# of course

VOLUME 6, NUMBER 1

SUMMER 2006

LUNCH OPTIONS

## What’s on your plate?

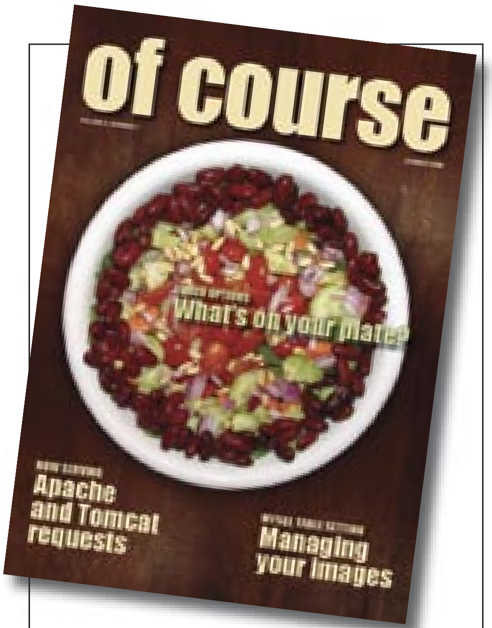
NOW SERVING

## Apache and Tomcat requests

MYSQL TABLE SETTING

## Managing your images

## Plus Perl, PHP, Python, Tcl and Linux



## Contents

## What's the difference?

**Q:** Why are the two different web servers both called "Apache"?

**A:** Apache httpd is a great server for sites with lots of content pages, whereas Apache Tomcat is designed as a container for Java applications where visitors will be running to a relatively few URLs which they'll need to access time and time again.

## Addition question

What's 2 + 2?

4, you say?

Yes, that's right. So, what's "Peter" + "Jones"?

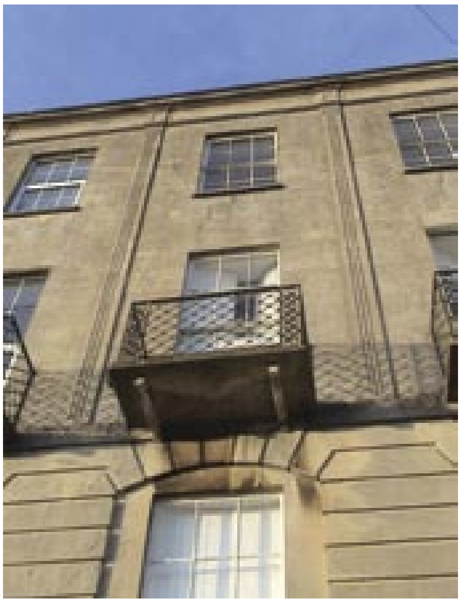
Trickier one, but you might say "Peter Jones".

What do you get when you add two rabbits together? Probably a lot more rabbits!

In Python, the + operator is *overloaded*, which means it does different things depending on what type of objects you're adding together.

You can even define your own additions for your own types of objects; + is just a pseudonym for the `__add__` method.

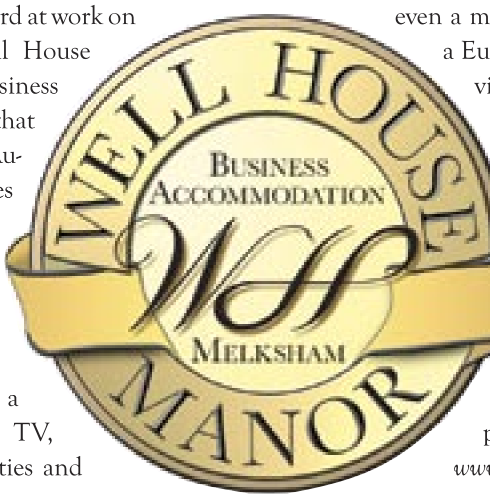
## Our centre of character and history . . .



In 1814, local entrepreneurs planned for Melksham to become a Spa town to rival Bath, just 12 miles away. A ballroom, a well head and three lodging blocks were built on the outskirts of the town where the chalybeate waters had  been found.

Well House Consultants is based in one of the old Spa Houses, and that's where our current training facilities are located.

## . . . is about to be joined by another



Lisa has been hard at work on drawings for Well House Manor – our business accommodation that opens in the Autumn for delegates attending courses in Melksham.

Each room will feature wired as well as wireless internet access, a large flat screen TV, full en suite facilities and even a mobile phone charger and a European electric outlet for visitors from outside the UK. Add to that all the more usual elements you find in modern high quality hotel rooms, and the drawings alone are no small task.

Watch our progress on the web site: *www.wellhousemanor.co.uk.*

## Training efforts

With some 40% of our public course trainees coming to Melksham by the train that arrives at 09:12, we're very concerned of the threat that this train will be withdrawn from December, along with many others. It would leave no services calling at the station at all between quarter to 8 in the morning and half past six at night.

We have various contingency plans in place, which we'll be describing much more fully in our next edition once we know for definite what's happening. Let's just say at this stage that you'll find our contingency plans are even better than what we're offering at the moment.

For full update on the train story, see the site we sponsor at *www.savethetrain.org.uk*.

A WORD FROM GRAHAM ELLIS

## Busy-ness booms

It's been a wild six months with 2006 starting our busiest on record, both in terms of our training and other activities.

You'll see a small article in this newsletter concerning our "save the train" campaign, which has taken up a huge amount of my time, and gained me quite some exposure in the local press, and even lead to parliamentary questions.

New training material includes an Introduction to JBoss deployment, LDAP, and setting up a DNS server, all of which are available only on private courses at the moment. And we've so many new tips and techniques and articles on MySQL that that there's probably a book in there, if only there was time!

Biggest news is our "Well House Manor" project. When we moved to Melksham, there was an abundance of local accommodation available for delegates staying in the town to attend courses. But over the five or six years, customer expectations have changed and our customers expect to be able to pay by credit card, they require en suite facilities, they want a light and healthy breakfast, they appreciate being close to our centre with good public transport links, and they need an internet connection to use after class finishes each day. They would also prefer to staying in the same hotel as other delegates on the same course.

We're in the process of purchasing a large property just up the road from us; some of you will have stayed there when it was operating as "The Old Manor", and come the Autumn we'll be opening as a business accommodation, meeting all those needs (and more).

Just five minutes' walk from the town centre, set in its own 0.75 acres, it'll be a quiet venue for courses. It's another character building which we'll be bringing up to modern standards and will help us raise the Well House Consultants Experience to a new level. An exciting development indeed!

Graham

## Sorting dandruff using Perl's sort and map functions

Sometimes, I call it "Sorting Dandruff". It's a clutter of tiny subs at the end of a Perl program which are used to provide callbacks for functions like sort and map.

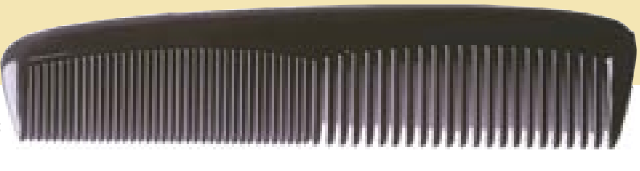
Example – code to capitalise and sort a list:

```
# Sorting Dandruff

@salad = qw(lettuce cucumber tomato pepper);
@sal2 = map(cfix($_),@salad);
@sal3 = sort bylength (@sal2);
print “@sal3\n”;

sub cfix {ucfirst(lc($_[0]));}
sub bylength {length $a <=> length $b}
```

I can simply write the little flakes of code that have fallen to the bottom in as anonymous blocks of code (in the case of sort) or as expressions (in the case of map).

Here's the better code:

```
# A Head and Shoulders above better example

@salad = qw(lettuce cucumber tomato pepper);
@sal2 = map(ucfirst(lc($_)),@salad);
@sal3 = sort {length $a <=> length $b}
      (@sal2);
print “@sal3\n”;
```



# What language should I learn? Perl? PHP? Tcl?

That's a question I'm asked from time to time, often followed up with "Which are the upcoming open source languages, and which are not growing so rapidly?"

When asked which language someone should learn, I tend to answer their question with a torrent of other questions - every language that we teach is superb for a certain range of applications, and it's vital to establish just what my questioner wants to do, what his current skill set is, and what any colleagues of his are using. I then have a good chance of coming up with two or three recommendations and going into them in more detail.

*At press time, we counted 633 books in our library, all relating to computer programming. Even for us the idea of reading every one cover-to-cover is daunting. But, if you don't know what a language is all about, how do you know if it's the right one for you? How do you know it's not?*

When it comes to talking about rising and falling languages, I have a gut feeling that I can answer with but I've had little concrete data to back that up. However, I was thumbing through a 2004 edition of "Of Course" over breakfast this morning, and I came across a table of search engine arrivals based on keywords that we published.

