required to prevent the machines from self-destructing will long be remembered as one of the highlights of the conference!

N.J.Nelson

“Grid and histogram controls for Windows implemented in Forth”

described how even Windows can be made subservient to programmers by means of MPE's proForth. Micros Electronics Ltd must be commended for offering the source code for the Windows controls and samples illustrating their use publicly available.

Howerd
Oakford

“The Point to Point Protocol in Forth” describes my own battle against the complexity of one of the corner stones of the WWW.

Every time I check my email the PC uses PPP to communicate with the Internet Service Provider. The interactive nature of Forth has allowed me to write a set of Forth words that describe the data structures and program states that make up PPP. The result is a program that that can open a PPP connection and transfer ICMP, IP and small UDP packets, which fits into about 18K bytes of code space and 100 bytes of RAM, about half the size of comparable "IP Stacks".

I plan to publish the source code once I can use the program to read my email...

All of the above papers are available on-line² in PDF format from :

<http://dec.bournemouth.ac.uk/forth/euro/ef00/>

I hope I have given an accurate interpretation of each paper presented at this year's EuroForth conference - my apologies if I have got it wrong, missed something out or not given *your* paper enough coverage, but space and time is limited...

What may not be apparent from the papers is the rest of the event - the conversations about Forth, Perl, C, piano tuning, life in Russia versus life in England; the opportunity to meet people from many countries and diverse backgrounds to share ideas; the demonstrations of real hardware and software...

So my thanks to all involved, especially Malcolm Bugler, Radka Jiraskova and Peter Knaggs for organising the conference, John Jones and Cathy Silvester for the after-dinner speech and every one who attended EuroForth 2000.

² The papers have also been printed and bound suitable for photocopying and are available from the FIG UK library, as are all published conference proceedings. See the web site for a full list.

See you all next time!



Alan J M Wenham

01932 786440

101745.3615@compuserve.com

Vierte Dimension 4/00

Alan Wenham

Alan provides a look at the latest issue of the German FIG magazine. To borrow a copy or to arrange for a translation of an individual article, please call Alan.

More on FIG USA

A report by the Forth Gesellschaft Directorate on the situation in FIG USA together with responses and opinions from Elizabeth Rather and Chris Jakeman. I am sure that FIG UK members are fully aware of these matters from comp.lang.forth

Media Night

Martin Bitter

mbitter@bigfoot.de

It is no longer possible to imagine daily life without computers. In schools it is a failing not to have computer-literate teachers. On the other hand, there are retired specialists from the electronics and computer programming fields, who one would be glad to employ to help. Various firms and social establishments have organised activities to bring the importance of computers to children and adults.

Based on the initiative by Fred Behringer of Forth Gesellschaft, Martin Bitter presented his school programme "Lego-Robot programming in Forth" (see also <http://www.forth-ev.de>) and this is his report of what took place.

Their eyes sparkled at the Forth Robots

Fred Behringer

behringe@mathematik.tu-muenchen.de

This is Fred's report of the "Media Night" on 8th July 2000 in Vaterstetten in Bavaria.

Letters

From Russia (concerning an article by Fred Behringer), from several places in connection with the general article "Forth for Fun", and from Germany about the short note acknowledging Forth by Michael Kalus.

Solution of the riddle of the ciphered cipher

Fred sets out the letters from Friederich Prinz, Michael Major, and Martin Bitter with suggestions for solutions to the riddle posed by Fred in VD 2/2000 and gives his own explanation.

Solution of the four problem riddle

Fred shows the letters from Michael Major and Herbert Fink with suggestions toward the solution of Fred's riddle from VD 2/2000 and sets out his own ideas.

"Avisé" AVR Virtual Stack Engine

Wolfgang Schemmert
interface@cinetix.de

Description of an experimental simple prototype board with a Micro-Controller type AT90LS4433. Reduced Forth, in 4 KByte, modified syntax, oriented toward BASIC. Pinouts, circuit diagram, and board layout are given.

Digital signatures and embedded systems

Ulrich Hoffman
uho@xlerb.de

How can one ensure a reliable digital signature, for instance in banking systems. A lecture that the author gave at the annual meeting in April 2000. From the Forth point of view, is the interesting processing of long numbers.

