

of course[®]

VOLUME 2, NUMBER 2

AUTUMN 2002

Our biggest feature yet

**Regular
Expression
Explosion!**

The update on

Tcl/Tk

The latest news on

**Perl • PHP • MySQL
Java • Expect • Ruby**

Home Grown



Welcome to this, the fifth edition of our "Of Course" newsletter.

It's now two years since we opened our own training centre in Melksham, Wiltshire and we've gone from strength to strength in the courses that we offer and our training facilities.

New this summer, we've added a lounge/diner to our student facilities so that you can arrive early to read the paper and watch the morning news, or stay on socially after the course to get to know your fellow trainees better. This expansion also means that we can be more flexible with our lunch arrangements, especially on larger courses. (For those anxious for an update, the redesigned customer loo is a work of art!)

When we opened in Melksham, we offered just Perl and Java courses, and since then we've added Tcl/Tk, PHP and MySQL. In this newsletter, we announce training in Ruby too. If you have a group of four or more trainees requiring the same course, we can run a private course for them, either at our training centre, using our facilities, or at your offices.

Everyone's talking about regular expressions and we're always being asked questions about them. This popularity gave us the idea that you'd want to know more about them as well, so we've featured them heavily in this issue. You'll find a series of articles and texts on the technical aspects starting on page 8, and some practical examples showing you regular expressions written to match everything from a date to a UK car registration. You'll find this table of examples on page 14.

There's also a feature bringing Tcl/Tk up to date. Tcl is a powerful language which is enjoying a long life due to its excellent initial design back in the '80s. This feature starts on page 6.

And that's in addition to our regular features such as an update on the current releases of the Open Source languages on page 13, and a training course schedule on page 15.

– Graham J. Ellis

Ruby is a gem in our eyes

Ruby is a truly object oriented Open Source programming language. It's designed with the strengths and capabilities of the major scripting languages in mind, and is slowly but surely gaining ground in the USA.

We like Ruby. It has an easy syntax, a rich set of libraries, the power of objects, regular expressions, special variables, hashes, arrays that grow automatically, and much, much more.

Our initial public Ruby course assumes prior programming knowledge, and quickly covers the fundamentals of the language.

Since Ruby is an object oriented language, we concentrate on its implementation of objects straight after covering the fundamental syntax,

and compare and contrast its objects to those in Perl and Java.

Our course concludes with sections covering the uses of Ruby on the Web, using Ruby to handle relational databases through SQL, and Graphic User Interfaces such as Ruby/Tk.

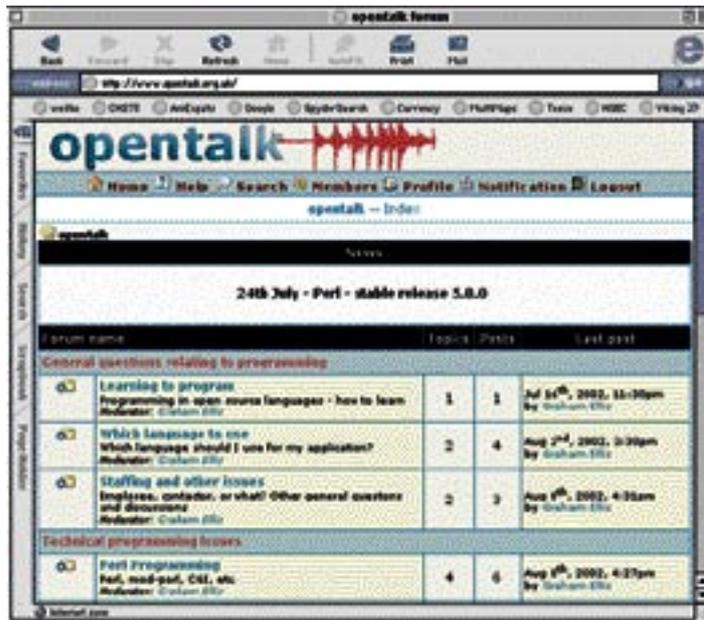
We can also run tailored private courses if you have a group of four or more who require the same training at the same time. Such courses are tailored to meet your needs. For example, we could run a three- or four-day course to teach Ruby to trainees with no prior programming experience.

Duration: 2 days

Venue: Melksham, Wiltshire

Price: £490.00 + VAT

Start Dates: 26 September, 2002
8 December, 2002



Open Source online forum a resource for questions

Need an answer to a programming question? Looked in the books, but not found the answer? Think that there's someone out there who might know, perhaps the person who taught you the language in the first place?

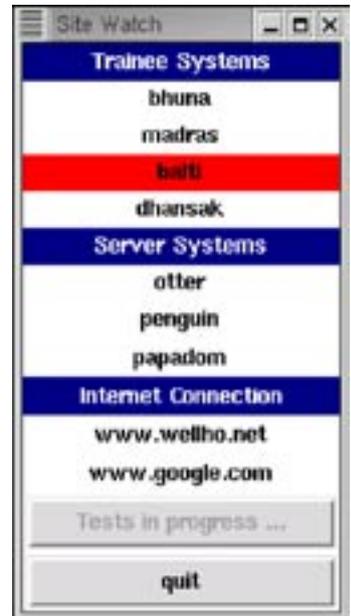
Although we suggest that you first check in books, in your course notes, and with colleagues before you cast wider, you have always been welcome to email your tutor, who will get back to you with 48 hours.

Now there is another service available which will give you even broader insights to your solutions and suggestions for your programming efficiency. The

whole Open Source community is able to participate in an open forum in which they can ask questions about open source languages such as Perl, Tcl and PHP, and have participants come back with a response. We regularly monitor it ourselves and you may find one of our tutors providing you with an answer or suggestion.

Just like Open Source, it's freely available and open discussions are welcome. "OpenTalk" can be found at <http://www.opentalk.org.uk>.

Topics such as Java, MySQL and more general programming issues are discussed as well.



"Site Watch" displays the status of systems on a network even while it continues to probe.

Tk's use of idle time

If you're using Perl/Tk or Tcl/Tk, you'll be writing a program to create a number of widgets, then display them under the control of a geometry manager such as pack grid or place. There's much more to a geometry manager than you might think, as the overall layout of your widgets might only become defined as you add the last widget to your window.

How, then, does Tk avoid your window resizing and rearranging in an irritating dance as it's initially displayed? Answer: Tk buffers up all your widgets and geometry placement requests, and only updates the display when it goes idle. In other words, your window isn't displayed at all until your program has completed the definition of all the widgets and gone into an idle state to await an event. Subsequently, your display will only be updated when the event queue is empty and your Tcl or Perl program is idle again.

If you've ever used Tk interactively (i.e. from the wish prompt), you have seen that your display is updated after each widget is passed to a geometry manager. Quite simply, this is because wish has emptied its event queue and is awaiting a new event, such as you typing in the next command.

When running a substantial piece of code behind a Tk GUI, you may wish to have your display updated even if you are not yet awaiting an event by forcing any display changes to be acted on by running the `update idletasks` command (Tcl/Tk) or calling the `update` method (Perl/Tk).

Second edition updated, but lacks some coverage

by Graham Ellis, Well House Consultants Ltd.

The second edition of *Mastering Regular Expressions*, published in July 2002, updates and extends the coverage of this fascinating subject. As well as covering the general

principles of regular expressions well and in depth, there are major sections on the use of regular expressions in Perl, Java, and in .NET.

If you're a PHP, MySQL, Ruby or Tcl programmer, you'll find the general chapters of this book useful, but you'll look in vain for a lot of detailed information about the particular language. You will find that the comparison tables highlight some of the differences though.

No doubt there will be a third edition of this book when Perl 6 comes along. I was reading a 25-page article on regular expression changes in Perl 6 a week or two back ...

By Jeffrey E. F. Friedl
2nd Edition July 2002
O'Reilly 0-596-00289-0

Create Portable DB Apps

PHP is a great language for building dynamic web pages, but its database access functions are not standardised. Each type of database uses a different function library.

ADODB (Active Data Objects Data Base) is a free, downloadable library enabling you to switch between MySQL, PostgreSQL, Oracle, MS SQL 7, Access, ADO, Sybase, DB2,

generic ODBC and others without having to rewrite your code.

Methods are provided for date handling, string concatenation and string quoting characters.

A metatype system is built in so that CHAR, TEXT and STRING are equivalent in different databases.

ADODB is available at:
<http://php.weblogs.com/adodb>

Flexing the PHP muscles

"PHP was written for use embedded with a web page, and that's how it's always used".



So you might think, and you might be right, but you're also wrong!

Our statement is correct in that PHP was written for embedded use, but the use of PHP has now grown beyond that use. Why is that?

PHP has an exceptionally strong and diverse library of functions, often better than those provided by other, more conventional, languages. And PHP programs can be run from the command line if you wish. Simply pass PHP a `-q` option to tell it not to generate HTTP headers, open the keyboard using `fopen`, and you're away!

Applications that you might like to consider writing directly in PHP include graphic manipulation applications and also email handlers because of the wide range of MIME, URL and email functions available. The image that illustrates this article, for instance, was produced by a PHP program that's just 14 lines long, from two .jpg images off our digital camera.



© 2002 GREG PULLEN

Like our business philosophy since our humble beginnings, the Well House hasn't changed much since this picture (above) was taken in the early 1900s. The only difference has been our growth. For continuity, we continued to keep the name even after we moved to larger accommodation.

We've been asked frequently why we are called "Well House Consultants".