How have things changed in the two years since? I checked up on last week's stats ...

Interest in PHP has risen dramatically - up from 1980 visitors per week in 2004 to over 4000 visitors a week this year. By contrast, Perl has only drifted upwards although it remains very slightly ahead of PHP (perhaps it will have been overtaken by the time this edition goes to print). Both Python and MySQL have also shot up - nearly three fold in each case - although starting from a lower base than Perl or PHP, they're catching up too. And Java has grown as well; although we no longer offer public Java courses in general, we do provide Apache httpd and Tomcat deployment training and I can confirm the robust health of the Java Web Application community.

Alone amongst our key topics, there has been a slight fallback - around 10% - in the number of queries that arrive with "Tcl" in there query string. I'm not surprised - Tcl's now quite an old language and although it's great for certain niche uses I think it's dropped from mainstream programming.

And any new and upcoming subjects on the horizon? Fewer than you might think - PHP, Python and MySQL are all excellent tools and if it ain't bust, then why fix it? Perl 6 is in many ways a new language and I'm sure we'll be seeing and hearing a lot more of that in the future, though it's taken so long to get to the current stage that I'm fearing it will wither on the vine. And there's a lot of talk about Ruby too, but as yet the activity from the community does bring it in as a high profile player. But it's one to watch.

# Brag Page

## Ah, it's a hard life, but someone's got to do it.

We have a lot to brag about! Students on courses ... we're proud to have trained a wide variety of students on public and private (single company) courses, at our training centre and at their offices throughout the British Isles and beyond.

# Model, view, controller: Not just a Java mantra

“Model, View, Controller.” So says the mantra of Java design.

Is this applicable to Perl or Python applications too? Yes, absolutely it is!

Why are you writing code? Because you have DATA to handle. That’s the “model,” and it’s a great idea to have all your data handling done via an application programmer interface. That way, the way that your data is stored can be varied with time, and provided that the interface is left compatible (you may extend it though), you’ve got a clean and clear way of talking data irrespective of whether it’s held in a database, in XML or in plain text files.

The “view” is the look and feel, the appearance, of the results you display. Typically, you’ll want the view to be in the hands of someone with a degree of human interfacing and graphic art skills. There’s likely to be HTML involved if the presentation is to be a web page.

Between the view and the model you have the “controller,” the business logic that knows what it wants of the data, and presents the view with the information required based on requests passed in. The controller should not be aware of how the viewer will be presenting the data. In a good setup it will not generate any HTML, leaving that to the viewer.

Nor should it be aware of the data storage being used within the model. In a good setup, it will pass requests in to that layer and be unaware of which architecture is hidden.

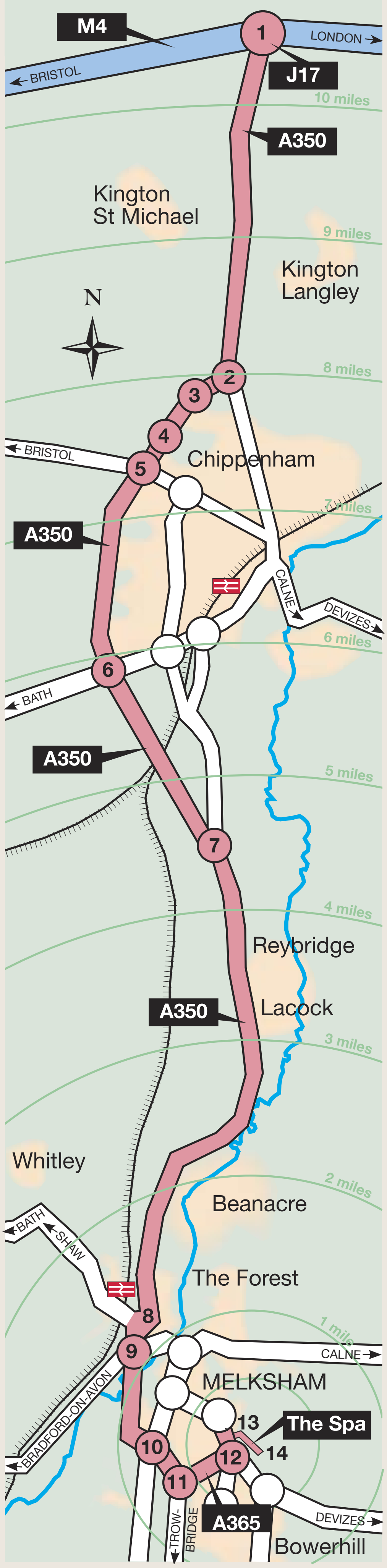
Why is MVC such a good idea? Because it allows each of the elements of your application to be tested and re-used separately. In our web site redesign of last Autumn, our old “test your knowledge” quizzes got overlooked and I was somewhat embarrassed to find, rather too long after the relaunch, them showing the old look and feel.

And that could have been a problem, especially as the scripts were written in Perl, whereas the new look pages all require PHP. But, in stepped the MVC-based design and I had the quizzes’, complete with new look, running in a couple of hours.

Because the interface between the view and the controller is well defined, I was able to put in our standard PHP framework to front the application, with calls to Perl code to do the business for each page. The model is currently held in plain text files but I have plans to switch that across to MySQL when I get a chance. Again, that should be easy since it’s just the model to change.

Our online quizzes are available for you at *www.wellho.net/quiz/*.

Test your knowledge and see how you get on. Or, email us your suggestions for some more questions ;-)



# Getting to our training centre

**DIRECTIONS FROM THE M4**

1. Start out at M4 Jct 17 (A350/A429/B4122),Wilts
   Follow signs for Chippenham on the A350
2. Malmesbury Road Roundabout
   At roundabout take 2nd exit following the A350
3. A350 Badger
   At roundabout continue forward
4. Brook
   At roundabout continue forward
5. Bumpers Farm (Junction with A420)
   At roundabout take 3rd exit
6. Chequers (Junction with A4)
   At roundabout take 3rd exit (signposted Poole, Warminster)
7. A350 Lackham
   At roundabout take 3rd exit
8. A350 Junction with A365
   At traffic signals forward
9. A350 Junction with A3102/B3107
   At roundabout take 2nd exit
10. Berryfields roundabout
    At roundabout take 2nd exit (signposted Devizes)
11. A350/A365 Junction
    At roundabout take 1st exit (signposted Devizes)
12. B3112
    At roundabout take the 1st exit onto Spa Road - B3112
13. The Spa
    Turn right into cul-de-sac
14. Well House Consultants Ltd
    Turn left in last gated entry at 404, The Spa, Melksham (4-storey Georgian)

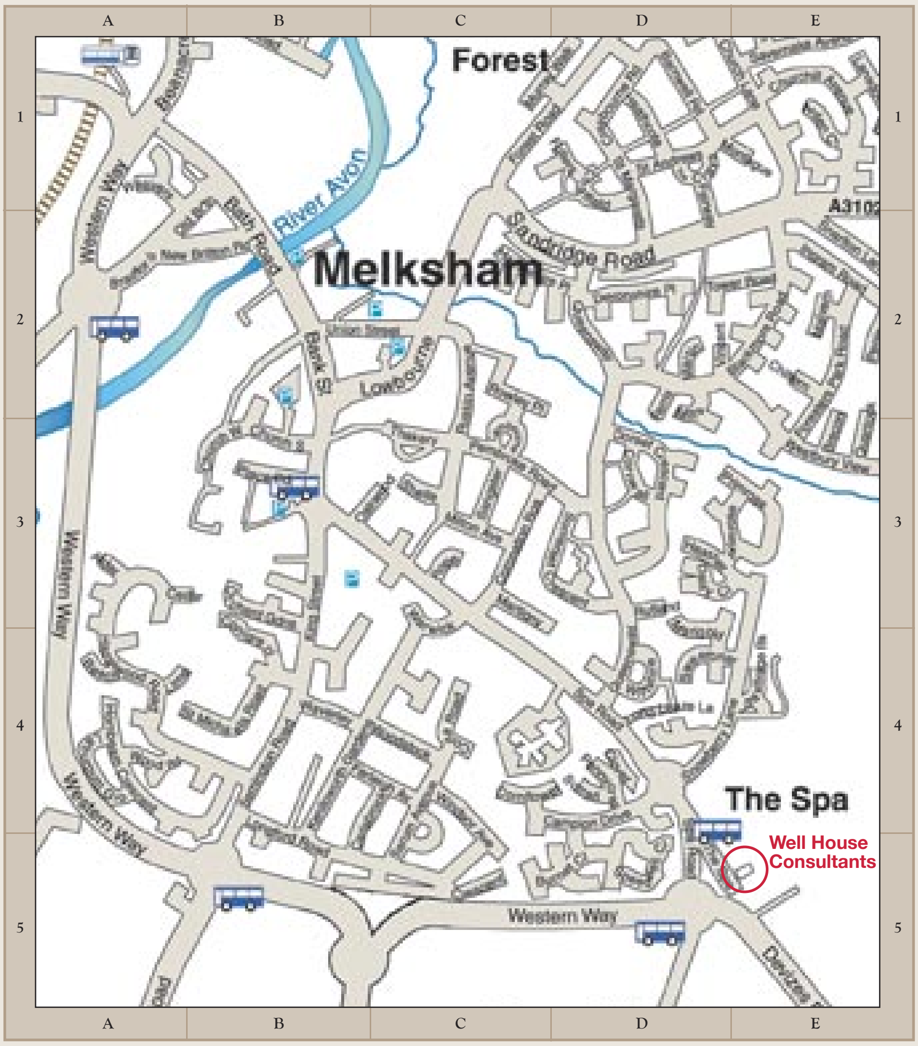
A350

A365

B3112

unclassified

Estimated travel time: 20 minutes in non-adverse conditions



# Phalanx Project scores on practicality

The Phalanx Project is designed to strengthen the documentation and testing of the most popular modules on the CPAN. *http://qa.perl.org/phalanx/*

