What on Earth is... PLONE?

Updating a website with fresh content is a pain – but if you give the job to inexpert users they might ruin your carefully-designed pages. Jono Bacon discovers a free system that will do the work for you.

This Plone thing sounds like some kind of wireless or VoiP handset. I'm guessing I'm wrong, unless this magazine is now *Wi-Fi World*...

We're not talking about telephones, don't worry. Plone is a powerful content management system, you know.

Content management system? That sounds less than thrilling. What does it mean?

Don't judge too hastily. A content management system (CMS) is a nifty piece of software that's designed to manage, process and effectively deal with content in different ways. In recent years, CMSs have attracted the attention of business, as they provide a means to manage large quantities of content in interesting ways. Traditionally, organisations have needed to have bespoke CMS systems specially written for them by software companies, but open source CMSs such as Plone provide a platform on which to build a tailored CMS and gain from the open source benefits such as access to source code and using the software free of charge.

So is a CMS a database, like *MySQL*?

No – a CMS provides much more. A CMS system gives the users tools to automate how information is managed. Let's assume that you run a website, and you want to post a list of events on it.

No doubt, when you add an event you'll want: (a) an easy and convenient way to add the information; (b) that the information will be displayed in the right place on the site; and (c) that the information triggers other types of content such as a press releases or a news items. A CMS helps you achieve all these things.

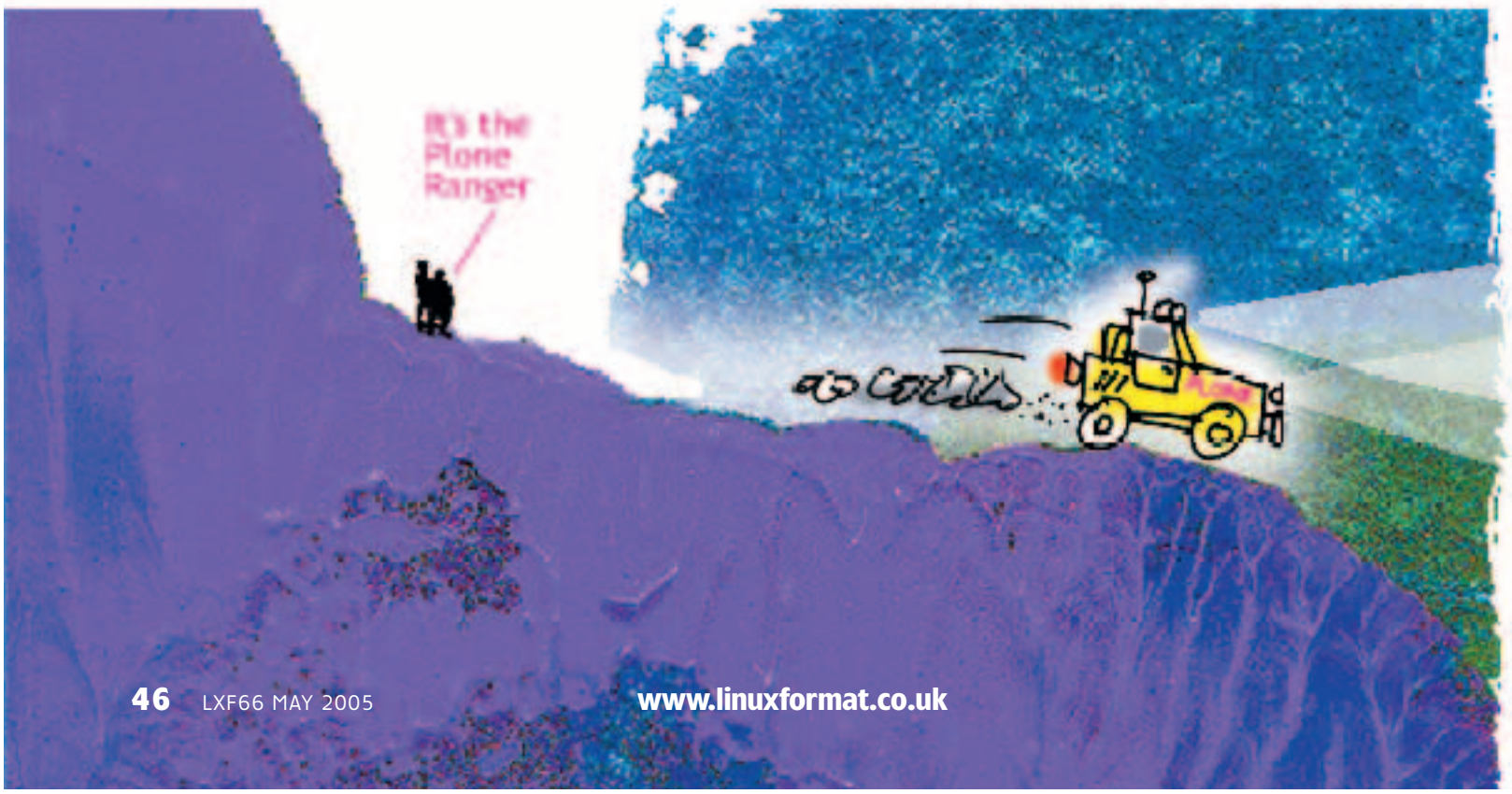
So a CMS is just a website?