Quite simply, Well House Consultants was founded in "The Well House", in Easterton near Devizes, Wiltshire, in 1995.

We started life as a tiny consultancy, offering our tutor through a reseller to bring to life subjects such as Solaris System Administration and C programming. We're no longer as tiny, nor are we still located in The Well House.

Our philosophy then (as now) was to provide a service excellence above and beyond the call of duty, clearly a philosophy that our customers appreciate as we've grown and continue to grow into the future.



Save time by cutting the loops

Many computer experts will tell you that languages like Perl, Tcl and PHP are interpretive, and so programs written in those languages will run slower than programs written in compiled languages such as C. Take a C program and translate it, line-by-line, into the equivalent statements in a language such as Perl. You will note things will slow down for sure. But the

more modern languages have some tricks up their sleeves.

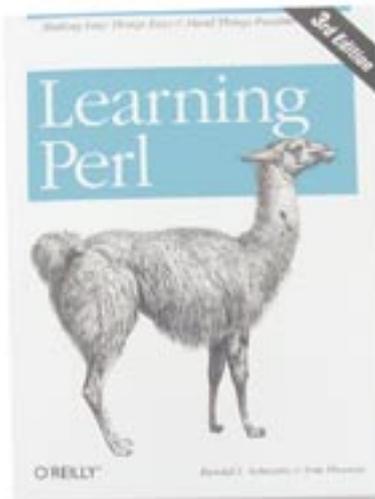
If you use the power of functions that operate on lists (or arrays in PHP and Tcl) as a whole, you'll cut out the loops from your program, slashing the work that's to be done at run time. The effects can be dramatic; code that runs as fast as a C program, but is developed in a fraction of the time.

From beginner to programmer, the books that helped flatten the learning curve

by John Moylan, now technical webmaster with the Consumers' Association

My first contact with Well House Consultants came nearly three years ago when I chanced upon their website while looking for programming tuition in the UK. At this point I had only enthusiasm but no proper knowledge of programming, and my alternatives were between Java and Perl to accompany my web design experience.

My correspondence with Graham continued by email and phone, and Perl emerged as the natural route; I signed up for the five day 'Learning to Program in Perl'. The course flattened my learning curve, and stepped me up a level where I could better understand the books I was reading. I'm still learning but that never stops, the road as they say, just goes on. I have built quite a personal library of Perl books, (though by no means close to the Well House library) and found various Internet resources over the past few years. However, there is no substitute for 'doing' and courses I attended at Well House Consultants were probably the most time-efficient method of learning as they got you 'doing'.



Books I find helpful:

Learning Perl

O'Reilly & Associates, Inc. 0-596-00130-0

This is the standard Perl tutorial that all others are judged by, though I found that it assumed a fair amount of programming knowledge (mainly C) and was Unix specific. I found too many occasions where the text read along the lines of "this works as it does in C". If you don't know C, that's no help! However, I understand the recently published third edition has recognised this slant and rectified it, and is now OS independent. There is also a version of the second edition specifically for Windows.

Programming Perl

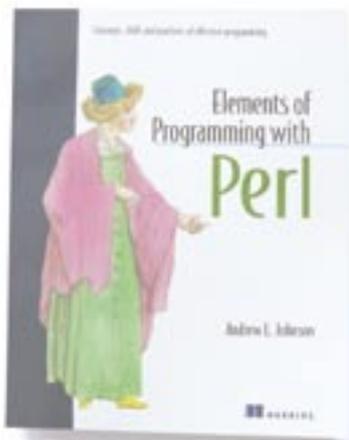
O'Reilly & Associates, Inc. 0-596-00027-8

Quite simply the Perl bible, you won't use this much to begin with but after using Perl for a while this becomes an essential reference of the language.

Perl Cookbook

O'Reilly & Associates, Inc. 1-56592-243-3

The first edition of *Programming Perl* had a chapter of Perl recipes, and this soon became the most 'thumbed' chapter in the book, giving real-world examples of Perl usage. However the second edition omitted this chapter and the screams could be heard far and wide, but O'Reilly had an ace up their sleeve in the guise of *The Perl Cookbook* – 791 pages of problems, solutions and examples of every aspect, for the Perl newbie to the seasoned hacker, it's an essential read.



Elements of Programming with Perl

Manning 1-884777-80-5

Ah, we get to my favourite. Think of this book as a beginner programmer's version of *Learning Perl*. It starts with absolutely no assumptions of your programming level and steps you through some nice easy tasks that get 'chained together', and midway through the book you realise that you have the basic elements of programming Perl (hence the name, I suppose). By the end, you will have a good understanding of Perl and what it can achieve, but more importantly, and if you do the exercises, you find yourself stepping away from being a beginner. The author is online to answer any queries regarding the exercises, prompt with his replies and clear and concise with his feedback. Given that this is included in the price of the book and done in his own time, I must add that Andrew Johnson has done a great service to

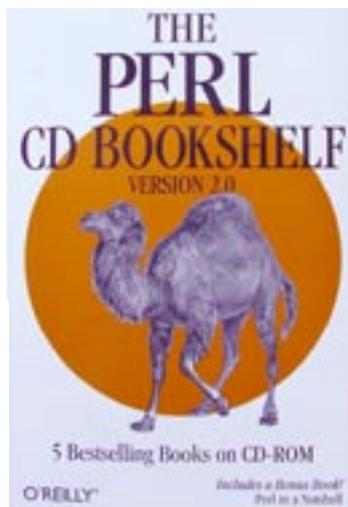
the Perl community. For the new/non-programmer I can't recommend this book highly enough.

Perl in a Nutshell

O'Reilly & Associates, Inc. 1-56592-286-7

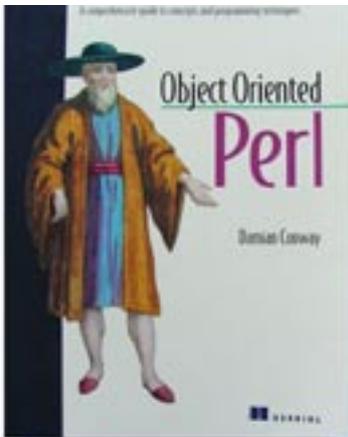
Think of this as a concise version of *Programming Perl*, a quick reference to functions when you don't need every nuance of documentation. I don't think it's essential, but see the 'Perl CD Bookshelf' below.

Perl CD bookshelf



O'Reilly & Associates, Inc. 1-56592-462-2

Perl in a Nutshell (hardcopy) with a fully searchable CD containing *Perl in a Nutshell*, *Learning Perl*, *Learning Perl on Win32 Systems*, *Programming Perl*, *Advanced Perl Programming* and *Perl Cookbook*. The latest (third) version of this omits *Learning Perl* and *Learning Perl on Win32 Systems* but includes *Perl for System*



Administration (bit of an error I think, O'Reilly). Fantastic value though!

Object Oriented Perl

Manning 1-884777-79-1

<general assumptions>
"Perl is read only"
"Perl is not for serious programming"

"You can't do Object Oriented programming in Perl"

"Perl is for quick hacks, you can't write a large serious project in Perl"

</general assumptions>
Not really for beginners but if you get the chance, just read the first three chapters of this book and laugh at the statements above.

Writing CGI Applications with Perl

Addison Wesley 0-201-71014-5

Like most people now, we get to use/learn Perl because of the internet and its role as the de facto programming language for CGI, and this is a superb CGI specific book. It assumes only a basic knowledge of Perl but also explains line-by-line what the code's doing. In the step-by-step tutorial it helps you build some pretty useful tools. Well worth the admission fee.

CGI Programming with Perl

O'Reilly & Associates, Inc. 1-56592-419-3

Now in its second edition, this book teaches CGI programming using Perl, written in a completely different style to *Writing CGI applications with Perl*. I found these CGI-specific books complemented each other nicely. There's a nice section on Perl templating solutions.

Perl Debugged

Addison Wesley 0-201-70054-9

Not a reference or tutorial, this is more of a 'defensive coding' book. Aimed at the beginning-to-intermediate developer, this teaches a coding style to minimise bugs in the first instance. I found it to be quite useful as there are few books of this type around and this is the only one I know of that uses Perl. It also teaches debugging techniques but emphasises that the coding style is a better defence.

Programming the Perl DBI: Database Programming in Perl

O'Reilly & Associates, Inc. 1-56592-699-4

Perl and database programming were made for each other, and this book gives extensive coverage of Perl's 'DBI' module. Unfortunately, the man pages for this module are just as extensive as this book; save your money and stick with them.

Perl and CGI for the WWW

Peachpit Press 0-201-35358-X

Perl for Dummies

IDG Books Worldwide 0-7645-0776-1

I never purchased the above books but got to have a good read in my local bookstore; thank heavens I did! These seem to be the obvious choice for the beginner; don't believe it. Not the most inspiring and possibly do as much harm as good.

So, Article in a Nutshell, what would I recommend to a beginner? A good book, such as *Elements of Programming Perl* and a good tutor.

Yet Another Bulletin Board – Open Source at its best

Don't go writing code if someone has already done the job for you! For each open source language, there's a whole range of additional modules out there for programmers to use, and a whole range of applications out there for people looking for complete solutions.