Let me explain a little further. Perl is a powerful language and it can do a lot of things. It's also very widely used, so chances are that if you want to do xxxx or yyyy with Perl, it's probably been done before.

So why re-invent the wheel?

It just wouldn't be practical or sensible to distribute all extra modules that people might like with Perl. The download would skyrocket in size, and everyone would have so much extra code that it would be very hard for them to separate the stuff that's useful to them from the frankly quirky.

And so the CPAN – Comprehensive Perl Archive Network – was born, where module writers can contribute their code, and budding module users can come and look for ready-written modules.

I've just visited the CPAN as I'm writing this article, and searched for XML. It offered me 2,212 results. Hmmm, now which one is best for me?

Should I go through them all? And will they be well documented to help me make my final decision? I'm sure you see the problem!

The Phalanx project works on the 80/20 principle, except that they're calling it the "98/2 principle". 98% of all downloads from the CPAN are for about 2% of the modules. So they've chosen the top 100 modules, and they're giving them special attention by indexing, testing, documentation, and making the way clear for quick and easy access and use of these particular gems.

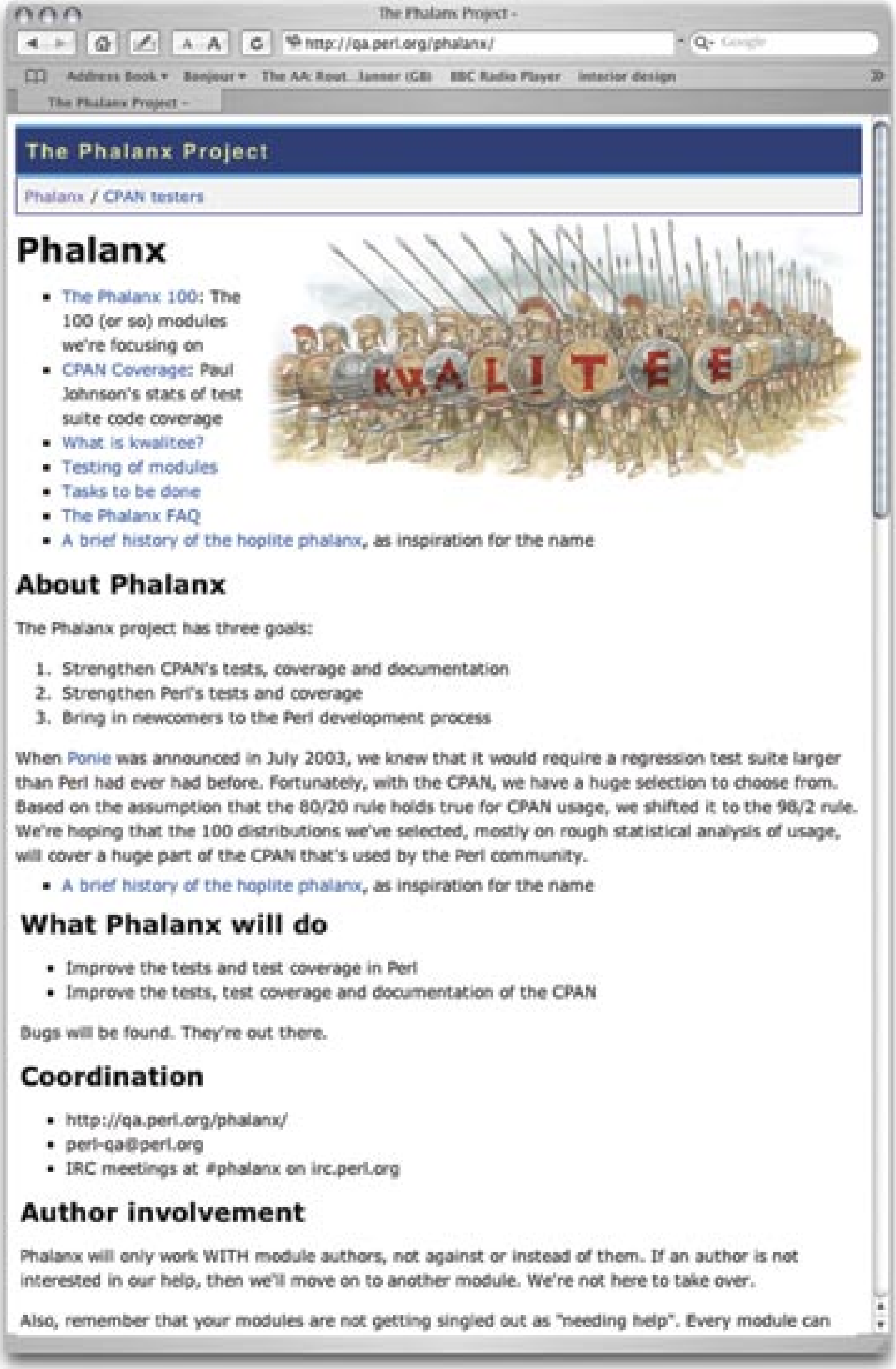
I have to agree with their action and their choice. I suggest you bookmark *http://qa.perl.org/phalanx/100/* and have a look at the modules they've selected.

Looking in the top 10 we regularly use, even on our public courses:

- **DBI** – the Database Independent Accessors
- **DBD-mysql** to connect the DBI module to the MySQL database
- **libwww-perl** including LWP – Library for Web Processes
- **XML::Parser** (I think the names tells you what this does)

Looking a little further down the list, it includes other favourites like:

- **libnet** which we use to make FTP and Telnet connections from Perl
- **Time-HiRes** which we use if we need something tighter than second intervals for timing
- **CGI.pm** which provides a very heavy package of CGI support.



The Phalanx Project

Phalanx / CPAN testers

**Phalanx**

- The Phalanx 100: The 100 (or so) modules we're focusing on
- CPAN Coverage: Paul Johnson's stats of test suite code coverage
- What is kwalitee?
- Testing of modules
- Tasks to be done
- The Phalanx FAQ
- A brief history of the hoplite phalanx, as inspiration for the name

**About Phalanx**

The Phalanx project has three goals:

1. Strengthen CPAN's tests, coverage and documentation
2. Strengthen Perl's tests and coverage
3. Bring in newcomers to the Perl development process

When Ponie was announced in July 2003, we knew that it would require a regression test suite larger than Perl had ever had before. Fortunately, with the CPAN, we have a huge selection to choose from. Based on the assumption that the 80/20 rule holds true for CPAN usage, we shifted it to the 98/2 rule. We're hoping that the 100 distributions we've selected, mostly on rough statistical analysis of usage, will cover a huge part of the CPAN that's used by the Perl community.

- A brief history of the hoplite phalanx, as inspiration for the name

**What Phalanx will do**

- Improve the tests and test coverage in Perl
- Improve the tests, test coverage and documentation of the CPAN

Bugs will be found. They're out there.

**Coordination**

- http://qa.perl.org/phalanx/
- perl-qa@perl.org
- IRC meetings at #phalanx on irc.perl.org

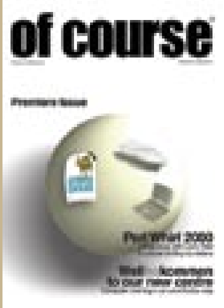
**Author involvement**

Phalanx will only work WITH module authors, not against or instead of them. If an author is not interested in our help, then we'll move on to another module. We're not here to take over.