Other Groups

Fred Behringer summarises Forthwrite 107 and 108 and notes items from FigLeaf (the Dutch FIG magazine).

From FIG Silicon Valley

Henry Vinerts

Henry Vinerts, the Forth correspondent from USA, reports on those parts of FIG International that have, up to now, overcome all their difficulties.

This is an extensive report on several aspects of FIG and the people therein. I hope that most of the content has been available to FIG UK members via other, English-speaking, routes.

Umlauts, a solution suggestion

Herbert Fink

herb.fink@t-online.de

This concerns the typing and printing of the German double-dot (eg. ä, ë, ü) which modifies the sound and sometimes the meaning of the vowels a, o, and u. There are problems with the original character sets for the IBM PC and also room for confusion.

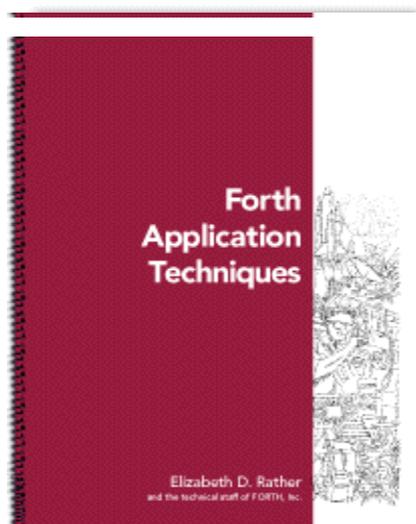
These vowels are normally represented in English as 'æ', 'œ' and 'ue' but these can be ambiguous in such English words as 'pœt', and 'æroplane'. Fred Behringer had asked for proposals from the German Forth community. Herbert Fink has responded, albeit not in Forth but in Visual Basic. His coding has significant interest.

Annual Conference

Forth Gesellschaft (German FIG) will be holding their annual conference at Wesel (near the Dutch border) on April 27-29th.

This is a Fri-Sun event with presented papers - including one from Chris Jakeman. Details will be available soon (in English too) on their web site <http://www.forth-ev.de> and non-members are especially welcome.

As a special event at the conference, Martin Bitter is planning a practical contest with Lego robots programmed in Forth.



Forth Application Techniques

Robert Ives

A new Forth book is always a major event. Robert takes a look at the introductory, tutorial book from Forth Inc. based on their Forth Application Techniques course manual and refined over many years.

Over 144 pages, this book includes substantial new material, drawings and diagrams, as well as many problems and exercises directed toward helping the student learn how to use Forth effectively to develop applications.

The new book is available from Forth Inc at the introductory price of \$20 + delivery and also, along with every major and recent Forth publication, can be borrowed (by members only) free of charge from the FIG UK library. You will find a complete index on our web site.

First a bit of background:

I started using Forth on a Commodore 64 with the help of Starting Forth by Leo Brodie. Compared to the built-in Basic, I was amazed what could be achieved. Since then I have dabbled with Forth, using it for quick solutions to small problems but restricted by the wants of the companies I have worked for.

Now on to the book!

It is always a treat to see a new book published on Forth, especially aimed at those new to the language.

The book itself comes spiral bound so that it can lay flat on the desk unlike a number of other publications (or perhaps I should say tomes).

The first section of the book gives a brief outline on how the publication came about. It also says:

"If you are not using this book in a class, the first element is missing. Realistically, this means you'll miss a lot of background details and explanations of why things work the way they do. In that case, we strongly recommend that you supplement this book with a more detailed text on Forth."

More about this later.

The coverage of topics is sufficient to get the beginner programming in Forth and, with the help of additional material, to allow them to produce some useful code.

It covers topics under each section in a concise manner which I did appreciate. The only problem I had with this approach is that it could feel a little disjointed. A couple of short programs as an illustration of combining methods at the end of the book would have helped, or even as a download from their website.

As for working through the examples, a complete beginner to Forth should consider using the free, trial version of SwiftForth as it reduces the frustration that can come about through an example/exercise not working (the answers to exercises are available from the web site).

As I worked through the text I came to realise that the interactivity of a class with this material is important. As mentioned at the beginning of this review, I am a “dabbler” when it comes to Forth but I can still remember the difficulty in first grasping the language.