When we were asked to look into a new bulletin board for a web site that's run by a contact of ours, we had a look around at Open Source software already written, and came up with a couple of products, recommended by friends who subscribe to such boards. After all, why write another board? We keep our skills sharp by actually

Forum name	Topics	Posts
YaBB Service		
YaBB 1.0.0	4	176
YaBB 1.0.1	127	4412
YaBB 1.0.2	112	301
YaBB English Support		
YaBB 1.0.0	1,202	8641
YaBB 1.0.1	473	1634
YaBB 1.0.2	221	781
YaBB 1.0.3	125	4611
YaBB 1.0.4	294	1234
YaBB 1.0.5	281	1233

working with the latest uses of the technology; much of that process is taking a look and adapting what has already been done by someone else.

The two pieces of software that were suggested were phpBB, which is written in PHP, and YaBB, which is written in Perl. For no better reason other than the fact that my subscriber friend preferred the look and feel of YaBB, that's the one we decided to use and adapt.

YaBB proved equally easy to install for testing on our web site, and maintaining on the user's web site. Knowledge of Perl was a big help in the initial configuration, but it shouldn't be vital. Anyone can cautiously follow the easy-to-understand, well commented instructions and change a few vital parameters in a configuration file. Mind you, once the software is up and running, the fun begins! You won't need to reconfigure any HTML code, aside from adding your own text; however, if you want your site to have a different look and feel, even novice web site designers will find changing

colours and adding images to be easy.

Before a board can go live, the administrator needs to choose form names, and then categories within each forum, and to make decisions about how the board will operate. For example, can a guest visitor post? Will you be appointing moderators for each of the subject area, or will you handle all this on your own? You should decide on lists of banned words and write an acceptable user policy (AUP).

And then there's the long task of encouraging people to come to your board and post on it ... no easy matter in the early days, when your board would make the Arctic look crowded!

Perhaps your board is concerned with a highly specialised subject, and in time it becomes a valuable resource. How then can you turn your content into web pages that search engines will be able to index, and have archives that the world will be able to browse? Within the YaBB source code there wasn't an obvious way, at first glance. Another contact (it's amazing how you end up talking with other board operators) told us that she had been making enquiries and come up with a blank. But then we looked at the structure of YaBB and realised just how well constructed it is.

Firstly, the data formats that YaBB saves its data in are clear and easy to understand, and secondly the YaBB software itself is well structured into a series of separate Perl files. By writing an extra piece of Perl, which is really no more than a utility (less than 150 lines!), we are now able to archive all the threads to HTML pages. How's this achieved in such a short program? By having a utility load in some of the YaBB subroutines that the authors have kindly structured off into separate files. Not only did this save a lot of work, but it means that smileys, or emoticons, can be accurately mirrored on the archive pages, and deal with all the special formatting such as links, bold text, quoting etc., secure in the knowledge that it's working as YaBB intended. It also means that when a YaBB update comes along, we should simply be able to call their new file of subroutines and support for new facilities will come with no fuss at all.

In Open Source spirit, we're sharing our little utility with the contact who got us into YaBB in the first place, and with the other contact who had drawn a blank when searching for this facility. The whole experience has been a glorious, sunny, refreshing reminder of the power of Open Source.

The following is a summary of some of YaBB 1 Gold - SP1's features. Note that this is not a complete list.

Every Page

- Notification of instant messages, greeting, date/time
- Navigation menu bar (text, image, or combination format)
- Forum navigation links or text (index/category/board/message... etc.)

Forum Index

- Stats: newest member, total members, total posts, most recent post
- Graphical alert of new messages in each board
- Date/member of latest posts in each board
- Name of board moderators
- Ability to mark every post in the forum "as read"
- Options not accessible to guests are "hidden" from them
- Flat text or javascript fading news, with YaBBC enabled

Topic Index

- Graphical alert each new or updated topic
- Date/member of latest posts in each topic
- Names of the board moderators
- 'Jump to' box for instant jump to a different board or category
- Ability to mark every post in the board "as read"
- Options not accessible to guests are "hidden" from them

Topic Display

- Number of reply on each post
- Author information: email, # posts, membergroup, IM links and userpic
- Option to view profile of authors
- Ability to quote messages or reply
- Date of each post listed
- User/Moderator ability to delete or modify posts
- Option to display a "printer-friendly" copy of the page
- Option to send a friend the topic

Post Page

- Easy to use editor-style interface
- Option to receive notification of replies
- Option to disable smileys
- Full 'editor' bar for inserting smileys and YaBBC tags
- Input field autojump javascript
- Full preview ability
- Option to receive topic "notification"
- Option to disable smileys

Other Pages

- Recent posts
- XX number of posts by a certain member
- Forum post search
- Full members list

Other Features

- Full administration and moderating options
- Ability to change settings and template on the fly
- Full YaBBC code in posts, with error-checking galore
- Instant messaging system for member-to-member communication
- Ability to customise the layout colours and CSS via one template
- Private boards which only certain membergroups can access
- Login length options
- Delete registered members
- Full profile options, including signature, settings, and personal information
- Email all registered users
- 'Maintenance Mode' for major board administration
- Online help for users and admins
- Moderator/Administrator ability to move topics
- Simple registration form and option to generate random password
- Post icons to designate the message type
- Admin option to disable the YaBBC code

A downloadable version of YaBB 1 Gold - SP1 can be found at:

<http://www.yabbforum.com/>

Bringing Tcl up to date



● *“The Tcl scripting language grew out of my work on design tools for integrated circuits at the University of California at Berkeley in the early 1980’s. My students and I had written several interactive tools for IC design, such as Magic and Crystal...”*

– John Ousterhout
“The Birth of Tcl”
<http://www.tcl.tk/doc/tclHistory.html>



Tcl (pronounced *Tickle*)
An embeddable command language

[Incr-Tcl] (pronounced *Inker Tickle*)
Originally a saleable product, but now a part of Open Source Tcl distributions

[Incr-Tk] (pronounced *Inker Tee Kay*)
Create composite new widgets using standard Tk building blocks

The first version of Tcl was in use as long ago as 1988, with Tk trailing along the following year. Being almost 15 years old, Tcl is one of the older languages we teach. But there’s still a consistent request for courses in Tcl. Why is that?

The answer is obvious if you use it. To this day there’s nothing better as an Open Source embeddable command language, and it remains the choice of many organisations which need to provide tailoring tools within their products.

For a language to survive as long as Tcl has is a great credit to its designer and author, John Ousterhout, and the countless others who have contributed along the way, and continue to do so to this day. You’ll still see the same fundamentals in Tcl 8.4 as were there in Tcl 1.0, but there’s much more too, building on John’s initial decision that “the language must be extensible”.

These days, the majority of our trainees aren’t embedding Tcl into applications themselves, but rather they’re using software, typically commercial software, that used Tcl as its command language or language of fine tuning. Books such as Ousterhout’s “Tcl and the Tk Toolkit”, which included 150 pages on writing Tcl applications in C, and on Tk’s C interfaces, are now specialist reading.

What are some of the more recent additions to Tcl that has allowed it to remain the embedded language of choice for so many?

For Larger Applications

Libraries group sets of Tcl procedures so that they can be shared between multiple applications, and the package facility supports facilities such as version numbering, autoload and conditional loading. Namespaces provide new scopes for global variables and procedures, so that code within a package does not have data or procedure names that conflict with the application in which they are used.

[Incr-Tcl] extends the language to support Object Oriented (OO) programming in much the same way that C++ extends C. [Incr-Tcl] includes the facilities that

Tracing a variable in Tcl

Do you want to trace a variable in a Tcl program? Be told when it's changed, and perhaps when it's read too? The trace command lets you do this, but you do need to provide a proc to be performed whenever the variable is written or

EXAMPLE (using the trace command)

```
The code
#!/usr/bin/tcl

puts -nonewline "Give me a big number: "
flush stdout

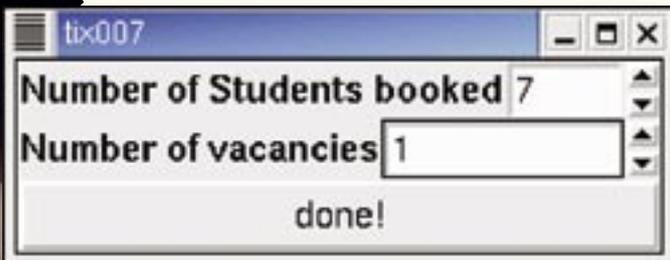
gets stdin value

while {$value > 100} {
    set value [expr $value / 2 - 2]
}

puts "final value is $value"
```

In operation

```
$ ./vt
Give me a big number: 1023
final value is 60
$
```



Example of a Mega-widget

you would expect from a thorough OO language, such as protection levels and multiple inheritance, as well as the basics of constructors and destructors, methods, object variables, encapsulation, polymorphism and more.

Extending Tk

The Tk GUI has always provided the ability to write a user interface in tens of lines of code that typically takes hundreds of lines in X Windows and C. And it's portable too.

[Incr-Tk] lets you create composite new widgets using standard Tk building blocks. These "Mega-widgets", as they're known, look and feel just like ordinary Tk widgets.

Classes of Mega-widgets such as the Tix Mega-widgets, iwidgets and Bwidgets have sprung up to provide File selection boxes, scrollable lists, adjustable frames, tree support, and much more. These are all now available Open Source for free download.

Other Extensions

The TclX extension adds a diverse set of extras onto the slim Tcl language, allowing it to practically move out of its original embedded mould, and become a practical alternative for a wider range of applications. Examples of the extension in TclX include extra string, list and maths functions, the ability to run other processing via exec and fork operations, and file I/O commands. There's also a number of network communications functions, though they are not so sophisticated as those in the "scotty" extension.