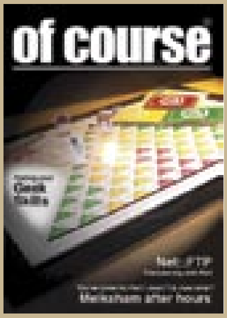
Also, remember that your modules are not getting singled out as "needing help". Every module can

# Back Issues

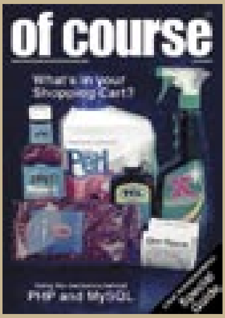
*All back issues are available by post to UK addresses at no charge. Please phone or email us for a copy or two. Copies of recent editions are also available to download from our web site: http://www.wellho.net/net/ofcourse.html.*



**Issue 1**
**Winter 2000/01**
**8 pages**
Perl Whirl 2000
Wellhokommen, Bien Venue
Tcl/Tk and Expect
Handling Huge Data
Book Review - The "New Camel"
Perl for Larger
The London and Dublin connection



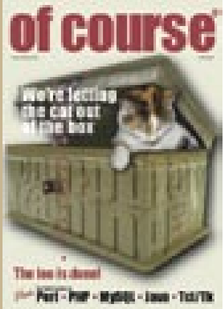
**Issue 2**
**Spring 2001**
**16 pages**
Quiz - Perl, Java and Tcl
Using FTP from Perl
In and around Melksham
Blistering Speed right to the laptop
Is Open Source right for you?
Focus on Perl Standards
Web logs and market research



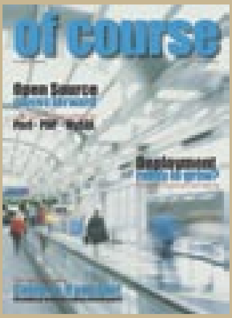
**Issue 3**
**Autumn 2001**
**16 pages**
Behind PHP
Java for a chat room
What is XML?
Relating to MySQL
Wireless networking
Sessions and State
On the Oban road
State of the Onion
Are you under attack from Codered?

**Issue 4**
**Winter/Spring 2002**
**16 pages**
Keeping data secure
Extracting text from Microsoft Word documents in Perl
Improved security in PHP 4.1
Daemons
Cookies - European law and regulations
The Parrot virtual machine
Case Study - MPC Data embed MySQL
More Dynamic than an array
Expect / Tcl application
Book review - "The Java Cookbook"

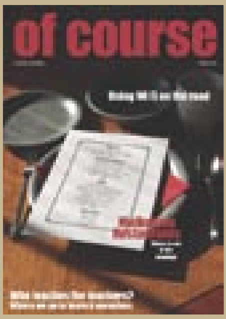
**Issue 5**
**Autumn 2002**
**16 pages**
RegEx
Tcl up to date
Learning Perl as a first language
Tk - screen updates
Portable PHP code
Using PHP in standalone applications
Running scripting languages fast
Open Source Bulletin Board



**Issue 6**
**Spring 2003**
**16 pages**
Perl 6 introduction
Public sector data
When size matters
Online e-commerce
10 relevant projects from SourceForge
Data Protection act - buyer beware
Setting up Mac OS X
Data Mining



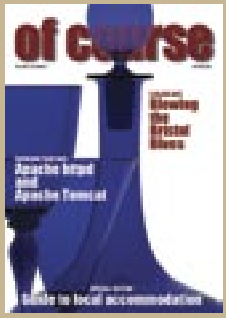
**Issue 7**
**Autumn 2003**
**24 pages**
PHP sticky boxes
Perl 5, Ponie, Perl 6
SQLite
Python and Jython
Broadband update
Data structure - link or copy?
Deployment training - new growth
Tailoring your site



**Issue 8**
**Spring 2004**
**24 pages**
Choosing which OS
OO design
What's a polecat?
Objects in Perl
Who teaches the teacher?
Broadband Access for the Traveller
Logging in to the BT Openzone service
Dining out in Melksham
References and variables in Perl
What do you put into a file?
What defines quality software?

**Issue 9**
**Autumn 2004**
**24 pages**
Legal aspects of a web site
Short tags in PHP
The speed of three
A numbers riddle
Handling huge data
PHP5: Switch?
Web site visibility
Mint under the pillow
Dynamics of web pictures
Extreme Programming
OOP DeMYSTiFieD

**Issue 10**
**Spring 2005**
**24 pages**
Geekmas greetings
Navigation skills
Embedding Layers
Libraries, real/virtual
Who tends MySQL?
Welcoming Leah
Seat pricing GUI in Tcl/Tk
You can do a lot with a little Python
Cruising with the Geek gods
Make your site's URLs more friendly
Simplifying complex joins
Apress's distinctive looks filling shelves
Distributing open source applications
Java course notes now available free



**Issue 11**
**Autumn 2005**
**24 pages**
He who dares, Forums
Bristol Blue Glass data study
Is it the place to be?
Wiki Watch
The X-Y-Z's of sharing
Function Convention
Brag Page
Expect for Windows
Getting to our centre
PHP: it takes all sorts
The makings of a cat
Local accommodation
Mapping your site
Apache httpd and Tomcat
A training web site
Major web site update
Your r's from your x's

# Options that ta

...and on the second day we had baguettes.
So many to choose from!

From: Liv
Date: 25 Ap
To: graham@we
Subject: best coffe

Hi Graham,

I hope all is well for your trainin business (I am sure it is) and the train still running.

I was wondering if you could let me know the brand an best coffee machine I have ever seen, the one whi room!

By the way, I am now doing a lot larger project" was a good p used with http://www nda.c legendary "Effecti rogra Randal L. S ead.

prog

R

Liv

# ake the biscuit

*From beverage service to course lunches, we not only offer a variety for every palate, but do it in a way that has people asking, "Where can I get my own?"*

It's not often that you go into a restaurant and you're so taken with the food, and so at ease, that you ask for the recipe. And yet we're very used to delegates on our courses remarking about the Wiltshire food, or asking us about our coffee machine, or "Where do you get those biscuits?"

We pride ourselves in making business a pleasure, and that extends not only to the learning experience but the snacks, drinks and lunches that we provide on our training courses.

On the first day of a longer public course, we typically take a business lunch at a local Indian or Chinese restaurant or pub. It gives our delegates a chance to get to know the people they'll be spending the rest of the week with, and it ensures there's a wide variety of food available just in case delegates have any preferences or limits as to what they can eat ahead of time.

The actual decision as to where we eat is left to the last minute, as the Three Magpies, Refa Tandoori, and Lee's Chinese restaurant can all put together a table for eight – the maximum of seven delegates plus the tutor – at half an hour's notice.

They say that variety is the spice of life, and right through your course, lunch will vary day by day. Come into the training room after 5:00 pm, just after the first day of the course, and you'll probably find lunch options listed on the board, with some crossed out and others annotated "Tuesday", "Wednesday" etc.

What are some of those options? The Baguette Bar in Melksham provides excellent filled baguettes, and delegates are invited to choose from the spread on show in the kitchen, and eat seated either in our more formal dining room (which can take the whole of the largest class), in the customer lounge if they prefer to be more casual, or even in the gardens on a fine day.

Tricia's Plaice was voted the best chippie in the whole of Wiltshire, and Tricia is just five minutes up the road from us. We have

*continued on next page*

# Latest Versions of Open Source Software

as of May 2006

| | | |
|---|---|---|
| **Perl**<br>*see http://www.cpan.org for additional modules* | **5.8.8** Stable/Latest<br>**5.9.3** Development<br>5.8.8.817 for Windows | *www.perl.com*<br>*www.activestate.com* |
| **Tcl/Tk** | **8.4.13.0** Stable<br>**8.5a4** Alpha release | *www.tcl.tk* |
| **Expect**<br>*Alert: Version 5.44 of Expect has been recalled due to unresolved problems. Please use 5.43.0.* | **5.43.0** Stable | *expect.nist.gov* |
| **Python**<br>*see http://www.vex.net/parnassus/ for Python resources* | **2.4.3** Stable<br>2.5 Alpha | *www.python.org* |
| **PHP**<br>*see http://pera.php.net and http://pecl.php.net for additional resources* | **4.4.2**Stable<br>*This release address a few small security issues, and also corrects some regressions that occurred in PHP 4.4.1.*<br>*All PHP 4 users are encouraged to upgrade to this release.*<br>**5.1.4** Stable | *www.php.net* |
| **MySQL**<br>*see http://dev.mysql.com for associated downloads such as the MySQL administrator and the MySQL Query Browser* | **5.0.21** Generally Available<br>**5.1** Beta release | *dev.mysql.com* |
| **Apache Tomcat** | **5.5.17** Stable | *tomcat.apache.org* |
| **Apache httpd** | **2.2.2** Stable | *httpd.apache.org* |