I hope that does not sound too negative, but I feel any potential purchaser should be aware of this (we all learn in different ways). If you intend to use Forth as a programming language of choice or you wish to gain a good technical understanding then, yes, I would recommend it (I intend to purchase a copy). However, if you are coming to Forth as an interested observer then I would suggest exploring the language with several of the on-line tutorials (see the FIG UK website - <http://forth.org.uk>).

Finally I would suggest to Forth Inc. that they consider producing some form of CD-based material to go with this publication (videos of lectures, trial version SwiftForth, etc.). This would of course add to the expense, but perhaps they could consider having it as a separately available purchase.

INTRODUCTION

A Brief History of Forth
The Philosophy of Forth

SIMPLE FORTH PRIMITIVES

STRUCTURED PROGRAMMING

DATA STORAGE

STRINGS AND CHARACTERS

NUMBER CONVERSION

VECTORED EXECUTION

ADVANCED CONCEPTS

Dictionary
Search Orders and Word Lists
Text Interpreter
Compiler
Exceptions and Error Handling
Defining Words

MULTITASKING

Basic Principles
The SwiftOS Multitasker
Multitasking under Windows
Resource Sharing
Application Design Issues

STYLE RECOMMENDATIONS

Forth Editing Standards
Open Firmware Coding Style
Naming Conventions

Letters

The Magazine Team are always pleased to get feedback and encouragement. Here we have feedback on recent articles from Ed Hersom, Fred Behringer and Garry Lancaster. On the newsgroup, John Jones described Forth in original terms approved of by member Paul Bennett and reproduced here with permission.

Ed Hersom

From: Ed Hersom [ehersom@cix.compulink.co.uk]
Sent: 20 November 2000 18:03
Subject: Permutations

Hi Chris,

I was pleased to hear my old algorithm has been put to good use.

Since 1991 I have ported Pygmy Forth into my HP 200LX palmtop. I can now engage in "armchair" computing (as befits my age - I'm well into my 80th year). If it is of any interest, I give below my Pygmy version of the Permutation algorithm. The changes are mainly cosmetic. These are:

```
R> --> PUSH
>R --> POP
```

and

```
R@ --> I (normally the loop index, but not in this case)
```

```
: PERM RECURSIVE ( this makes PERM legal in the body of the definition)
  DUP 1 =
  IF PUSH CR .S POP
  ELSE DUP FOR PUSH I 1- PERM ROLL POP NEXT
  THEN ;
```

It is interesting to note that although the index in the FOR/NEXT loop is negative-going, the sequence of permutations is the same in this algorithm as in the previous version.

Fred poses a little challenge in his letter.

Fred Behringer

From: Fred Behringer [behringe@sunstatistik1.mathematik.tu-muenchen.de]
Sent: 10 December 2000 08:55

Hi Chris,

Here is another version of EXCHANGE which uses but 8 words instead of the 10 words employed by Wil Baden in Forthwrite's issue 109

(which was

```
: EXCHANGE ( addr1 addr2 -- ) DUP @ >R OVER @ SWAP ! R> SWAP ! ;  
Ed.):
```

```
: EXCHANGE ( addr1 addr2 -- ) 2DUP @ SWAP @ ROT ! SWAP ! ;
```

The above version of EXCHANGE has 16 bytes in its parameter field (data field). The following code definition has but 10 bytes in its code body.

```
CODE EXCHANGE ( addr1 addr2 -- )  
  BX POP      DI POP  
  0 [BX] AX MOV  0 [DI] CX MOV  
  CX 0 [BX] MOV  AX 0 [DI] MOV  
  NEXT END-CODE
```

The above is in (16 bit) ZF or Turbo Forth assembler. I wonder whether either version, high-level or low-level, can be reduced further?

Garry isn't using Forth in his new job (yet) but Forth helped him get the job and the Forth approach proves valuable. He also adds a useful comment on "BLT is not a Sandwich".

Garry Lancaster

From: Garry Lancaster [glancaster@uk2.net]
Sent: 28 November 2000 23:31

Dear Chris,

Thanks very much for ForthWrite 109 - an interesting and informative read, as always.