Within the base Tcl itself, a byte code compiler was added at Tcl 8.0, improving script performance by a factor of two to 20 times and, at Tcl 8.1, a new regular expression engine was introduced more than doubling the number of special sequences for seeing if a string matches a pattern. There have been numerous other additions as well.

Tcl and Tcl/Tk are truly remarkable. They've stood the test of time well, have developed yet remain modern tools to this day, and are set for a bright future.

OK, is that right?

Let's add some traces on the variable value, and watch that variable each time it's read or changed:

EXAMPLE (adding some traces)

```
The code
#!/usr/bin/tcl

# tracing a variable - example

proc watch {varname key op} {
    if {$key != ""} {
        set varname ${varname}($key)
    }
    upvar $varname var
    puts "$varname is $var (operation $op)"
}

# -----
trace variable value w watch
trace variable value r watch
# -----

puts -nonewline "Give me a big number: "
flush stdout

gets stdin value

while {$value > 100} {
    set value [expr $value / 2 - 2]
}

puts "final value is $value"
```

In operation

```
$ ./vt
Give me a big number: 1023
value is 1023 (operation w)
value is 1023 (operation r)
value is 1023 (operation r)
value is 509 (operation w)
value is 509 (operation r)
value is 509 (operation r)
value is 252 (operation w)
value is 252 (operation r)
value is 252 (operation r)
value is 124 (operation w)
value is 124 (operation r)
value is 124 (operation r)
value is 60 (operation w)
value is 60 (operation r)
value is 60 (operation r)
final value is 60
$
```

Good, we can see how it has worked now.

The `trace variable` command takes three parameters:

- the name of the variable to trace
 - the operation (`r` for read, `w` for write); and
 - the name of the command to run when the read or write operation happens.
- You can turn a trace back off using the `trace vdelete` command.

The proc itself takes three parameters:

- the variable name
 - the name of the element (if it's an array element that you're tracing); and
 - a variable telling you if it's a read or a write (you can also use `u` for unset)
- In our example, we've added code to allow you trace an array element if there's an element name, and we've also used `upvar` to get at the value of the variable – probably a requirement that you'll always have in a trace procedure.



The human mind is remarkably good at recognising a pattern. If I drop an email address into the middle of a sentence, you would find it very easy indeed to read the sentence and tell me that:

- it contains an email address
- the email address is "graham@wellho.co.uk"
- the domain is "wellho.co.uk"
- the user's name is "Graham"
- this appears to be an email address of someone in Great Britain ("co.uk")

Historically, finding and extracting a string such as an email address automatically within a program was quite a task. Perhaps you might:

- search the string forward for an @ character
- search backward for a white space, looking for the start of the email address
- then search forward for the next white space, to find the end of the email address

Sound complicated? Yes, it was, and our example is even simplified too! Enter the regular expression.

A regular expression is a pattern that you use to describe what you're looking for within a piece of text. The pattern can be very specific, such as "I want to find the sequence of characters g-r-a-h-a-m". Or they can be much less specific, such as "I want to find a sequence of non-space characters with an @ in the middle". Regular expression handlers can provide a woolly or fuzzy match for these. The really clever bit is that you specify the match pattern, by defining the fuzzyness yourself, through the use of regular expressions.

Writing a regular expression

When you write a regular expression into your program, there are two stages that are gone through before the match is completed. The first stage is a

"compile" phase, in which the regular expression is analysed and converted to some internal structure, and the second is the use of this internal structure to perform the actual match. For efficiency's sake, some languages give you an element of control over the compiling then matching stages.

In Java you create a pattern object, using the static compile method of the Pattern class - for example:

```
Pattern email = Pattern.compile("^\\S+@\\S+$");  
and you can then use this pattern to match against as many different strings as  
you like, using the matcher method:  
Matcher fit = email.matcher(args[i]);
```

When you have done the match, you then examine the results:

```
if (fit.matches()) {  
    If you're using Perl, Perl itself will compile a regular expression for you the first  
    time it encounters it, then retain the internal structure for you so that it doesn't  
    have to recompile every time you match to the same expression. However, if  
    you include a variable within your regular expression things can slow right down  
    as Perl has to recompile the regular expression just in case the variable has  
    changed every time it comes to it. If you know that the contents of a variable are  
    not going to change during one particular run of your program, you can "help"  
    Perl by telling it not to recompile every time using the "o" modifier, for example,  
    you might write:
```

```
if ($p =~ /\S+@\S+\. $top/o) ...  
in a program that searches for all the email address from one top-level domain  
each time you run the program. In the latest releases of Perl, you can also  
compile a regular expression into a scalar variable using the "q" operator.
```

Here's a Perl program that takes in a series of email addresses, and tells you which ones have a specified top level domain:

```
#!/usr/bin/perl  
$top = qr/\S+@\S+\. $ARGV[0]/;  
shift;  
foreach $em (@ARGV) {  
    $em =~ $top and print "$em is SELECTED\  
n";  
}
```

And if we run that:

```
$ qrdemo uk graham@wellho.net lisa@wellho.co.uk  
other@ec.ac.uk gred@aol.com  
lisa@wellho.co.uk is SELECTED  
other@ec.ac.uk is SELECTED  
$
```

continued on next page



Regular Expressions

Regular expressions are used to define patterns or templates against which you can match a string of characters, to test whether or not that string conforms to the rules specified by the pattern.

```
#!/usr/bin/perl  
  
print "Please enter text: ";  
  
if (<STDIN> =~ /(\S+)@(\S+(\.\S+))/) {  
    print "name $1 ... domain $2 .... top level $3\n";  
} else {  
    print "no email address there\n";  
}
```



THE SPA B&B

CONIGRE FARM

THE OLD MANOR

Local accommodation around Melksham

People attend courses at our Melksham training centre from far and wide, and there's a good range of well priced accommodation available nearby. Whilst we try to ensure the accuracy of these details, we might

not be informed of any changes, so please verify this information with the establishment when you book your own accommodation. Pricing shown is for single occupancy; many offer family and double occupancy facilities.

There are also self catering, caravan and camping facilities nearby, as well as other hotels, guest houses and B&Bs. Melksham Tourist Information Centre (01225 707 424) can provide you with up-to-date details.

1 The Spa B&B 402, The Spa, Melksham SN12 6QL H. Pavey P: 01225 707 984	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	0.1	0.7	no	no	no
£20.00-25.00 single, one room en-suite, includes breakfast Large comfortable listed Georgian home. Quiet position with open views. Ample parking.					
4 The Old Manor 48 Spa Road, Melksham SN12 7NY L. Bailey P&F: 01225 793 803 E: theoldmanor@yahoo.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	0.5	0.3	no	no	no
£25.00-35.00 single, one room en-suite, includes breakfast Set in 3/4 acre, delightful gardens; large English breakfasts a speciality; lovely decor.					
7 The Kings Arms Hotel 20 Market Place, Melksham SN12 6EX P: 01225 707 272	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	0.8	0.0	no	yes	no
£35.00 single, all rooms en-suite + £6.95 full/£4.95 continental bkfst Recently completely renovated; located in the heart of town.					
15 Conigre Farm Semington Road, Melksham SN12 6BZ D. Edwards P: 01225 702 229 F: 01225 707 392 W: www.cfhotel.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	0.9	0.1	yes	yes	yes
£43.00-52.00 single, all rooms en-suite, SkyTV, includes breakfast Family-run hotel and restaurant offers finest foods, quality accommodation and friendly environment.					
38 Longhope Guest House 9 Beanacre Road, Melksham SN12 8AG D. Hyatt P&F: 01225 706 737 E: longhope@aol.com	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	2.0	0.9	no	no	no
£28.00 single, all rooms en-suite, includes breakfast Large Victorian house set in its own grounds; warm and friendly, well decorated.					
47 Beechfield House Beanacre, Melksham SN12 7PU P: 01225 703 700 F: 01225 790 118 E: csm@beechfieldhouse.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	2.7	1.6	yes	yes	yes
£80.00-90.00 single, all rooms en-suite, includes breakfast Comfortable country house, 21 en-suite bedrooms, four-poster beds available, heated outdoor swimming pool.					
45 The Shaw Country Hotel Bath Road, Shaw, Melksham SN12 8EF J. Lewis P: 01225 702 836 F: 01225 790 275	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	2.8	1.7	yes	yes	yes
£46.00 single, all rooms en-suite, includes breakfast A 400-year-old farm house, supremely restored. Mulberry Restaurant has a growing reputation for good food.					
51 Woolley Grange Hotel Woolley Green, Bradford-on-Avon N. Chapman P: 01225 864 705 F: 01225 864 059 E: info@woolleygrange.com	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	6.1	Bradford-on-A approx. 1 mile	yes	yes	yes
from £90.00 single, all rooms en-suite, includes breakfast Jacobean manor house in open countryside with outdoor heated pool, croquet, tennis, badminton.					
48 Travel Inn, Cepen Park (#267) W. Chippenham Way (M4, J17) Chippenham SN14 6UZ P: 01249 462 096 F: 01249 461 387 W: www.travelinn.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	8.6	Chippenham approx. 1 mile	no	yes	yes
£41.95 single, all rooms en-suite + £6.00 full/£4.00 continental bkfst 79 rooms with Brewsters and petrol station nearby.					
52 Travelodge (#813) Moto Services, A36 / A350 Bypass Warminster BA12 7RU P: 01985 219 539 F: 01985 214 380 W: www.travelodge.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
	training	town			
	12.1	Warminster approx. 1 mile	no	yes	yes
£49.95 single, all rooms en-suite + breakfast from £4.45 31 rooms with Burger King, Little Chef and petrol station nearby.					

