



informit.com

Advanced Search | Search Help

Keywords

- Home
- Bookstore
- Safari Bookshelf
- Articles
- Reference Guides

Login My Acc

Site Help

Home > Articles > Networking and Communications > Security > Interview with John Coggeshall, Internationally Respected Developer and Technical Consultant for Zend

Contents

- 1. CIPS Connections

Related Book



PHP 5 Unleashed \$44.99 (Save 10%)

You May Also Like

- 1. [Financial Advice for the Suddenly Unemployed](#)  
By [Edie Milligan](#)  
Jul 26, 2002
- 2. [Consulting: Prerequisites for Success](#)  
By [Robert Bacal](#)  
Aug 30, 2002
- 3. [Configuring the PIX Firewall for SSH \(Secure Shell\)](#)  
By [David Chapman, Jr](#)  
Feb 15, 2002

[See All Related Articles](#)

Search Related Safari Books



Search electronic versions of over 1500 technical books for **ibarak**.

SEARCH

# Interview with John Coggeshall, Internationally Respected Developer and Technical Consultant for Zend Technologies

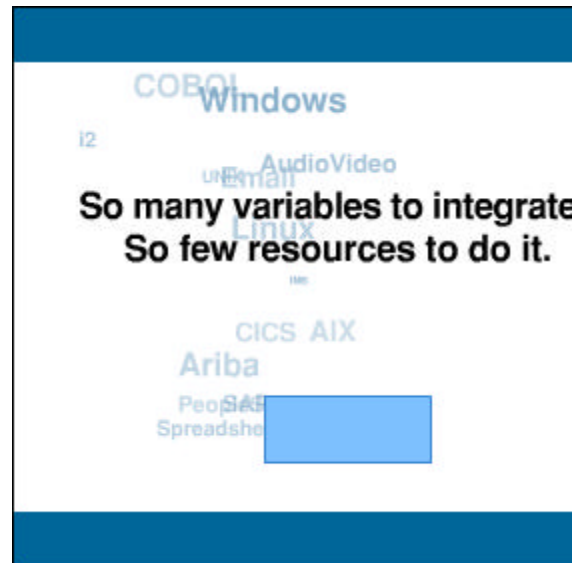


By [Stephen Ibaraki](#).  
Article is provided courtesy of [Sams](#).  
Date: Mar 18, 2005.

Stephen Ibaraki interviews John Coggeshall, a Technical Consultant for Zend Techn and the author of two published books and over 100 articles on PHP technologies. J discusses the most common PHP problems and solutions, and the five more importa he's learning about programming.

Other Articles By [Stephen Ibaraki](#).

John Coggeshall is a Technical Consultant for [Zend Technologies](#) where he provides professional services to clients around the world. He got started with PHP in 1997 and is the author of two published books and over 100 articles on PHP technologies. John also is an active contributor to the PHP core as the author of the tidy extension, a member of the Zend Education Advisory Board, and a frequent speaker at PHP related conferences worldwide. His web site, <http://www.coggeshall.org/> is an excellent resource for any PHP developer.



### Discussion:

**Q:** John, thank you for sharing your authoritative views on PHP with our audien

Ads by Goooooogle

**[Find Computer Books](#)**

Select from millions of Titles! Free Shipping & Easy Store Returns  
[chapters.indigo.ca](http://chapters.indigo.ca)

**[Books](#)**

Find Gifts For Mom from 100s of Retailers. Shop Now!  
[shopping.sympatico.ca](http://shopping.sympatico.ca)

**[Save upto 40% on Books](#)**

at Amazon.ca Free Shipping on orders over \$39  
[www.amazon.ca](http://www.amazon.ca)

**[Vancouver Tech Jobs](#)**

Over 1100 Tech Jobs in Vancouver. SW, HW, QA, IT, Web, Management.  
[www.TechJobsVancouver](http://www.TechJobsVancouver)

A: Of course, I'm more than happy to be involved.

**Q: Describe your experiences at Camp CAEN and how you got started with PHP**

A: CAEN didn't really shape my involvement with PHP, but it was where I was first intr the language. It was back in 1997 or so, and while I was in the CAEN labs, a friend of pointed me to php.net for the first time. I started playing around with the PHP 3.0 beta soon downloaded a copy of MySQL as well. Once I saw how easy it was for me to wri database-driven web applications, I was hooked and never looked back.