Dave Pochin's article "BLT is not a sandwich" particularly caught my eye, since I started working for a leading 3D graphics card producer recently, writing 2D drivers for Win9x. (I got the job mostly thanks to my hobbyist work in assembly & Forth on the Z88 & Spectrum!)

One of my first tasks on joining the company was to check and correct microcode programs implementing the 256 "blitmode" functions referred to in the article. I was right at home at once, since the usual way of describing these (aside from the few odd names such as "blackness", "patcopy" etc) is in good old Reverse Polish notation. Hence, PATPAINT is usually referred to as DPSnoo (~S|P|D in C-speak). The more complex of these (eg PSDPSonox) are hideously complex in C notation, yet remain simple to understand and implement when written in RP.

Another interesting point is that "blitmode" is a bit of a misnomer, since "dwROP" (signifying a DWORD-sized Raster OPERATION value) is in fact specified for almost all Windows graphics functions, and not just blits; for example, lines, polygons and text. The fact that Win32Forth calls this value "blitmode" indicates that it might not be available for these other functions for some artificial reason.

Thanks again for a great issue.

Finally, a note reproduced from comp.lang.forth with permission:

**John Jones and
Paul Bennett**

Subject: Re: Why choose Forth?
Date: Tue, 19 Dec 00 21:38:19 GMT
From: peb@amleth.demon.co.uk ("Paul E. Bennett")
Organization: HIDECS Consultancy
Newsgroups: comp.lang.forth

In article <uSfU\$PAPFOP6EwMA@pathfind.demon.co.uk>
johnj@pathfind.demon.co.uk "John Jones" writes:

- > My main theme on Forth in a business environment is
- > to view it as a machine not a language. A machine where you can extend
- > the instruction set to suit the commercial goals you are working to
- > fulfill. I feel (intuitively and difficult to support rationally) that
- > computer science took a turn into a blind alley when it followed the
- > language paradigm. It forced a human abstraction onto a simple machine
- > that subsequently demanded convoluted language definitions of its own in
- > order to fulfill its goals. In doing so it separated man from machine,
- > to mans detriment. Forth keeps the man-machine linkage quite clear,
- > while also introducing a level of abstraction that builds on the
- > physical reality; i.e. Forth's virtual machine structure is always
- > mapped onto the physical machine. In other 'languages' this, in my
- > opinion essential, mapping is hidden or even non existant.

Hey!!!!!! Hold that thought a moment. Well done John. I think you have just distilled a good way of expressing some feelings and vague thoughts I have been having. It certainly highlights some important differences between Forth and other languages.

--

Paul E. Bennett<email://peb@amleth.demon.co.uk>
Forth based HIDECS Consultancy<http://www.amleth.demon.co.uk/>
Mob: +44 (0)7811-639972NOW AVAILABLE:- HIDECS COURSE.....
Tel: +44 (0)1235-814586 see http://www.feabhas.com for details.
Going Forth Safely EBA. www.electric-boat-association.org.uk..

FIG UK Committee



Chair	Chris Hainsworth , Microplex Ltd., 5a Riverfield Road, STAINES TW18 2EE 01784 457565 chris.hainsworth@dial.pipex.com
Secretary	Doug Neale , 58 Woodland Way, MORDEN SM4 4DS 020 8542 2747 dneale@w58wmorden.demon.co.uk
Editor	Chris Jakeman , 50 Grimshaw Road, PETERBOROUGH PE1 4ET 01733 753489 cjakeman@bigfoot.com
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 jennybrien@bmallard.swinternet.co.uk
Librarian	Sylvia Hainsworth , Microplex Ltd., 5a Riverfield Road, STAINES 01784 457565 sylvia.hainsworth@dial.pipex.com
Ordinary Member	Jeremy Fowell , 11 Hitches Lane, EDGEBASTON B15 2LS 0121 440 1809 jeremy.fowell@btinternet.com

Membership enquiries, renewals and changes of address to Doug.
Technical enquiries and anything for publication to Chris J..
Borrowing requests for books, magazines and proceedings to Sylvia.

FIG UK Web Site

For indexes to Forthwrite, the FIG UK Library and much more, see <http://forth.org.uk>

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 and Chris Jakeman, 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://forth.org.uk> . 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.