Regular Expressions

continued from previous page

Regular expressions are adaptable to a wide variety of programming languages.

Regular expressions in Java

In the early days of Java, a regular expression engine was written by Johnathan Locke and donated to the Apache Software Foundation.

You can still download it from <http://jakarta.apache.org/regexp/index.html>.

As of the release 1.4 of Java in Spring 2002, there's a standard package called *java.util.regex* that's shipped with the Java Runtime Environment (JRE).

These new Java regular expressions are described as being very "Perl-like", though some of the methods provided for matching (such as `lookingAt`) will surprise you when they seem to assume a `begin` anchor by default.

Regular expressions in Perl

Regular expressions are fundamental to Perl.

The regular expression handler is rich in facilities, many being accessed through terse sequences that make Perl regular expressions powerful, short, and hard for the novice to understand.

In recent releases, such as Perl 5.8, enhancements have been made to give even better facilities, and to support some of the more verbose syntaxes in use in other regular expression handlers. In Perl 6, there will be major changes to regular expressions to enable them for an even bigger future.

Regular expressions in Tcl

Tcl supports both regular expressions and a "glob style" of pattern matching, such as you would use in file name matching in operating system commands.

Up to and including release 8.0 of Tcl, the regular expressions handler offered only the more fundamental facilities, but as from Tcl 8.1, advanced regular expressions were added, providing further capabilities and cleaner syntax for commonly used groups.

Since Tcl is an interpretive language that makes its own use of the `[` and `]` characters, for instance, Tcl regular expressions tend to be grouped in curly braces, `{` and `}`, or be loaded with backslash protections that can make them look quite daunting to the newcomer.

continued on next page

SERVICES

- Banks
 35 Barclays
 10 HSBC
 21 Lloyds TSB
 11 NatWest

Emergency

- 2 Hospital

General

- 19 Post Office
 32 Public Library
 22 Tourist information

LEISURE

- 18 The Blue Pool
 3 Christie Miller Sports Centre

FOOD & DRINK

Groceries

- 40 Aldi
 25 Iceland
 54 Made in America
 31 Sainsbury
 36 Somerfield

Takeaways

- 20 Acropolis
 14 The Chippy
 30 Chop Suey House
 53 Cod Joe's Caribbean Takeaway
 5 Cornerstone Coffee Shop
 26 Country Cottage Pasties
 37 The Fisheries
 13 Golden Falls Chinese
 34 Kebab House
 39 Lee's Fish and Chips
 27 Melksham Chinese

Restaurants (most do takeaways)

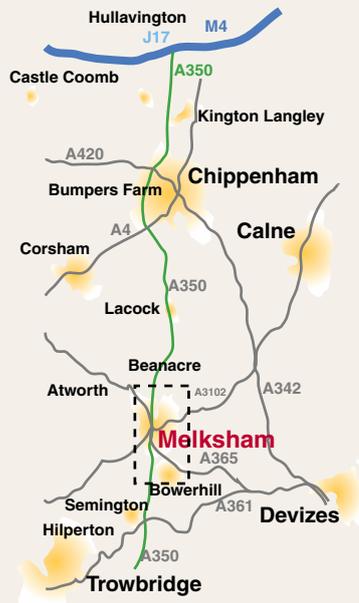
- 8 Big J's Farmer's Cafe
 49 Brewsters
 42 cafe inside Leeke's
 39 Lee's Chinese
 43 Little Chef
 41 McDonald's
 23 Melksham Tandoori
 46 Mulberry Restaurant
 12 Refa Balti House
 40 West End Inn

Sampling of Pubs

- 29 The Bear
 33 Bud's
 50 Foresters Arms
 28 The Grapes
 44 The New Inn
 9 The Parson's Nose
 6 The Tavern

HOTELS / B&Bs

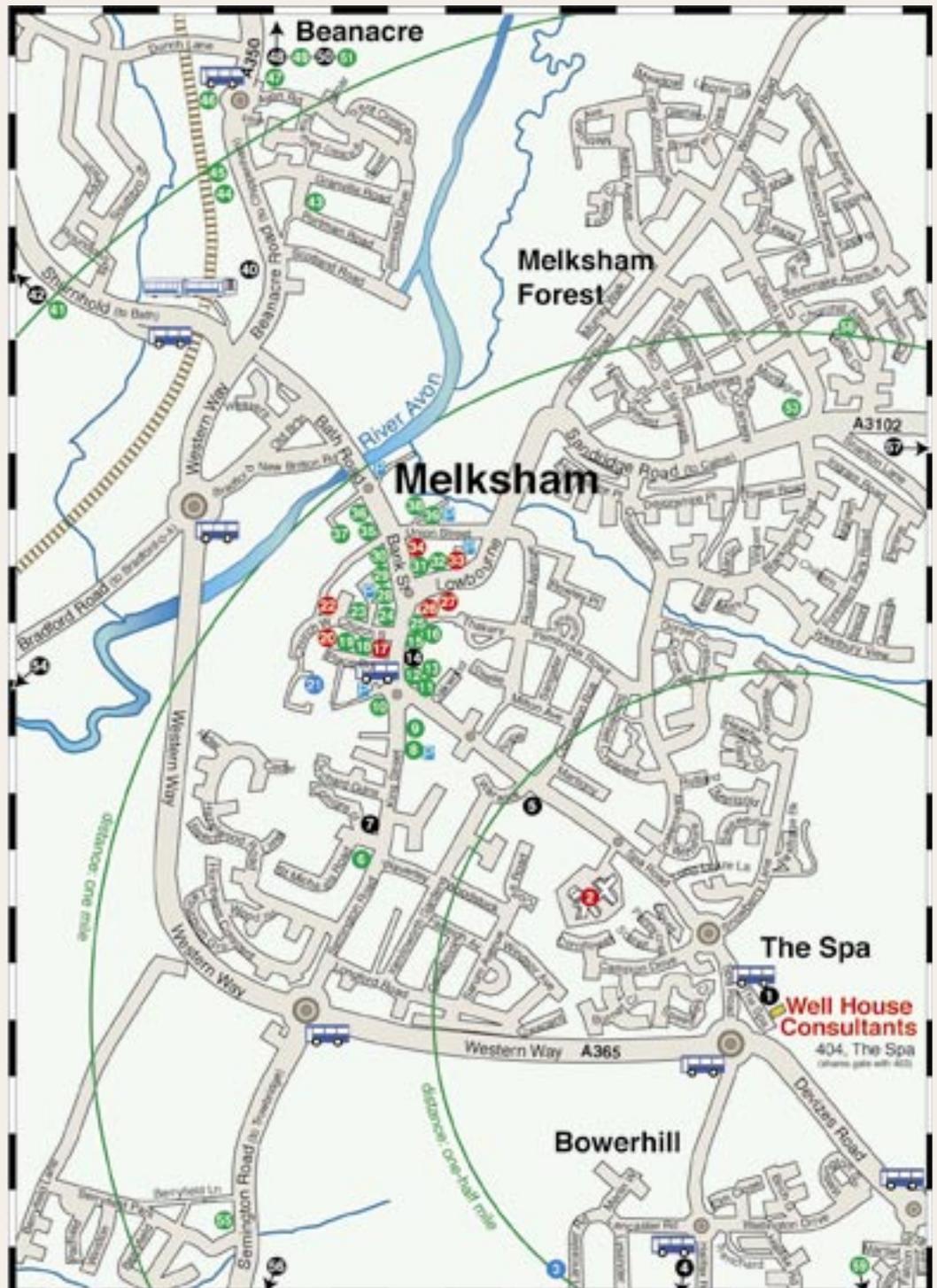
- 47 Beechfield House
 15 Conigre Farm
 7 The Kings Arms Hotel
 38 Longhope Guest House
 4 The Old Manor
 45 The Shaw Country Hotel
 1 The Spa B&B
 48 Travel Inn
 52 Travelodge
 51 Woolley Grange Hotel



From the city or town to Well House Consultants' door step as the crow flies* ... (in order of distance away)

Bath	10	Leicester	120
Swindon	20	Cambridge	140
Salisbury	20	Sheffield	160
Bristol	25	Manchester	165
Southampton	45	Liverpool	165
Cardiff	50	Dover	175
Oxford	55	Leeds	185
Reading	60	Norwich	200
Portsmouth	70	York	205
Swansea	90	Newcastle	275
Birmingham	90	Glasgow	355
London	105	Edinburgh	355
Plymouth	115		

* These are the mileages we use to determine our expenses for on-site courses.



Although regular expressions are often short, they're written with whole strings of special characters and notations, and newcomers don't find them easy. What can this short program possibly achieve in six lines? Quite a bit, actually!

```
#!/usr/bin/perl
print "Please enter text: ";
if (<STDIN> =~ /(\S+)@(\S+(\.\S+))/) {
    print "name $1 ... domain $2 .... top level $3\n";
} else {
    print "no email address there\n";
}
$ ./emad
Please enter text: This is a line of text without an email address
no email address there
$ ./emad
Please enter text: The address of lisa@wellho.net is in this line
name lisa ... domain wellho.net .... top level .net
$
```

It's able to identify whether or not a string of text contains something that looks like an email address, and if it does, it will tell you the user name, domain, and top-level domain too.