*Since the previous edition of "Of Course", there has been a major new release of Apache httpd – the 2.2 branch, now at release 2.2.2. Because the httpd web server software is so much at the heart of web sites, and because the previous versions have been very stable indeed, we have seen no great rush to upgrade from releases 2.0 and even 1.3. We have, however, used Apache httpd 2.2 during some training courses and found it to run reliably.*

*Notably, Apache 2.2 now ships with a mod_proxy_ajp module that removes the need for users who require to talk to Apache Tomcat to download and install the jk connector; in effect it's now built in. This is an extremely positive move because the jk connector required to be compiled for exactly the right version of httpd and it was notoriously difficult to synchronise binary distributions of httpd and jk.*

*Some of the other changes to Apache httpd relate to busy sites. There's a mod_proxy_balancer that allows requests to be forwarded to a bank of other servers to do the real work. Each of the workers can be graded according to what proportion of the traffic is to be passed to each, and that proportion can be based on the number of requests passed on, or the relative traffic level in bytes. Improvements to mod_cache and related modules have seriously improved caching robustness and productivity. A graceful-stop signal has been added so that httpd can be instructed to refuse further requests, but wait to shut down until all outstanding requests have been completed.*

## Where'd you get those biscuits?

*continued from page 11*

copies of her menu to hand; orders are taken mid-morning, and at lunchtime your fish (or alternative if you're not keen on fish) will be collected by one of the Well House Consultants' team just before the class breaks for lunch.

If delegates vote for a lighter option, we can provide a more traditional buffet lunch including salads, a selection of breads, fruits, and prepared snacks from a local supplier.

Kebabs, pizza, chicken and chips, and takeaways from McDonald's and the Indian and Chinese restaurants also feature occasionally on our training centre menu.

We'll always have snacks and beverages, including tea (regular, infusions and herbal), freshly ground coffee (regular and decaf), bottled water and a wide range of soft drinks on tap, and at no extra charge.

The tutor joins for lunch, and this often gives students the chance to ask any questions that might not be covered on the course, or to continue discussions that might be too narrowly focused to apply to the rest of the class.

When lunch is taken at our training centre, you'll often find delegates move on to spend a few minutes playing our appropriately named "Tomcat" pinball machine (set to free play), stretching their legs around the garden, or checking their email on our wireless Internet. A few minutes off topic from the course can do wonders for the concentration in the afternoon, and with just one course running at the centre at a time, there's not the traditional behind-the-scenes rush to get "Class A" back in before "Class B" breaks for lunch.

Oh, to answer some of your questions:

- Lotus Caramelised Biscuits, the original caramelised biscuit produced by Lotus bakeries, Belgium. Available through Garraways (*www.garraways.co.uk*). £8.95 for a box of 300.
- Any of the coffee beans from The Roast and Post Coffee Company (*www.realcoffee.co.uk*), but we especially enjoy Old Brown Java, 500g for £9.99.
- We prefer our salad dressing freshly made on the day. For a healthier option, we add cider vinegar and hemp oil to Good Seasons Italian Dressing mix, available from Skyco Online (*www.skyco.uk.com*). Each 0.07 oz sachet is £1.82.

# Managing images through PHP and a MySQL database

*If a picture is worth a thousand words, and a database can hold a limitless number of pictures, think how much this information would be worth to you.*

We've got a lot of pictures (and I mean a LOT of pictures!) on our web site, and each of them has its own URL such as *www.wellho.net/pix/oilseed.jpg* or *www.wellho.net/pix/commuters.jpg*.

These might look like they're simple images, but we've recently moved them all into a MySQL database, and all of the URLs point to a PHP script to retrieve the images when they're needed. Why have we done this?

First, to reduce the */pix/* directory on our server to a more manageable size.

Second, so that we can keep a piece of descriptive text with each image. This will allow us to go on and search our images, and supply alt tags for them automatically.

Third, so that we can monitor and control access easily.

*continued on next page*

# Managing your pictures using PHP and MySQL

*continued from page 13*

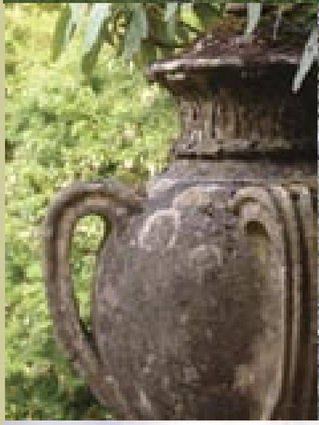
Hundreds of visitors arrive at our website every week looking at our pages that cover the saving and recovery of images from a MySQL database using PHP. Although you need to understand a number of different technologies to write code like this, the resulting application can be quite small. Here’s the code of our “feeder.php” script that does all the clever work:

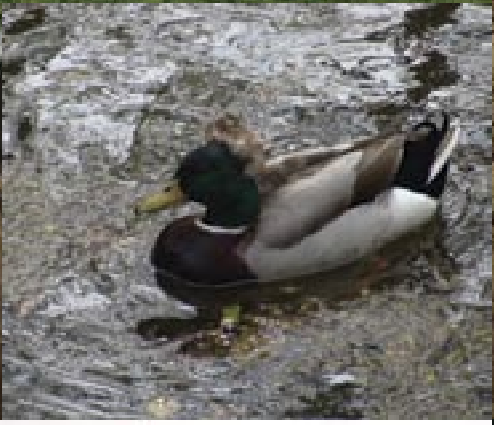
```
<?php

# Feeder script - images from our web library
# Collect file name, check that it’s not tainted

$error = 0;
$image = $_REQUEST[image];
$filename = “$image.jpg”;
if (preg_match(‘/^-A-Z0-9_/i’,$image)) {
    $error = 1;
} else {

# If a regular file image exists, read that and
# add it to the database

mysql_connect(“localhost”,”trainee”,”abc123”);
mysql_select_db(“wellho”);
if (file_exists($filename)) {
    $instr = fopen($filename,”rb”);
        $imagebytes = fread($instr,filesize($filename));
        $image4sql = addslashes($imagebytes);
    @mysql_query (“create table im_library (“.
          “iid  int primary key not null auto_increment,”.
          “filename   text,”.
          “imgdata   longblob,”.
          “descriptor     text)”) ;

    $q = “select * from im_library where filename = ‘$filename’”;
    $r = mysql_query($q);
    $rset = @mysql_fetch_assoc($r);
    if (is_array($rset)) {
          $where = “where iid = $rset[iid]”;
          mysql_query (“update im_library set imgdata = “.
                “\”$image4sql\” $where”);
    } else {
          $where = “”;
          mysql_query (“insert into im_library “.
          “(filename, imgdata) values (\””.
          $filename.  “\”, \””.
                $image4sql.  “\”) $where”);
    }
    @unlink ($filename);

# Otherwise, collect the image from the database
```

code at: http://www.wellho.net/resources/ex.php4?item=h999/feeder.php

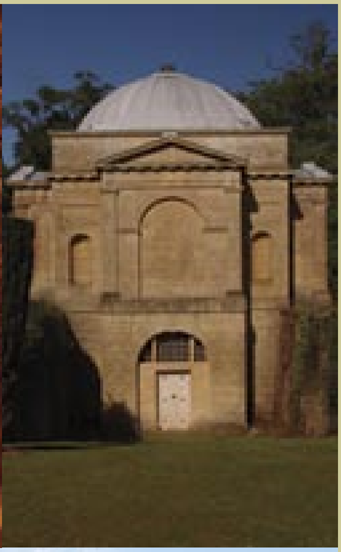
```
} else {
    $q = “select * from im_library where filename = ‘$filename’”;
    $r = mysql_query($q);
    $rset = @mysql_fetch_assoc($r);
    if (is_array($rset)) {
         $imagebytes = $rset[imgdata];
    } else {
         $error = 1;
    }
}
}

# if an error, collect the error page

if ($error == 1) {
    header(“content-type: image/png”);
    $high = 100; $wide = 400;
    $im = ImageCreate($wide,$high);
    $fg =ImageColorAllocate($im,255,255,128);
    $bg =ImageColorAllocate($im,0,64,128);
    ImageFill($im,0,0,$bg);
    ImageFilledRectangle($im,10,10,$wide-10,$high-10,$fg);
    $fontsize=20;

    $msgs = array(“Image missing:”,$filename);
    $ystart = 20;
    foreach ($msgs as $msg) {
    $bbox = imagettfbbox($fontsize,0,”ArialBold.ttf”,$msg);
    $xstart = ($wide-$bbox[2]+$bbox[0])/2;
    # $ystart = ($high+$bbox[1]-$bbox[7])/2;
    $ystart += $bbox[1] - $bbox[7];
    Imagettftext($im,$fontsize,0,$xstart,$ystart,$bg,
         “ArialBold.ttf”,$msg);
    }

    ImagePNG($im);
    exit();
    }

# Send out image

header(“Content-type: image/jpeg”);
header(“Cache-control: max-age=3600”);
print $imagebytes;
?>
```

*continued on page 19*





# *Why Wiltshire?*



You may wonder why we've chosen to have our training facilities in Melksham, Wiltshire.