**Q: How about your time at Kettering and with Delta Tau Delta?**

A: College, in general, was where things really started taking off for me and PHP. Bet CAEN and my freshmen year at Kettering, I had become a frequent poster to the php-mailing list where I would constantly be answering questions about PHP-related things from all the time that I spent there that I got noticed as an author. And by the time I joi Tau Delta at Kettering, I had started my writing career. It was a very good call for me professionally, because I could fit it in around my studies without killing myself. As I cc through college, I found myself complementing everything I had learned in the field wi studies. It made for a really killer combination.

**Q: From your time as author of Zends Code Gallery Spotlight, what are the ten n common PHP problems and can you provide an overview of their solutions?**

A: The ten most common PHP problems and solutions? That's almost an article within think. Still, looking back at the Spotlight column, topics such as E-mail, HTML Forms, Validation, and generation of things like PDF documents were all very popular column get the occasional e-mail about them from time to time and most were written almost 4 ago. I think thats because I always tried to address a very specific topic and to give a l back solution to it. I don't think I can just pull out of the air ten problems that everyone encounters today, because PHP is now used in so many more ways than it used to be then.

**Q: Can you comment on your articles featured in the the php|architect publicati**

A: I've written a few articles for php|architect. I think three features to be exact. The fir article about SQLite, a great new relational database extension for PHP, which doesn' on a client/server model. That's really useful because it gives you a way to write applic which are database dependent without making a heavy database like MySQL or Orac requirement to use it. Basically, it's like using SQL to work with flat-file databases.

I believe it was my second piece which was featured on the cover the IntSmarty proje IntSmarty is a technology to solve the problem of multi-lingual web site development b harnessing the Smarty templating engine for PHP. It's different from other solutions be allows you to put logic into your translation tables so when you do a translation, you c: do it properly. This is most valuable when doing things with numbers, I have X apples instance. What if X is zero? The phrase then should be, I have no apples. If X is 1 the phrase is, I have 1 apple. Even in English alone we need to apply logic to the actual la our application, and IntSmarty allows us to do that.

The third article I published with php|architect was an overview of PHP 5. It discusses awesome new technologies you can find in it, such as the drastically improved object the new PHP 5 extensions that improve PHP's support for everything from the everyd: page to Web services.

**Q: Discuss your involvement with and some tips for: PHP Tidy, BLENC, and Pur**

A: These are all projects of mine which are at varying degrees of stability. Tidy, BLENC Pure are all extensions for PHP 5. The tidy extension is a part of the standard PHP dis and gives you the ability to manipulate HTML documents in a way never before availa PHP. For instance, with tidy you can automatically tidy up user-submitted HTML and c

errors like missing closing tags, unquoted attributes, and more. BLENC stands for Blo encoded and is designed to encrypt a PHP script to keep unwanted eyes out of it. Although not the most secure approach, the primary reason I wrote it was for my work as a consultant. When you write code for a client you really don't want them messing around with it. If I manage to break it, they almost always come back to you wanting it fixed. I wanted a way to encrypt scripts that worked transparently to PHP so this is what I came up with. Purely for PHP 5. I'm still working on releasing basically it can be described as a transparent function cache for PHP 5. You give it the name of a function and it will automatically cache the results of a function call in shared memory, based on the parameters it was given. Then, next time you call that function with the same parameters, it'll simply return the result without executing the function for whatever length of time. This sort of partial-caching of select functions can improve the performance of an application.

**Q: What essential points can you share on topics ranging from Smarty to SQLite?**

A: Well, thinking of those two technologies specifically, the first thing that comes to mind is it's important to apply the right tool to the right job. Smarty is an excellent template engine for PHP, but it's not for everyone to use all of the time. Understanding the technologies for application development is more than just knowing how to code, but rather knowing enough about the strengths and limitations to pick the right tool for the job. For instance, although SQLite is a great technology it's not very well suited when you have to do a lot of writing to a database concurrently. Then again, you can also use it to quickly make tables in memory and perform queries against them very cool if you are taking data from a third-party non-relational source and need some quick analysis of it. Same thing goes for Smarty, although it is a template engine, you need to know when you really need a template engine and when it's more trouble than it's worth. Like I said, it's all about understanding the technologies.

**Q: Please comment on PHP Fundamentals with O'Reilly.**

A: PHP Fundamentals is a column that has been running for a very long time, but I have been too busy lately to do much with it. The concept behind it was to provide a ground up series of articles, teaching the basics of PHP for someone who has little programming experience. It started it has now moved into more advanced subjects such as databases and security. I try to write each article in the column in such a way that by the end of it you really understand the important concepts. So if you are just getting started, I would really recommend you click on it to get your feet wet.