Regular expressions in MySQL

MySQL provides pattern matching in commands, such as `SELECT`, through the `LIKE`, which uses `%` to match any sequence of characters, and `_` to match any one character. It also provides `REGEXP` (and the synonym `RLIKE`) which is a regular expression match that provides a good set of the more fundamental facilities of regular expressions. Note that if you want to explicitly match a special character such as a `*` in a `REGEXP` clause, you need to specify `*` since one of the backslashes is stripped off as the query is parsed before it ever gets to the regular expression handler.

Regular expressions in PHP

PHP includes not one but two regular expression handlers. The `ereg` function (and others with similar names) use POSIX-style regular expressions, with character groups such as `[[:alnum:]]` for an alphanumeric character.

The `preg` functions use Perl-style regular expressions. For example, `\w` is used for an alphanumeric character. Although there's more typing involved, we suggest that newcomers use the POSIX regular expressions for code that they're writing. The resultant regular expressions are much easier to read, and therefore easier to maintain.

Regular expressions in Ruby

Ruby regular expressions are literals written between slashes, with modifiers appended after the second slash. They resemble Perl rather than POSIX-style regular expressions (thus `\w` for an alphanumeric character). Since Ruby is an OO language, `regexp` objects can be constructed and used through a variety of methods such as `=~` and `match`. Methods on a `MatchData` object can be used to retrieve values from a match.

Other uses of regular expressions

Early, but still current, uses of regular expressions include the `grep` (Global Regular Expression Processor) command of Unix, Linux and Mac OS-X operating systems, and its extended cousin `egrep` that supports further special character sequences.

The `awk` pattern scanning and processing language also has its roots in Unix and relies on regular expressions for many of its facilities. Even editors such as `sed` (Stream Editor), `vi` and `emacs` let you search for text that matches a regular expression.

continued on next page

Elements of a regular expression

Regular expressions are implemented within each language, and so the facilities vary somewhat. You'll find most of the following in most regular expression handlers:

Type	Definition	Examples	Usage	
Literals	characters that match exactly	a B X 3 8 = @ \n \t \\ \. \\$	individual character new line and tab backslash, dot and dollar	
	Character Groups	any one character from the group	[abcAB] [a-f0-9A-F] [^0-9]	a b c A or B digit or letter A to F not a digit
		Assertions (anchors)	specific condition must apply	^ \$
Counts			previous element in regular expression to occur a number of times	\b * + ? {2,6}
	Groupings	hold together various sections of a regular expression	() (?:)	group and capture group, no capture
		Alternation	provide an "or" capability	

Regular Expression Modifiers

Do you want your regular expressions to ignore case? In some instances, you'll say "yes, please", but in others it will be "no thanks" or even "hell, no!"

Most regular expression handlers can be passed additional parameters, sometimes known as modifiers, which modify the default behaviour of the regular expression engine. For example, in Perl you can add one or more characters onto the end of your regular expression such as:

```
i    ignore case
s    allow "." to match "\n"
x    allow comments in a regular expression
m    allow ^ and $ to match the start and end of embedded lines
```

In PHP, separate functions are provided (`ereg` and `eregi`) for matches that are, and are not, case-sensitive. In Java, extra flags can be passed into the `compile` method. In most regular expression handlers, additional syntaxes allow modifiers to be selected only for a group.

What is the latest open source software version?

Table and comments were compiled in August, 2002

In this regular feature, we give you the latest news headlines from the Open Source world, and keep you updated on the latest releases of the core software and languages that we cover.

COMMENTS ON Perl

At publication date, Perl 5.8.0 is newly released and is intended for use in production environments. Release 5.6.1 is still available for download and described as the most stable release, but it will only be available from the archive once 5.8.0 has proven itself.

The biggest changes in Perl 5.8.0 are an overhaul of the language's handling of Unicode, and the provision of a new threads module providing authors of Perl programs with the ability to write "parallel" code. Other changes include the deprecation or removal of a number of very old and very obscure features, and a number of platform-specific issues.

If you're happily running Perl 5.6 (or even 5.005), you probably won't need to rush out and upgrade to Perl 5.8. But if you've been waiting for stable improvements in extended character handling, or in writing code that can appear to be doing two things at the same time, it might now be a good time to upgrade.

Since Perl 5.8 has only just been released, there isn't yet a 5.9.0 development version available, and Perl 6 is still somewhere in the future.

COMMENTS ON Tcl/Tk

Tcl/Tk 8.4 is now available as a Beta release.

There's a new `lset` command and enhancements to commands such as `regsub` and `trace`.

On the Tk GUI, there are three new widgets - a `spinbox`, a `panedwindow` and a `labelframe`, plus enhancements to other widgets.

Expect is a mature API that changes little since it already does what it sets out, and does it very well. However, there is a new release available, 5.38.0, dated June 2002.

COMMENTS ON PHP

PHP releases seem to have been coming along thick and fast over the last 6 months! The current stable

Subject	Stable Version	Development/Future	Web Site
Perl	5.6.1 or 5.8.0	5.9.0 and 6.0 (future)	www.perl.com
Tcl/Tk	8.3.4	8.4 (Beta 1)	www.tcl.tk
Expect	5.38.0		expect.nist.gov
PHP	4.2.2	4.3.0	www.php.net
Ruby	1.6.7	1.7.2	www.ruby-lang.org
MySQL	3.23.51	4.0.3	www.mysql.com
Java, standard	1.4.0	1.4.1 (Beta)	java.sun.com
Java, Enterprise	1.3.1	1.4 (Public draft specification)	java.sun.com
C (Gnu)	3.1.1	3.2 (Aug02); 3.3	gcc.gnu.org
Python	2.2.1		www.python.org

Where nothing is shown in the development column, there is presently no test version nor development tree publicly available; it is likely that some development is proceeding outside the public arena.

release is 4.2.2, and was released in response to a vulnerability in releases 4.2.0 and 4.2.1 - please upgrade if you run either of these releases.

If you have PHP scripts that you wrote more than a few months ago, the most obvious change in PHP 4.2.x is that global variables are no longer populated from form elements, cookies or the environment by default. Since global variables were the usual way of getting such information into PHP prior to release 4.1, this means that code you wrote under (say) PHP 4.0.5 is liable to break if you upgrade to 4.2.2. Perhaps this is the cost of using such a dynamic young language. We do applaud the changes, however, as we consider them to be in the longer term interests of PHP.

The 4.3.0 development version of PHP is available for download. It includes a major new release of the "Zend Engine" which is the internal virtual machine that runs your PHP code.

Although there's said to be great improvements in performance, and a much improved object model, we would suggest that users of PHP on live web sites stick with 4.2.2 or even an appropriate earlier release at this present time.

COMMENTS ON Ruby

Ruby is a relatively young language, and the changes in the development thread of Ruby (1.7.x) tend to be in the form of additions and enhancements, and since Ruby is an Object Oriented language, they have a tendency to

take the form of additional classes or methods.

COMMENTS ON MySQL

If you're using MySQL in a production environment, stick with the latest 3.23.xx release; the last two "xx" digits change every couple of months, usually small fixes, as this is intended to be the current stable thread.

Release 4.0.x has moved from Alpha to Beta status since our last newsletter, but should still not be used in a production environment. There are major rewrites of the MySQL internals in release 4, so the testing and validation process is taking some time. There is, however, a list of features being added at each subrelease, which we'll summarise for you in a future issue of the newsletter once there's a production release 4.

COMMENTS ON Java

Since our last newsletter, Java2 Standard Edition release 1.4 has moved from a Beta release to a full official release.

Amongst the important new packages included in J2SE release 1.4 is the `java.util.regex` package, which adds regular expression handling to the core Java distribution. This is such an important facility to many users that we have a feature article on regular expressions elsewhere in this newsletter, and we updated our public Java course to add a complete module on the subject within a few weeks of the release of J2SE 1.4.

Another new package in Java 1.4 is `java.nio`, which provides a complete new API to handle input output based on buffer classes rather than the stream model of `java.io`, which dates from release 1.0 of Java.

`javax.xml.parsers`, `javax.xml.transform`, and subpackages, are also new in Java 1.4, and consolidate the JAXP extension into the main release.

There hasn't been a new major release of the Enterprise extensions since the last newsletter, but Sun have published a draft specification for release 1.4. Since Java is not Open Source, even though it's distributed at no charge, you won't see any "development" releases; after the public specification, the next thing you'll see will perhaps be an alpha or beta release.

Updates and Upgrades at Well House Consultants

Material for courses is reviewed for every single running, and often updated right before the course.

The changes to our course material are updated in line with the latest developments.

STOP PRESS
We have just doubled the speed of our internet connection. You can use our new 1 Mb link direct from your trainee workstation at our centre.

As a result of the upgrade we've just completed a major upgrade to all of our training systems and increased our fleet to 15 laptops per course. Public courses are still limited to eight, but on private courses we can train up to 15.

Dates through December 2003 for our public courses are now listed on our web site, www.wellho.net.

Some favourite regular expressions

We thought you might find these useful. They're all written using Perl-style regular expressions.