Actually, we could be based anywhere in the UK, and still some 80% of delegates on our courses would have to stay over. That's how niche subjects like Python and Tomcat are, and how far people will travel for our Perl, PHP and MySQL courses.

Wiltshire is much more central than you might anticipate, as it is just 100 miles from London, easy to get to from the Midlands, Wales and the North, and with always-improving air links into Bristol from Scotland, Ireland and mainland Europe.

Our courses run at a quiet and spacious centre in the market town of Melksham, an ideal venue for concentration away from your office, yet on line and within easy reach as is needed.

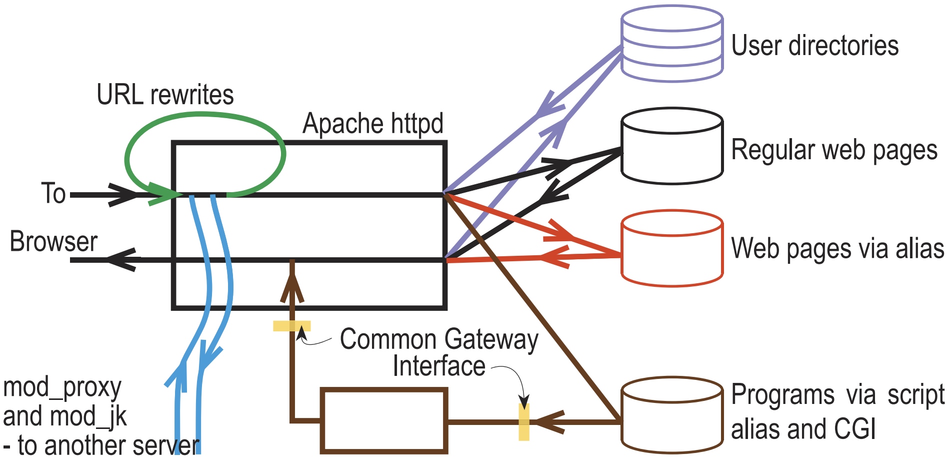
On long summer evenings, there's plenty to see as it's central to a vast tourist area encompassing Bath, Salisbury, Bristol and the Cotswolds. Perhaps you'll enjoy Wiltshire so much that you'll bring your family back for a long weekend to explore further.

# Exploring Apache httpd facilities

Apache httpd is a piece of web server software. At its simplest, a user connects to it and asks for a particular page by name. That page name is then added on to a directory path called the *DocumentRoot* and the resultant file is sent back to the user.

A user would type in the url "http://www.wellho.net/course/lbfull.html".

The browser strips off the "http:" bit (that's the language that it's to use to talk to the server) and the domain name "www.wellho.net" (that's the bit that tells it which system to ask) and it connect and sends `GET /course/lbfull.html`.



At the server, the document root is added in; if the document root was */home/wellho/htdocs*, then the full name of the file to be passed back would be */home/wellho/htdocs/course/lbfull.html*.

But system administrators often want to serve more than a simple subtree.

For the simplest of web sites, serving each URL with its own file it attractively simple but typically users want to run programs, forward requests to other servers, or use a single page to feed many different URLs. And Apache httpd provides facilities for all of these.

## Aliases

An Alias lets the server administrator graft a different part of the file system on to the tree under the DocumentRoot. It's rather like a symbolic link – a "see instead" type instruction. If we were to configure `Alias /icons/ "/usr/share/apache2/icons/"` then any request to a URL that started "http://www.wellho.net/icons/" would be redirected to a file in */usr/share/apache2/icons*.

Aliases are commonly used for standard Apache resources such as Icons and the Apache httpd manual, which you don't want cluttering up your own web tree.

## ScriptAliases

A ScriptAlias is like an Alias, but, additionally, any file found at the new location is executed via the common gateway interface (CGI), and the result of running it is passed back rather than the file itself. In other words, you use ScriptAliases to point to a directory of programs you may want to run. Here's a sample configuration line:

```
ScriptAlias /cgi-bin/
"/home/wellho/cgi-bin/"
```

Although normally used with a directory named "cgi-bin", you can use a ScriptAlias to turn any of your web directories into a directory of programs. But do note it's much safer to have a separate cgi-bin type directory than to rely on a file extension such as *.cgi*.

## User Directories

If you have `mod_userdir` configured in to your server, then any urls of the form "~username" will be redirected to a *public_html* directory in that user's own area. This is a very useful way of providing each individual with his or her own web space on an intranet system. For instance, if Lisa's home was */users/lisa* then "http://www.wellho.net/~lisa/whm/overallplan.html" would be pointing to the file */users/lisa/public_html/whm/overallplan.html*.

## Rewriting URLs

With `mod_rewrite`, you can have many URLs all pointing to the same resource, and you can rename individual resources if you like. You'll see an example in our PHP and MySQL sample script on pages 14 and 15 in this newsletter, where we forward all requests for *.jpg* images in the */pix* directory to a database of images. This allows us to track requests on the fly and rapidly spot external links in to our images.

`mod_rewrite` is also used on our web site for on-site training quotations. If you visit page *http://www.wellho.net/quote/CH.html* (for a quote on a course given in Chester), you'll actually be referred on to a script that covers all 124 postcode areas in the UK.

## Proxying

All the examples we've seen so far have diverted the request within Apache httpd. In other words, httpd has handled the requests in full, but has called up different resources to do so. `mod_proxy` and `mod_jk` both allow you to take a URL and refer it on to a different web server.

If `mod_proxy` intercepts a request, it causes Apache httpd itself to turn into a client and make a request of another server, using the same http protocol. Most commonly, this proxied request might be to a different server, such as an Apache Tomcat Server running Java applications.

`mod_jk` also forwards a request, but in a protocol known as Apache-Java Protocol (AJP). It's more efficient and more flexible than proxying, but is a little harder to set up, and has required very careful coordination of software release numbers between the various components (until the very latest versions of Apache).

We cover the configuration of both httpd and Tomcat on our Deploying Apache httpd and Tomcat course, and we also cover httpd configuration on our Linux Web Server and Deploying LAMP courses.

## Managing your pictures

*continued from page 15*

That's the complete script. The only thing I have changed is the account name and password for the MySQL database.

Apache httpd's mod_rewrite has been used to allow us to continue to write code as is the image really is in the */pix/* directory, but then change the URLs to references to the script. Here are the two vital lines from the *.htaccess* file:

```
RewriteEngine On
RewriteRule ^(.*)\.jpg /pix/
feeder.php?image=$1
```

To add a new image to the database, I just drop the *.jpg* file in the */pix* directory, and the feeder script takes care of moving it to the database when it's first called up, deleting the original from the directory.

We've also added code that will give you an image telling the name of the image that's missing if you call for an image that exists in neither the database nor a plain file. That's a great help as we develop pages, and it gives us an obvious alert when there's a problem. It's best not to throw out such a glaring error page if you're liable to get a lot of missing pages on your site; we don't.

If you want to run this code, just visit some of our Wiki pages such as *www.wellho.net/share/melksham.html* where all the images are saved on this database, as are many of the images in my Horse's Mouth blog at *www.wellho.net/horse/*.

If you want to learn how to do this sort of thing and you're new to PHP, our public PHP course runs at our training centre every two months. If you already know some PHP, but would like to spend a day with us going through subjects such as graphics in PHP, MySQL blobs and http headers, let us know and we can set up a day under our "extra" program.

# Various MySQL clauses and their uses

In MySQL, the simplest of SELECT commands sends the whole contents of a table to the client.

That’s exactly what we want on some occasions, but falls far short of other more sophisticated requirements, which can be met using a variety of additional keywords and clauses.

The extra clauses and keywords have to be specified in the right order too. Here’s an example showing most of the clauses in use and in the right order (I keep this example on our web site solution centre to that *I* can refer back to it too!)

```
SELECT ‘Name of Estate Agency’,
         ‘Estate Agent\’s Town’,
         ‘Estate Agent\’s Phone’,
         ‘AID (key)’,
         ‘count of matches’,
         ‘average price’,
         ‘lowest price’

         UNION

(SELECT agent, town, phone, aid, count(asking),
     avg(asking) AS avga,
min(asking)
         INTO OUTFILE “/tmp/weekend”
         FIELDS TERMINATED BY “,” ENCLOSED BY ‘”’
         FROM agents RIGHT JOIN sales ON aid = agid
         WHERE asking < 480000 AND locate != “Westbury”
         GROUP by aid
         HAVING count(asking) > 1
         ORDER BY town, avga DESC
         LIMIT 4)
```

So what are all these various clauses?