**Q: Describe your most interesting projects for Zend and share five helpful lessons for our audience.**

A: Working with Zend has given me the opportunity to really work with a lot of high performance web sites, from the entertainment to the financial industry. I've also had a great experience being a member of the Zend Education Advisory Board which developed the PHP certification test. All of the projects have presented unique challenges and I really enjoyed the dive into them. Here are five things I'd have to say these:

1. Avoid 'whack-a-mole' problem solving

All too often when working with companies I see them trying to solve this problem one by one, and not looking at the big picture. If you are having trouble with your web application because of a bug, it's really worth the time to step back and make sure you are addressing the issue. I've seen too many people get into the mindset where they fix the bug and put in a band-aid fix, and then two days later the same bug crops up somewhere else. Then they do it all over again without ever really addressing the problem. I call this 'whack-a-mole' problem-solving because I always think of a mole game where the mole sticks his head out from a hole so you hit him with a hammer to see him come out of a different hole. That sort of problem-solving won't get you anywhere.

2. Obscurity is not security

This is another huge problem in PHP web sites that I've seen security through. Too many companies and too many developers write PHP applications which are insecure, or they make the scary assumption that just because a URL isn't linked anywhere on the web site that no one will ever know it's there. So they don't protect it. Another great mistake that I see web sites make is their blind trust and confidence in third party data. By this I mean taking a piece of data from a third party source (for instance, a form submission), and then throwing that data straight into your application without doing anything at all to make sure it's what you expected it to be. Not only does this sort of thing make your applications less reliable, but sooner or later, malicious users will figure out a way to take advantage of your neglect and will compromise your applications.

### 3. Cross site scripting attacks

This falls in line with trusting user data, but it's a big enough problem that it really deserves its own point. Cross-site scripting attacks have become a favorite weapon for malicious users to compromise the security of both companies and their clients. Client-side scripting attacks occur when your web site accepts a piece of unfiltered data from the user, stores it into the database, and then displays that data to other users. The problem here is that because the data was accepted originally without any filtering, it could contain anything including for instance, JavaScript code. When this data is displayed to other users, the JavaScript code executes and can do all sorts of things. Again, I cannot stress this enough, always filter and validate data taken from an outside source.

The second type of cross site scripting attack I've experienced occurs on the server side and is honestly probably much worse than the client-side version. In PHP, these are always related to URL wrappers available in PHP. For those who might not know, URL wrappers are the ability to open a remote file resource using standard file functions such as `fopen()`, `fgets()`, etc. Even though URL wrappers are very useful and powerful, failing to use them properly can land you in all sorts of trouble. The most common problem is when the remote resource being opened using this technique is determined by a variable which might be compromised by the user. For instance,

```
require_once($_LIBPATH/mylibrary.php);
```

As a malicious user, my goal here is to alter the value of `$_LIBPATH` from what it was to say, <http://www.coggeshall.org> instead. If I can do that, then PHP will open an HTTP connection to `http://www.coggeshall.org/mylibrary.php` and execute whatever PHP code it finds there on the local server. This sort of security hole gives a hacker the ability to execute anything they want on your servers. I actually encountered this on a very popular web site by setting two GET parameters in the URL of their pages I could have their servers download source code from my web site and execute it on theirs, including having complete access to their databases. This stuff, and people need to make sure they are aware of the potential vulnerabilities.

### 4. Measure twice, cut once

Carpenters around the world know that you always measure twice and cut once when you are building something. I wish more web developers thought in these terms when they went off coding as well. Architecture is worth the time early on to develop a plan and care how little time you might have to actually finish the project being worked on. Often in my job I see companies with a development staff which just started coding a web site without giving any consideration to the architecture they needed to build the site work. Without fail they always end up regretting it. Either the site ends up in a whack-a-mole mentality or it simply can't scale when it really starts becoming popular. It is always worth the time, early on, to think about what you have to build, before you build it, because you'll waste more time later trying to fix it and will still end up with an inferior product.

5. The Manual is your friend

This isn't really something I learned dealing with clients, but is, without a doubt thing you have to learn if you are going to be a serious PHP developer. PHP is source project, developed in large part by countless people volunteering their time. Beyond the application itself, there is an entire group of people who have same just to document PHP's abilities. Before jumping on a mailing list and asking question, or even worse, expecting the PHP community to have a free custom hot line, be mindful of this and use the resources we've provided you first. It's rare we people in the community are unhelpful mean jerks, but when someone obviously hasn't taken a single moment to investigate the solution themselves before asking someone else, it tends to wear on you. Reading a question where the poster has obviously done their best to get the answer themselves is infinitely more likely to get a truly helpful response than someone just wanting someone else to do the legwork for them.