Email address

example: graham@wellho.net

```
/ (? : ^ | \s ) [ -a-z0-9_ . ] + @ ( [ -a-z0-9 ] + \. ) + [ a-z ] { 2,6 } ( ? : \s | $ ) / i
```

UK postcode

examples: SN12 6QL and EC1N 2LV

```
/ \b [ a-z ] { 1,2 } \d { 1,2 } [ a-z ] ? \s * \d [ a-z ] { 2 } \b / i
```

American zip code

examples: 90011-1774 and 10047

```
/ \b \d { 5 } ( ? : [ - \s ] \d { 4 } ) ? \b /
```

Date (UK style)

examples: 10/8/2002 and 12 August 2002

```
m ! \b [ 0123 ] ? \d [ - / \s \. ] ( ? : [ 01 ] \d | [ a-z ] { 3, } ) [ - / \s \. ] ( ? : \d { 2 } ) ? \d { 2 } \b ! i
```

Time

example: 21:55 or 9:55 p.m. (if you still use that style)

```
/ \b \d { 1,2 } : \d { 1,2 } ( ? : \s * [ aApP ] \. ? [ mM ] \. ? ) ? \b /
```

URL for a web page

example: http://www.grahamellis.co.uk/index.html

```
m ! https ? : // [ -a-z0-9 \. ] { 4, } ( ? : : \d + ) ? / [ ^ # ? ] + ( ? : # \s + ) ? ! i
```

VISA number

example: 4567 6552 5532 7761

```
/ \b 4 \d { 3 } [ \s - ] ? \d { 4 } [ \s - ] ? \d { 4 } [ \s - ] ? \d { 4 } \b /
```

MasterCard number

example: 5353 6789 0087 6534

```
/ \b 5 [ 1-5 ] \d { 2 } [ \s - ] ? \d { 4 } [ \s - ] ? \d { 4 } [ \s - ] ? \d { 4 } \b /
```

UK phone number

example: 01225 708225

```
/ \b 0 [ - \d \s ] { 10, } \b /
```

UK car registration plate

```
/ \b [ A-Z ] { 2 } ( ? : 51 | 02 | 52 ) [ A-Z ] { 3 } \b /
```

current series example: WN02PNG

```
/ \b [ A-HJ-NP-Y ] \d { 1,3 } [ A-Z ] { 3 } \b /
```

previous series

```
/ \b [ A-Z ] { 3 } \d { 1,3 } [ A-HJ-NP-Y ] \b /
```

previous series

```
/ \b ( ? : [ A-Z ] { 1,2 } \d { 1,4 } | [ A-Z ] { 3 } \d { 1,3 } ) \b /
```

old series - letters first example: J407LCM

```
/ \b ( ? : \d { 1,4 } [ A-Z ] { 1,2 } | \d { 1,3 } [ A-Z ] { 3 } ) \b /
```

old series - digits first example: 404WHC

UK National Insurance number

example: YX 17 11 94 A

```
/ \b [ a-z ] { 2 } \s ? \d { 2 } \s ? \d { 2 } \s ? \d { 2 } \s ? [ a-z ] \b / i
```

Book's ISBN number

examples: 1-56592-257-3 and 0-596-00205-X

```
/ \b ( ? : [ \d ] - ? ) { 9 } [ \d x X ] \b /
```

Currency - price in pounds

example: £19.50

```
/ \b \xa3 ? \s * \d + \. \d { 2 } \b /
```

Bank Sort code

example: 40-36-17

```
/ \b \d { 2 } - ? \d { 2 } - ? \d { 2 } \b /
```

Notes:

- We've used non-capture groups throughout. You're free to use our expressions and add in extra brackets if you wish to extract parts of the match.
- Some of these examples are fairly rudimentary tests. It's up to you to check they're suitable for your application.
- Many of these examples can only check that a string of text is in the correct format. There's no way that we can check if a particular car registration was issued, unless of course we have access to the DVLC database. :-)

Regular Expressions

continued from previous page

Here's an example of some regular expressions in use to search through our DVD collection.

The first examples are looking for literals only:

- Film titles containing a `:
enter a pattern `America's Sweethearts
Bridget Jones's Diary
- Titles containing the string "ger":
enter a pattern **ger**
The Wedding Singer
Tiger
Crouching Tiger
Hidden Dragon
- Let's now add a search for a character group; "." means any character, so we looking for the sequence anything - T - h - e:
enter a pattern **.The**
Dollar For The Dead
The Dark Side Of The Sun
The Lord Of The Rings
- And now we'll look for anything that contains a character which is NOT a space, letter or digit:
enter a pattern
[^ a - z A - Z 0 - 9]
Brimstone & Treacle
Lara Croft - Tomb Raider
M:I-2
America's Sweethearts
Bridget Jones's Diary
- Let's find all the sequels by looking for file names that include any digit:
enter a pattern **\d**
The Mummy 2
Howling 6
M:I-2
Toy Story 2
- Now let's use anchors to look for titles starting with "The":
enter a pattern **^The**
The Peacekeeper
The Wedding Singer
The Mask
The Mummy 2
The Whistle Blower
The Grinch
The Dark Side Of The Sun
The Perfect Storm
The Patriot
The Piano
The Color Purple
The Lord Of The Rings
- ...and ending with "ger":
enter a pattern **ger\$**
The Wedding Singer
Tiger
- Now count; lets look for "g", perhaps "a" and "l":
enter a pattern **ga?l**
Coyote Ugly
Legally Blonde

The following public courses are scheduled at our Melksham, Wiltshire training centre.

Dates are until mid 2003; see <http://www.wellho.net> for further dates. If you have four or more trainees, we'll run an extra private course for you at our training centre or at your office, and tailored to meet individual company needs.

PERL

Perl is a programming language used in data manipulation, web interaction and systems administration. Its wide range of facilities make it ideal for large and small jobs, and it can act as an excellent glueware between otherwise unconnectable applications.

We offer two different foundation courses depending on your background, and two advanced courses to meet specific application needs.

Learning to Program in Perl

A foundation course for those who haven't programmed before, or who are rusty. Covers the fundamentals of programming in Perl, and the major features of the language that are required in most applications.

☒ 5-day course, runs:

Monday 25 November 2002 to

Friday 29 November 2002

Monday 17 February 2003 to

Friday 21 February 2003

Monday 19 May 2003 to Friday

23 May 2003

Perl Programming

For those with programming experience. Covers the fundamentals of Perl by comparing the language with other languages, then proceeds on to the major features of the language that are used in most applications of Perl.

☒ 4-day course, runs:

Monday 2 September 2002 to

Thursday 5 September 2002

Monday 4 November 2002 to

Thursday 7 November 2002

Tuesday 7 January 2003 to Friday

10 January 2003

Monday 31 March 2003 to

Thursday 3 April 2003

Using Perl on the Web

Using Perl on a web server through CGI, and other web-based interfaces such as automated FTP, clients, servers, modPerl, etc. An advanced course that assumes fundamental Perl knowledge.

☒ 2-day course, runs:

Monday 16 September 2002 and

Tuesday 17 September 2002

Monday 2 December 2002 and

Tuesday 3 December 2002

Monday 17 March 2003 and

Tuesday 18 March 2003

Perl for Larger Projects

Scheduled Courses

Prices

1-day courses	£ 290.00	470.00 euros
2-day courses	£ 490.00	800.00 euros
3-day courses	£ 690.00	1,130.00 euros
4-day courses	£ 890.00	1,460.00 euros
5-day courses	£ 1,090.00	1,790.00 euros

For multiple students from the same company booked on the same course, simply calculate the total cost using the formula:

£200.00 for each student per day plus £90.00

(e.g. 2 students / 4 days = £1,690.00)

Prices are valid for all courses offered in this schedule and on our web site <http://www.wellho.net> for bookings confirmed by 31st December 2002.

This advanced course covers topics that you'll need to know if you're using Perl for a large project. Content includes variable scoping, namespaces, object orientation, POD, XML, SQL database access, and more.

☒ 3-day course, runs:

Wednesday 18 September 2002 to

Friday 20 September 2002

Wednesday 4 December 2002 to

Friday 6 December 2002

Wednesday 19 March 2003 to

Friday 21 March 2003

Perl Extra

Perl Extra is only available to individual trainees who have attended one of our public Perl courses, and require further advanced training in a combination not offered by our other advanced training.

☒ Dates are scheduled to be mutually convenient to the trainee and the trainer.

JAVA 2

Java 2 is a wide-ranging, portable, network-aware language that you may wish to run on servers, on clients, or on stand-alone hardware. We'll have you writing JSP pages, servlets and applets by the end of your training, no matter whether or not you've programmed before.

Java Programming for the Web

This course takes the newcomer to programming right through to writing JSP, servlets and applets. It covers the world of Java, programming in Java, and the programmer interfaces used on Java web-based applications.

☒ 5-day course, runs:

Monday 7 October 2002 to Friday

11 October 2002

Monday 13 January 2003 to

Friday 17 January 2003

Monday 28 April 2003 to Friday 2

May 2003

Java Extra

Java Extra is only available to individual trainees who have attended our public Java courses, and require further advanced training.

☒ Dates scheduled to be mutually convenient to the trainee and trainer.

PHP

PHP is a language that you embed in

web pages to run on your server. You should already know something of the web, HTML, SQL and programming. We offer two courses: a technology day that covers enough of those prerequisites to get you started, and a PHP programming course that covers both the language and its web use.

Technology for PHP

The structure of the web, web pages through HTML and relational database access through SQL covering the prerequisites needed to follow on to our PHP programming course.