Not exactly, no.

Huh?

Let me explain. Firstly, although most CMSs are web-based, not all of them are. Secondly, the web-based CMSs are not just websites, they're web *applications*, and are designed to give people the opportunity to add and manage information in a non-technical way.

For example, if you were to add an event to a non-CMS website, you'd probably need to add the



event details directly to the database and possibly edit some HTML or other static content. On a CMS, you could log into the system and fill the details into a form, which would then submit the information to the right place in the database. On a clever CMS, the event information could be sucked from another site, reformatted and then added to your database and displayed.

Why not just create a website that does all this?

The main thing to remember about a CMS is that it provides a pre-written website that provides you with many of the tools and features that you would otherwise have to code into a website yourself. In addition to this, a CMS often provides the ability to run new technologies and plug-ins that you may not have the time to investigate and implement yourself. This means that by using a CMS, you have the opportunity to keep up to date with the latest technology, and not have to constantly rewrite chunks of your website.

There are, however, a few limitations with CMSs. If you think of a CMS as a generic website to which you can add your own information, you are restricted by the way the developers of the CMS have decided to do things. This can involve limitations in how the content is stored or, more typically, limitations in how the information is displayed.

Some CMSs are very inflexible in this visual representation of the content, and getting around this has proved to be extremely difficult. Luckily, Plone is more flexible than these other CMSs and you can configure every aspect of how the website is displayed and managed.

How flexible?

Plone is a very, very capable open source CMS. The Plone team (a mixture of Americans, Norwegians and Brazilians), has developed it as a platform that can provide virtually unlimited flexibility in how you want a website to be set up. In some other CMSs, you are forced to use a particular template for structuring your site. You can change the images, fonts and colours and so on, but are still limited to where information is located and how it is accessed on the page.

Plone breaks these rules and gives you the chance to locate your information anywhere on your page. In addition to this, Plone is incredibly extensible.

Extensible? Sounds like more jargon...

Sorry: extensible just means you can add additional modules and features to the system for your specific needs. In many cases, these additions are limited by the constraints of the CMS in question. Within the Plone CMS, this extensibility covers wide ranges, and a huge range of vastly different Plone additions are available. On top of this, the in-built flexibility of Plone gives you the ability to use these additions in many different ways.

So give me some details – what kind of additions?

Anything that's related to website functionality. Web photo galleries, address books, weather information, RSS syndication tools, instant messaging, groupware or collaboration tools, navigation bars, document control... the list goes on. Plone also includes nifty features such as WYSIWYG editors for adding content to sites and tools to create specific types of content that your site can manage.

This all sounds interesting, but I'd be concerned that the design of my site would be compromised.

Plone includes a templating system that allows you to adjust the design of different parts of your site. Each of these templates is controllable with Cascading Style Sheets (CSS), the technology that's used to design and colour the web. CSS is a complex and involved subject with a huge array of possibilities and options, and the Plone developers have been keen to give users the ability to change common parts of the Plone templates without having to resort to CSS hacking. This gives you the flexibility to make simple changes or create advanced CSS files that give you a unique design.