Beyond that, while I am on the subject of the manual, let me also mention this: looking for a particular function to do something like word wrapping, text formatting, the odds are it already exists as an internal PHP function. It's always worth taking a look in the manual to make sure PHP can't already do what you need it to do before writing a custom function. Internal functions are easier to maintain and faster across the board.

**Q: What are your favorite conference topics? Can you give us a glimpse of what you're talking about in the future?**

A: I don't know if I have a favorite topic I like to talk about. Anything I'm speaking on is something that I find very interesting to begin with. So I guess my favorite topics are the ones that everyone else finds interesting too.

I'll be giving a number of talks this year: PHP 5 and Web Services, Migrating from PHP 4 to PHP 5, Smarty, PHP 5 / Java integration, and maybe even an Enterprise Architecture talk. All are all going to be great talks so I hope to see some of your readers there!

**Q: How will your book, PHP 5 Unleashed, contribute to the success of its readers?**

A: PHP 5 Unleashed is two years in the making and I put a great deal of my life into making it the best PHP 5 book it could be. It's 700+ pages, so beyond serving as an acceptable reference it is also completely packed with information on the entire range of PHP 5 technologies. Readers are going to love this book because it focuses on the practical. It is organized so that you can flip to a particular topic and find what you need quickly. My code examples are all done so that not only do they illustrate the concepts I'm trying to get across, but they might be useful to you just to cut/paste into your own applications. I also discuss a number of non-PHP 5-specific technologies such as WML, Web Services, Data Encryption and more. Even if you are still a PHP 4 developer, I think you'll find this book to be a nice transition.

**Q: What are your most treasured lessons you want to share on regular expressions?**

A: I definitely have a few things to say about regular expressions first, when it comes to writing them, 99% of the time if you are going to write them, use the PCRE library instead of the POSIX library (the preg\_\* functions instead of ereg\_\* functions). The PCRE is more supported and often just better at the job. Second, if you are the sort who has trouble writing regular expressions, I have two pieces of advice for you:

1. Don't use them.

Regular expressions aren't for every problem, and often they are considerably more complex than just parsing out some data using the other string functions in PHP. If you can do the same parsing of a string accurately using 2 or three lines of code you're probably better off doing it that way.

2. Find yourself a nice regular expression helper.

There are a number of programs out there which allow you to paste in a test string then write your regular expression. Good ones highlight the part your regular expression has matched so far, show you what matches have been made if you are trying things out, and all and all are incredibly useful.

**Q: Describe some of the ways to avoid common pitfalls when building large sites in PHP.**

A: As I said before, Architecture is the key. But when building a large site specifically, sure you are thinking in terms of scalability. Your sites should come out-of-the-box ready to be scaled from both the database and web server side of things. Understanding how to use technologies such as MySQL master/slave replication is important. If you are going to use sessions make sure they are stored in the database so your server farm can all get to them if for whatever reason you have to receive files from the user at runtime (for instance, to be able to upload files), those are likely best suited for a NFS mounted drive so there is no lag between when the user uploads and when each server in the farm can get at the user's image.

**Q: What are your suggestions for performance tuning PHP?**

A: I've written a number of articles on performance, but really the key to performance is understanding where things are slow in the first place. There are a number of options for trying to profile a web site, and personally I recommend Zend Studio. I know I work for Zend so maybe that endorsement doesn't carry a lot of weight, but I honestly think it is the best for PHP available. If you absolutely must go with open source, the Xdebug extension is a good call, although it's not anywhere as easy to use as Studio. Once you've identified what's slow there are a number of options to speed things up. Zend again offers a number of enterprise class tools for this if you are running large sites (namely Zend Platform), but if you are open-source Joe, you might want to look into open source alternatives like APC or Memcached. No matter which product you use, simply adding compiler caching is going to speed things up significantly. Beyond that, if your code can't be optimized any faster you can start caching slower function calls, and if possible, entire pages. If you do all of that, I think you'll find many of your problems are going to go away. (Not that it's always easy to do everything I said.)

**Q: Do you have additional books planned in the medium term?**

A: Yes, as a matter of fact I do. I'm currently getting started on my fourth book php|architect's Guide to Programming Smarty which is going to cover Smarty from front to back. As the book suggests, it's being published by php|architect and should be on the shelves (and perhaps in PDF form) sometime this year.