☒ 1-day course, runs:

Monday 9 December 2002

Monday 3 March 2003

Tuesday 27 May 2003

PHP Programming

This course covers programming in the PHP language, and also topics such as state, cookies, interfacing to SQL and XML handling, and emailing from the server that you'll need to write effective server side code.

☒ 3-day course, runs:

Tuesday 10 December 2002 to

Thursday 12 December 2002

Tuesday 4 March 2003 to

Thursday 6 March 2003

Wednesday 28 May 2003 to

Friday 30 May 2003

MYSQL

This course introduces you to the world of SQL database engines. We study the SQL language and how it's used from PHP, Perl and Java, and administration including user account and security controls, and database design so that you'll be putting together well thought out, maintainable, expandable databases.

MySQL

Our MySQL course covers the structure of a relational database, the SQL language, database design, installation and administration of MySQL, and its use from open source languages.

☒ 2-day course, runs:

Monday 23 September 2002 and

Tuesday 24 September 2002

Monday 16 December 2002 and

Tuesday 17 December 2002

Monday 10 March 2003 and

Tuesday 11 March 2003

Monday 16 June 2003 and

Tuesday 17 June 2003

TCL, TCL/TK and EXPECT

The Tcl language is a scripting language with a wide range of extensions, such as Expect and Tk, which make it very effective for applications such as writing Graphic User Interfaces and automating repetitive operator tasks.

Tcl and Expect Basics

This course teaches you Tcl, from fundamentals through most of the major features, and also the Expect and [Incr-Tcl] extensions, but not the Tk extension.

☒ 3-day course, runs:

Monday 18 November 2002 to

Wednesday 20 November 2002

Monday 3 February 2003 to

Wednesday 5 February 2003

Monday 9 June 2003 to

Wednesday 11 June 2003

Tcl/Tk and Expect

This course covers the Tk extension to Tcl which provides the Graphic User Interface. Learn how to build widgets, how to pack them into frames, and how to use them to provide interactive, GUI-based programs. A knowledge of Tcl is a prerequisite of this course.

☒ 2-day course, runs:

Thursday 21 November 2002 and

Friday 22 November 2002

Thursday 6 February 2003 and

Friday 7 February 2003

Thursday 12 June 2003 and

Friday 13 June 2003

RUBY

Ruby is a modern Object Oriented scripting language that's rapidly growing in popularity.

Ruby Programming

Our Ruby course takes you rapidly through the fundamentals of the language then concentrates on the classes and libraries provided so that you can use Ruby on the Web, for database access, to handle XML and more. For those with prior programming experience in another language.

☒ 2-day course, runs:

Thursday 26 September 2002 and

Friday 27 September 2002

Wednesday 18 December and

Thursday 19 December 2002

Wednesday 18 June 2003 and

Thursday 19 June 2003

Honesty works, even in the training business

by Graham Ellis, Well House Consultants Ltd.

WSYWIG - "What you see is what you get" - used to be a big feature of editing programs; now it's expected that any editor worth its salt will automatically give you a running preview of your work, so there will be no nasty surprises at the last minute when you think your job is done.

I believe in providing "WSYWIG" training. Our course description may not look so glamorous on the web site as one that's padded out with buzzwords, but it tells you what you're getting. I've worked in offices where there's been bad news that we've had to pass to our customers and topic "what shall we tell the customer" has been discussed. Well - if you *do* end up in such a situation, how about telling the truth ;-)

We don't just take this "WYSIWYG" approach to training just because we're going for the moral high ground; we also go for it because it makes excellent business sense. Happy customers - customers who receive the product they expect (or perhaps a little more than they expect) tell their friends and colleagues and are our ambassadors, and it's much easier for us to take further bookings if we're recommended to people than if they've just happened across us on line.

Our approach is different - but it works. It's a real pleasure training customers who are friends, rather than just customers, and our circle of friends

grows. Amongst our customers, we can number big banks and governmental outfits, pharmaceutical giants and large retailers, engineering companies and high tech R&D departments ... and also the startup operation down the road that just happens to use PHP or MySQL!

Have a look at page 3, and see our humble beginnings in 1995; then look at page 15, and see what a wide range of courses we're running now from our purpose fitted training centre. Our approach works!

Let's see how we make "What you see is what you get" work; here are some typical questions that we're asked from time to time, with the answers.

Q "You're selling me the course. Will you tell the tutor why I'm coming on it?"

A "No need - I am the tutor". We don't employ a separate sales team; if you have a question on course content, then it'll be the tutor who answers it for you. It might take a few hours for him to get back to you (perhaps by an overnight email), but we think you'll prefer the correct answer, even if it takes longer to reach you, to a rushed response.

Q "How do I know this is the right course for me?"

A Because the tutor will only accept your booking if he knows the course will be good for you. It's not in his interest to accept a booking from you if you'll be sitting in the course, unhappy that you're in the wrong place.

Q "XXX list extra content for their course of the same length"

A Maybe they do, but how much coverage do they have time to give that extra content? To ensure WYSIWYG, the web site descriptions of our courses, the booking confirmations, the course notes, and the examples that the tutor projects ALL come from the same electronic source. We may well mention a lot of extras that XXX describes as being on their course, but in fairness to a balanced description it won't appear on our web site unless it has significant coverage.

Q "You say the course is full, but it's important that I come. Surely there is room for one more?"

A Sorry - no, there isn't. We take a maximum of eight trainees on a public course; more than that, and the tutor doesn't have sufficient time to give everyone the personal attention they deserve. We do always have a spare workstation, mind, just in case of a hardware failure.

Q "Should I leave my hotel booking to the last minute in case you don't have enough people book?"

A No, book now. Melksham is popular and the nearby B&Bs can get fully booked. We don't cancel courses; it's unusual for a course to run with just one trainee, but it has happened.

Q "Do you want me to pay before you'll confirm my place?"

A We may ask for a deposit, but we don't expect you to pay for the course as a whole until you actually attend; if you're with a company with a good credit standing, we'll invoice on completion of the course, and ask you to pay within 28 days.

Q "What discount will you give me?"

A None. You don't expect to go into Tesco and negotiate a discount on your bread, or expect to get a discount off your train ticket, do you? We're very much aware that the computer training industry has been in the habit of increasing the list price of courses so that it can heavily discount to everyone. As I was writing this article, I checked the current price of a company just up the road from us who offer a Java course. They're quoting £1595.00 + VAT, compared to our £1090.00 + VAT; for sure, they're offering a 30% discount off the course that starts on Monday - that's still more expensive than our course, and I'll bet the group size would be larger too! (By the way, I'm writing this on Saturday.)

Q "Will you give me a discount on the course that starts tomorrow; isn't some

income better than none?"

A No; we're not desperate to fill every place on every course, and other customers would feel cheated because they've paid more. Have you ever sat beside some smug b****rd when you're on an airline flight who's shouting out just how cheap he got his ticket? Doesn't it make you feel good to know you paid more? I think not!

Q "What discount will you give me? - the course isn't for me, it's for one of my customers"

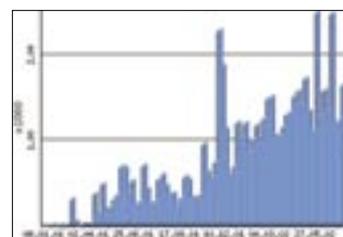
A We won't give you a discount. You are free to charge your customer whatever you like for our course, plus your own fees - and he'll probably end up paying less overall than he would for a course elsewhere. We believe in transparent pricing, and we explain this policy on our web site. You're welcome to copy it to your customer by way of explanation.

I find it tremendous fun running Well House Consultants this way, and customers tell me that the approach is refreshing. That's not just lip service to us, either; look at how we've grown in the last few years and you'll see that our customers are voting for us with their feet.

Our administrator Lisa came up with a catch phrase "Come as a customer, leave as a friend" and we both felt it was a bit corny. Truth is, we provide WYSIWYG training, and the catch phrase is right on cue, even though a bit twee.

It's Sunday tomorrow, and a friend of ours is dropping by for a coffee. We think we'll end up discussing the latest developments in Perl.

We're growing!



Growth over the last two years of our quiz web sites

There seems so much "doom and gloom" around in the training business at the moment - companies downsizing, merging and closing facilities, and heavy discounts off inflated list prices. We have noticed that some sectors of customers are quieter at the moment, but overall business grows; clearly, we have a product that customer companies keep coming back for.

Your contacts at Well House Consultants:

Administration:	Lisa Ellis	lisa@wellho.net	
Technical:	Graham Ellis	graham@wellho.net	
Phone:	01225 708 225	+44 1225 708	225
(International)			
Fax:	01225 707 126	+44 1225 707	126
(International)			
Web Site:	http://www.wellho.net		
Postal Address:	404, The Spa, Melksham, Wiltshire, UK SN12 6QL		

Our office hours are 08:00 to 18:00 Monday to Friday.

You'll usually reach us at weekend and in the evenings too, as we often divert the phone when we're not around. If you email us, we'll receive your email within 24 hours, 365 days a year.

This newsletter is published by Well House Consultants Ltd. and is distributed free. Please contact us if you're reading someone else's copy and would like to be added to the circulation list, or if you would like to be removed from our list.

All material in this newsletter is copyright Well House Consultants, Ltd., and may not be reproduced in any form or by any means electronic without prior written permission of Well House Consultants except in the case of brief quotations embodied in critical articles and reviews.

Well House Consultants has attempted throughout this publication to distinguish proprietary trademarks from descriptive terms by following the capitalisation style used by the manufacturer.