- After the word SELECT, we specify which columns from the data we want to have returned, including calculated fields using MySQL functions. Some of these may be given aliases with an AS clause so that we can refer to them again later in the command.
- An INTO OUTFILE command allows for the output to be redirected to a file direct from the MySQL daemon, and the FIELDS clause allows us to choose between a tab separated file, a CSV file, and other options
- The FROM clause tells us the table(s) on which to run the query, and how they should be linked together (joined)
- The WHERE clause lets us select which or the incoming (raw) rows are to be a part of the output
- The GROUP BY clause lets us collect a whole series of incoming (raw) rows into a composite row on the output. It lets us return that there are 10 houses for sale in Melksham rather than listing them all out, for example
- The HAVING clause lets us select which of the composite rows are to be output
- The ORDER BY clause controls the order in which the results are output
- The LIMIT clause controls the maximum number of results you want
- The UNION allows you to output one result set followed directly by another in the same result set; in this example, we’ve used it to title the file.

Below are the results from running that query on some appropriate tables.

```
Name of Estate Agency”,”Estate Agent’s Town”,”Estate Agent’s
Phone”,”AID (key)”,”count of matches”,”average price”,”lowest price”
“”,””,””,””,”2”,”370000.0000”,”275000”
“Kavanaghs”,”Melksham”,”01225 706860”,”10”,”3”,”219166.6667”,”195000”
“Jayson Kent”,”Melksham”,”01225 707798”,”4”,”3”,”175816.6667”,”127500”
“DK Residential”,”Trowbridge”,”01225 759123”,”3”,”2”,”343500.0000”,”222000”
```

# Schedule

## Perl

**Perl Programming**

*5 days* £1210

Covers the major features of the language and their use for newcomers. This course now runs in conjunction with Learning to Program in Perl and covers version 5, with footnotes where appropriate to tell you about Perl 6.

*12-16 June 2006*
*11-15 September 2006*
*4-8 December 2006*

**Using Perl on the Web**

*2 days* £550

This advanced course takes the Perl programmer through to making full use of Perl on the Web. CGI, mod-Perl, standards. Sessions, cookies, content management, etc.

*10-11 July 2006 (option)*
*12-13 October 2006 (option)*
*14-15 December 2006 (option)*

**Perl for Larger Projects**

*3 days* £770

This advanced course takes the Perl programmer through the facilities provided in the language which make it particularly suitable for larger projects. Objects, huge data, SQL databases, XML, efficiency and other topics.

*9-11 October 2006 (option)*
*11-13 December 2006 (option)*

## PHP and MySQL

**Technology for PHP**

*1 day* £330

Overviews the technologies that you'll need to make good use of PHP. A lead-in day for our PHP Programming course.

*4 September 2006*

**PHP Programming**

*4 days* £990

PHP is a scripting language used to add executable content to Web pages. This course covers the fundamentals of PHP, such as linking a series of forms together into a session, maintaining state, cookies, security and other matters. We also cover PHP's interfacing to relational databases, with a particular study of MySQL.

*3-6 July 2006*
*5-8 September 2006*
*6-9 November 2006*

**Object Oriented Programming with PHP**

*1 day* £330

This advanced PHP course introduces you to object oriented programming principles, and teaches you how they are implemented in PHP.

*7 July 2006*
*10 November 2006*

**MySQL Relational Database**

*2 days* £550

Structure, installation, administration, database design and programming.

*17-18 July 2006*
*16-17 October 2006*

## Apache httpd and Tomcat

**Deploying Apache httpd and Tomcat**

*2 days* £550

How to source, install, configure and support Apache and Tomcat for the deployment of Java web applications.

*8-9 June 2006*
*17-18 August 2006*
*26-27 October 2006*

## Python

**Python Programming**

*3 days* £770

This Python course starts at the fundamentals, then studies collections, functions, modules and packages, and the use of and creation of data types, plus an overview of OO principles and an introduction to development environments and resources.

*5-7 June 2006*
*14-16 August 2006*
*18-20 October 2006*

## Tcl

**Tcl Programming**

*3 days* £770

Fundamentals of the Tcl programming language including Expect, libraries and regular expressions. Further topics covered on the course include writing your own commands (procs), packages, libraries, namespaces and other - you'll leave the course with a thorough grounding in Tcl.

*9-11 October 2006 (option)*
*11-13 December 2006 (option)*

**Tcl - the Tk Toolkit**

*2 days* £550

Covers widgets, geometry managers and event handlers of the Tk GUI so you can write and maintain graphic applications.

*12-13 October 2006 (option)*
*14-15 December 2006 (option)*

## General

**Regular Expressions**

*1 day* £330

Dedicated to pattern recognition and matching, dependent on recognizing sequences of characters.

*13 October 2006 (option)*
*15 December 2006 (option)*

**Deploying LAMP –**
**Linux, Apache, MySQL, Perl / PHP**

*4 days* £990

Installing and supporting applications under open source technologies.

*9-12 October 2006 (option)*
*11-14 December 2006 (option)*

Or, take them individually:

*Linux Basics* *day 1 (£330)*
*also on 10 July 2006*
*Linux Administration* *day 2 (£330)*
*also on 11 July 2006*
*Linux Web Server* *days 3&4 (£550)*

**Programming in C**

*9-10 October 2006 (option)* £550
*11-12 December 2006 (option)* £550

**C++ for C Programmers**

*12-13 July 2006* £550
*11-12 October 2006 (option)* £550
*13-14 December 2006 (option)* £550

**Ruby Programming**

*10-11 July 2006 (option)* £550
*12-13 October 2006 (option)* £550
*14-15 December 2006 (option)* £550

*Prices valid for bookings confirmed by 31 August 2006. See Web site for US dollar and Euro prices. Prices do not include VAT. Book via: http://www.wellho.net/*

# Quickly check if value “x” is in list “y”

Have you ever written a piece of code where you need to check whether a value is present in a list? The traditional way is to use a `for` loop to check each element in turn. That can be done in almost any language, and is quite routine to write. But the code may be a little long, and it’ll run very slowly once you have a lot of checks to make. In our web site generation script, written in Perl, we’re making nearly 200,000 values against a list of over 200 words, and that’s simply not feasible with the “bull at a gate” approach.

**Using hashes**

Both Perl and Python support unordered collection variables, where elements can be referenced very quickly by a key. These variables are known as hashes in Perl, and as dictionaries in Python.

We set up a hash where the keys are the values that we want to check against, and each has a value “1” placed in there so that it can return a true value when tested. Here’s my Perl code:

```
while (<DATA>) {
        chop;
        $common{$_}=1;
        }
```

Later on, when I want to test a value, I can simply write:

```
unless ($common{$_}) {
```

to specify a block of code to be run if the word is not in the list.

In Java, a special class (a HashSet) is provided in *java.util* which you can use to store keys, in essence without values, and check efficiently whether a key then exists.

**Python’s `in` operator**

The specifiers of Python have gone one step further and have provided you with an “in” keyword where you can simply write

```
if trythis in wordlist:
```

where `wordlist` is a list or tuple of elements to be checked, and `trythis` is the test value.

For a really heavy job, it’s still worth using a dictionary for efficiency’s sake but the one-liner, as ever, leaves me in great admiration for the team that specified the language.

We use this technique to check all the meta keyword tags in our pages to ensure that we’re not asking to be indexed on 200 very common words which are unlikely to be relevant search terms.

Words like:

morning
evening
person
afield
offices
commute
group
range
links
needs
pound
bring

money
price
important
selected
companies
computing
excellent
further
particularly
concluding
require
requirement

computer
looking
look
together
keyboard
screen
some
also
own
provide
provides
staff

house
center
centre
while
answer
not
subject
subjects
working
worked
system
systems

give
giving
always
sales
number
whilst
longer
based
take
has
initial
initially

## Iterators – expressions that change each time you call them

If you’re programming and you write the same expression into your code twice without changing any of the variables, you’re simply writing the same expression twice, right? For example, here’s a piece of Perl code that exits if `$userval` is 0, but adds it in to a total and keep going if it’s not zero.

```
if ($userval == 0) {
        print “job done\n”;
        exit;
        }

$total += $userval;
```

If you’re reading from a file, though, things are different. Refer to the same file handle twice, and you’ll get the next line read in each time, thus:

```
if (<FH> == 0) {
        print “job done\n”;
        exit;
        }

$total += <FH>;
```

will read from a file handle twice. If the first line read is zero, the program exits. If the first line was not zero, a second line is read and that value in that line is added to `$total`.

Why the difference? Because the read from operator – <> in Perl – is what we call an iterator, and each time we refer to it, it moves on to the next value.