**Q: With your knowledge of key trends in Open Source, what developments should we be following and why?**

A: One thing that has really caught my eye and gets me excited is that we're getting to a world where an operating system doesn't matter. As browsers become more feature rich and technologies such as XUL emerge, full-fledged web based applications like an e-mail client or word processor won't be such a far-fetched reality anymore. We're starting to see it all things such as Google's Gmail, but I think that's only touching the surface of it all. I'm not going to say anything more yet, but I wouldn't be surprised if you saw things from me on this PHP in the coming year. From a more philosophical point of view, the Internet has changed from a focus on service and more toward information. As everything has matured it's not about the technology behind making a book store or auction site, it's about the database that site which adds the real value now. The fact I can order something from my computer is cool anymore -- it's expected what's cool now is the 10 quality reviews I can read about a book, and have someone show me other books which I'm probably also going to want to read. Services are going to allow us to start pulling these pieces of information together and combining them in ways never before possible, and that's really going to make for some cool things down the road.

**Q: John, can you provide some wide-ranging comments on topics of your choice?**

A: I see a number of different technologies and trends coming together which are likely to drastically change the way people think and live in general in the world.

The first is a no-brainer - the maturing of the Internet has forever altered our ability to access information and communicate with the world. Its more than just the Internet in general that I think what it has enabled is even more interesting.

#### **Social Software programs and Internet Identity:**

The first thing is the idea of Social Software, programs which allow you to share your thoughts and ideas with the world, meet new people, and keep in contact with old friends. Although I don't really think Social Networking sites have really matured yet (such as Orkut, Friendster etc.), you can start to see the business-centric ones such as LinkedIn take shape.

I also think this is where companies like Sxip, who are pushing for the adoption of an Internet Identity, will start to make real progress because the concept of Internet Identity is useful. However, I don't think anyone will be successful in making it a reality without piggy-backing on another social technology. In any case, as we move forward, I think you'll see these social networks not only become more useful but also much more integrated than they are today. Amongst each other as well as with related technologies. Although it's hard to predict exactly, I wouldn't be surprised if five years from now your phone's address book was filled with these networks and you had all of this information at your fingertips at all times. I think maybe our children will be using them to get that number for their hot date on Saturday night. I never know for sure. It's an interesting thought - I'm sure that being able to view the social network of the person our children are dating would offer some sense of comfort to us parents!

#### **Smart Phone:**

On the note of phones, another piece of the technology puzzle that I think will open up doors for technologies like PHP is the growth of the Smart Phone. When you blur the lines between computer and phone and start carrying around a wireless link to the Internet in your pocket (with a decent interface and bandwidth), there are a lot of really slick things you can do. As these Smart Phones become more affordable and accepted, I think they will without a doubt launch a whole new massive market for business and development online. It will enable you to have like the social-network in your pocket and I really think its a critical piece to things advancing down the road. I believe, ultimately, anything that puts more information in a digestible form at your fingertips is going to become an invaluable piece of society in general.

#### **Effects of technology on the family:**

As far as more personal aspects of things, I guess I can talk a little bit about them. To me I am a very happy father of a beautiful daughter -- her name is Diana Kathryn Coggeshall. When I think about the future of technology, I try to imagine how she'll use it someday years down the road. My Grandmother lived from the days of the horse and carriage to the moon and beyond; that has always been amazing to me. Thinking about a grandchild that happened in her lifetime makes me excited about all of the things that I'll see in mine. I think about how much Diana will see in her lifetime on top of that, I'm just blown away. It makes me really appreciate the opportunities I have in shaping the technology of the future. That said, I actually still do love to program, although to be honest, not really so much anymore. I would much rather be writing a new extension for PHP using C than using a new extension within PHP. Maybe it's just going back to my roots, but there is just something about programming that I find stimulating. In either case, I find it all very exciting and I love being a part of it.

#### **Zend:**

Last but not least I guess I can talk a little bit about the future of Zend. I haven't been working for Zend for very long, however if I had to choose a single word to describe the future of the company, I'd pick "bright". They have been incredibly good to me in my short time there. Although I cannot discuss details, I can say that I know first hand their devotion to the product.

for success in making PHP the single best language for web application development world. It's an exciting place to work and I'm very happy to be involved in the organizati

**Q: John, we look forward to seeing future contributions from you. Thank you ag your time, and consideration in doing this interview.**

A: Thank you for taking the time to interview me. It was a pleasure answering your qu and I hope you've found what I've said to be insightful.

### **Make a New Comment**

---

You must [login](#) in order to post a comment.

[About](#) | [Legal Notice](#) | [Privacy Policy](#) | [Press](#) | [Jobs](#) | [Write For Us](#) | [Contact Us](#) | [Advertise](#) | [Site Map](#)

© 2005 Pearson Education, InformIT. All rights reserved.  
800 East 96th Street Indianapolis, Indiana 46240

**informit network**