This one often catches new programmers. The solution is to read the value just once and save it into a variable, thus:

```
$userval = <FH>;

if ($userval == 0) {
        print “job done\n”;
        exit;
        }

$total += $userval;
```

# Calendar

Public scheduled courses are shown in tan. Open dates indicated as a guide only and are reserved for private, on-site courses. If you have three or more students and would like a course custom-tailored to your needs, please phone us to make sure the dates are still available. The office is closed for Christmas and Easter holidays.

## JUNE 2006

| | | | | |
|---|---|---|---|---|
| 29 | 30 | 31 | 1 | 2 |
| PYTHON<br>5 | 6 | 7 | APACHE/TOMCAT<br>8 | 9 |
| PERL<br>12 | 13 | 14 | 15 | 16 |
| 19 | 20 | 21 | 22 | 23 |
| 26 | 27 | 28 | 29 | 30 |

## JULY 2006

| | | | | |
|---|---|---|---|---|
| PHP<br>3 | 4 | 5 | 6 | PHP<br>7 |
| LINUX<br>10 | 11 | C++<br>12 | 13 | 14 |
| MYSQL<br>17 | 18 | 19 | 20 | 21 |
| 24 | 25 | 26 | 27 | 28 |

## AUGUST 2006

| | | | | |
|---|---|---|---|---|
| 31 | 1 | 2 | 3 | 4 |
| 7 | 8 | 9 | 10 | 11 |
| PYTHON<br>14 | 15 | 16 | APACHE/TOMCAT<br>17 | 18 |
| 21 | 22 | 23 | 24 | 25 |
| CLOSED<br>BANK HOLIDAY<br>28 | 29 | 30 | 31 | 1 |

## SEPTEMBER 2006

| | | | | |
|---|---|---|---|---|
| PHP<br>4 | PHP<br>5 | 6 | 7 | 8 |
| PERL<br>11 | 12 | 13 | 14 | 15 |
| 18 | 19 | 20 | 21 | 22 |
| 25 | 26 | 27 | 28 | 29 |

## OCTOBER 2006

| | | | | |
|---|---|---|---|---|
| 2 | 3 | 4 | 5 | 6 |
| OPTIONS WEEK<br>9 | 10 | 11 | 12 | 13 |
| MYSQL<br>16 | 17 | PYTHON<br>18 | 19 | 20 |
| 23 | 24 | 25 | APACHE/TOMCAT<br>26 | 27 |
| 30 | 31 | 1 | 2 | 3 |

## NOVEMBER 2006

| | | | | |
|---|---|---|---|---|
| PHP<br>6 | 7 | 8 | 9 | PHP<br>10 |
| 13 | 14 | 15 | 16 | 17 |
| 20 | 21 | 22 | 23 | 24 |
| 27 | 28 | 29 | 30 | 1 |
| PERL<br>4 | 5 | 6 | 7 | 8 |

CUSTOMISING YOUR COURSE NEEDS

## LAMP: What's on your wishlist?

On tailored LAMP deployment courses, we'll go right through from Linux installation (RedHat or SuSE) via httpd, PHP and Perl to configuring MySQL accounts as needed by the customer.

And if the delegates need us to cover programming in the various languages deployed, we can do that too.

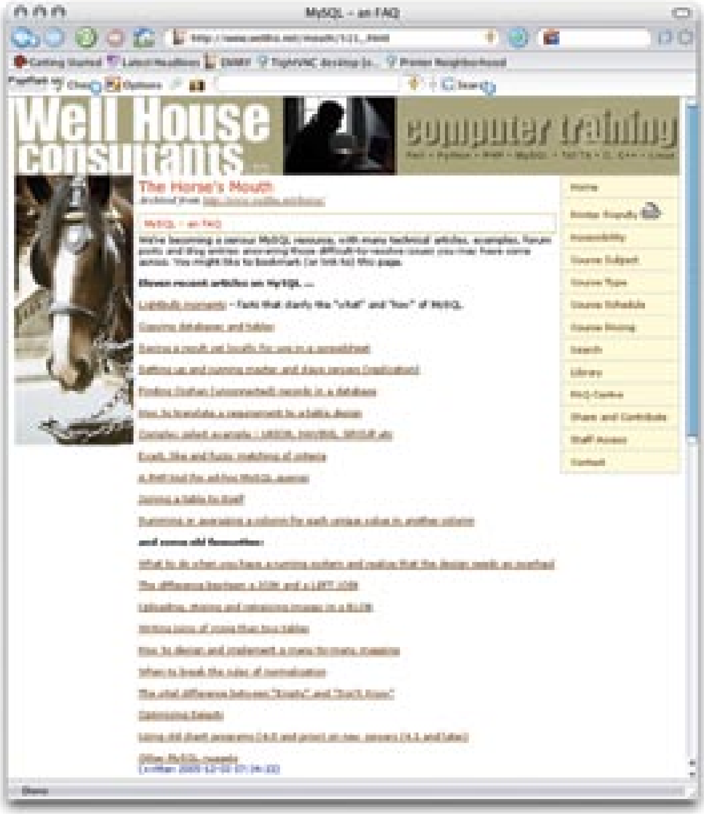
Especially with deployment courses, every customer's need is a little bit different. That's why our standard LAMP deployment week is also offered as three individual courses of 1, 1 and 2 days.

We encourage you to email us if you're interested in deployment training. It is set up to meet your needs. The next available LAMP course starts on 10th July under our 'options' program.

## Resource for tips and techs

We've added a lot of tips and techniques to our web site resources recently. Have a look at *www.wellho.net/mouth/515_.html* where you'll find links to articles such as:

- Lightbulb moments – facts that clarify the "what" and "how" of MySQL
- The difference between a JOIN and a LEFT JOIN
- The vital difference between "Empty" and "Don't Know"
- Exact, like and fuzzy matching of criteria
- Finding Orphan (unconnected) records in a database
- Writing joins of more than two tables
- Using old client programs (4.0 and prior) on new servers (4.1 and later)
- Uploading, storing and retrieving images in a BLOB
- Copying databases and tables
- Saving a result set locally for use in a spreadsheet
- Setting up and running master and slave servers (replication)
- How to translate a requirement to a table design
- Joining a table to itself
- How to design and implement a many-to-many mapping
- When to break the rules of normalisation



The Horse's Mouth

MySQL - an FAQ

We're becoming a serious MySQL resource, with many technical articles, examples, forum posts and blog entries answering those difficult-to-resolve issues you may have come across. You might like to bookmark (or link to) this page.

**Eleven recent articles on MySQL ...**

Lightbulb moments – facts that clarify the "what" and "how" of MySQL

Copying databases and tables

Saving a result set locally for use in a spreadsheet

Setting up and running master and slave servers (replication)

Finding Orphan (unconnected) records in a database

How to translate a requirement to a table design

Complex select examples - UNION, DISTINCT, GROUP etc

Exact, like and fuzzy matching of criteria

A shell tool for ad-hoc MySQL queries

Joining a table to itself

Summing or averaging a column for each unique value in another column

**and some old favourites:**

What to do when you have a running system and realise that the design needs an overhaul

The difference between a JOIN and a LEFT JOIN

Uploading, storing and retrieving images in a BLOB

Writing joins of more than two tables

How to design and implement a many-to-many mapping

When to break the rules of normalisation

The vital difference between "Empty" and "Don't Know"

Optimising Selects

Using old client programs (4.0 and prior) on new servers (4.1 and later)

Other MySQL resources

## Options Programme

Even though our more popular courses are Perl, Python and PHP, we can provide you with training on less-booked Tcl and Tk, Ruby, C and C++ through our Options programme. Under the programme, a week is set aside every quarter when we'll run a public course on whichever course(s) we get confirmed bookings first.

The first Options weeks are the weeks commencing 10th July (already assigned with Linux and C++) and 9th October 2006. See *www.wellho.net/net/options.html*.

Options courses are held at our Melksham, Wiltshire training centre, in classes with a maximum of seven delegates. Prices are £330.00 for a one-day course, £550.00 for two days, £770.00 for three days, £990.00 for four days and £1,210.00 for five days, all prices exclusive of VAT.

### Your Contacts at Well House Consultants:

| | | |
|---|---|---|
| **Administration** | Lisa Ellis | *lisa.ellis@wellho.net* |
| **Technical** | Graham Ellis | *graham@wellho.net* |
| **Accounts/Contracts** | Leah Davies | *leah@wellho.net* |
| **Phone** | +44 (0)1225 708 225 | 0845 8382 404 *(Lo-call)* |
| **Fax** | +44 (0) 1225 707 126 | 0845 8382 405 *(Lo-call)* |
| **Web site** | *http://www.wellho.net* | |
| **Postal Address** | 404, The Spa, Melksham, Wiltshire, England SN12 6QL | |

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

You'll usually reach us at the weekend and in the evenings, too, as phone messages and faxes are emailed to us automatically. 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.