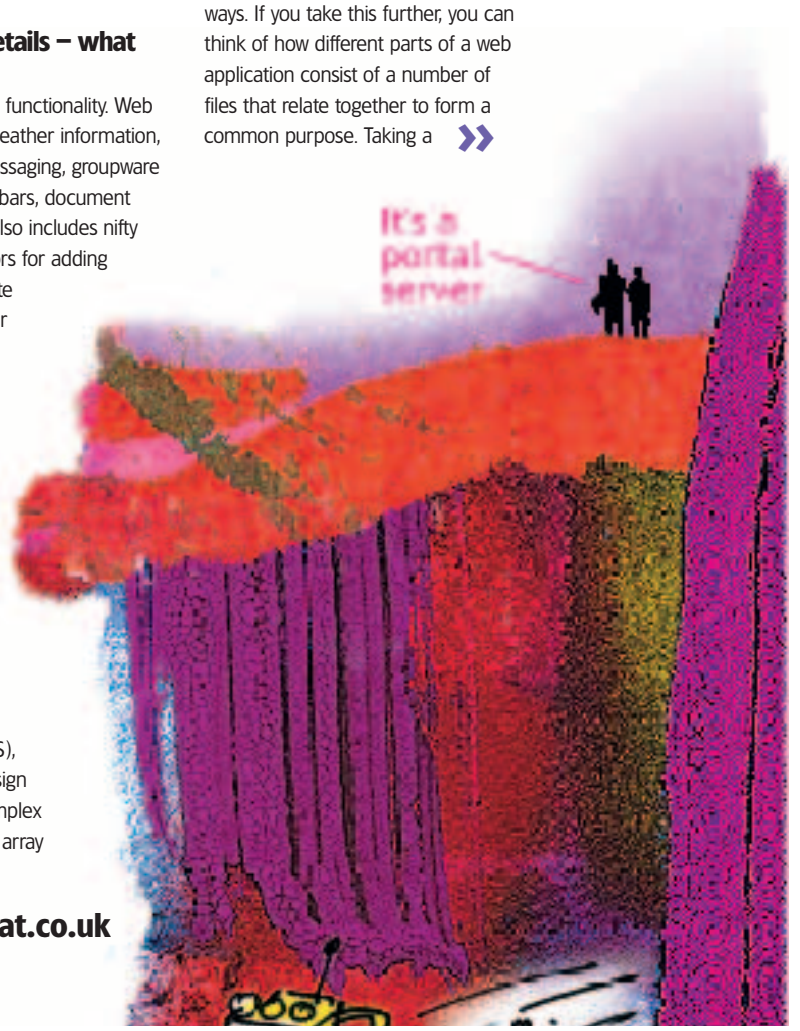
So Plone is flexible and feature-driven? Why do I sense a 'but' coming?

The only problem is complexity. With the sheer flexibility of the system comes an enormous amount of choice in how you set up the CMS. It's made Plone possibly the most complex open source CMS. In addition to the number of variable options and levels of flexibility, Plone is complicated in the way that the constraints applicable to some other CMSs don't apply to Plone. Unfortunately, this entails a bit of learning – you'll need to work out exactly what you can carry over to Plone from the CMS you're used to to implement your specific design.

OK, so I'll have to do a bit of work. Where do I start?

The first place to begin is the Plone architecture, and this in turn begins with Zope. Zope is a special application server which provides an underlying framework for building web applications. Several large organisation use it, including Red Hat, the US Navy and General Electric.

The intriguing bit about Zope is its object-oriented approach to web applications. If you think about the different types of information that are contained within a website, they are all merely objects that relate to one another in different ways. If you take this further, you can think of how different parts of a web application consist of a number of files that relate together to form a common purpose. Taking a



shopping basket as an example, there are several files that are written to implement the shopping basket functionality, but each of these different files is fundamentally related to the concept of a shopping basket object.

This concept also applies hierarchy to the mix. If you were to visit **www.foo.com**, you would be accessing the root directory of the web server. If you access **www.foo.com/bar/**, you would be accessing the **bar** folder that's further down the hierarchy in the web server.

How is Zope different from Plone, exactly?

Zope is the platform that Plone runs on. Zope provides a means of creating special objects that can interact with each other in different ways, and Zope also provides its own transactional object database called ZODB. It's as much a database platform as a web application platform.

Does this mean that Zope/Plone doesn't need a database such as *MySQL* or *PostgreSQL*?

That's right. Within the Plone system, virtually everything is provided to give you a complete application server. It's like the LAMP (Linux, *Apache*, *MySQL*, PHP) system in one – it includes its own database and even its own web server.

Fear not, though: you can use Plone with *Apache* (some members of the Plone community recommend *Apache* for use as an underlying web server). You can also connect some aspects of Plone to other databases, but putting the entire database in a separate server is difficult – mainly because ZODB provides an object database, whereas *MySQL*, *PostgreSQL* and the like provide a relational database.

Tell me more about these objects.

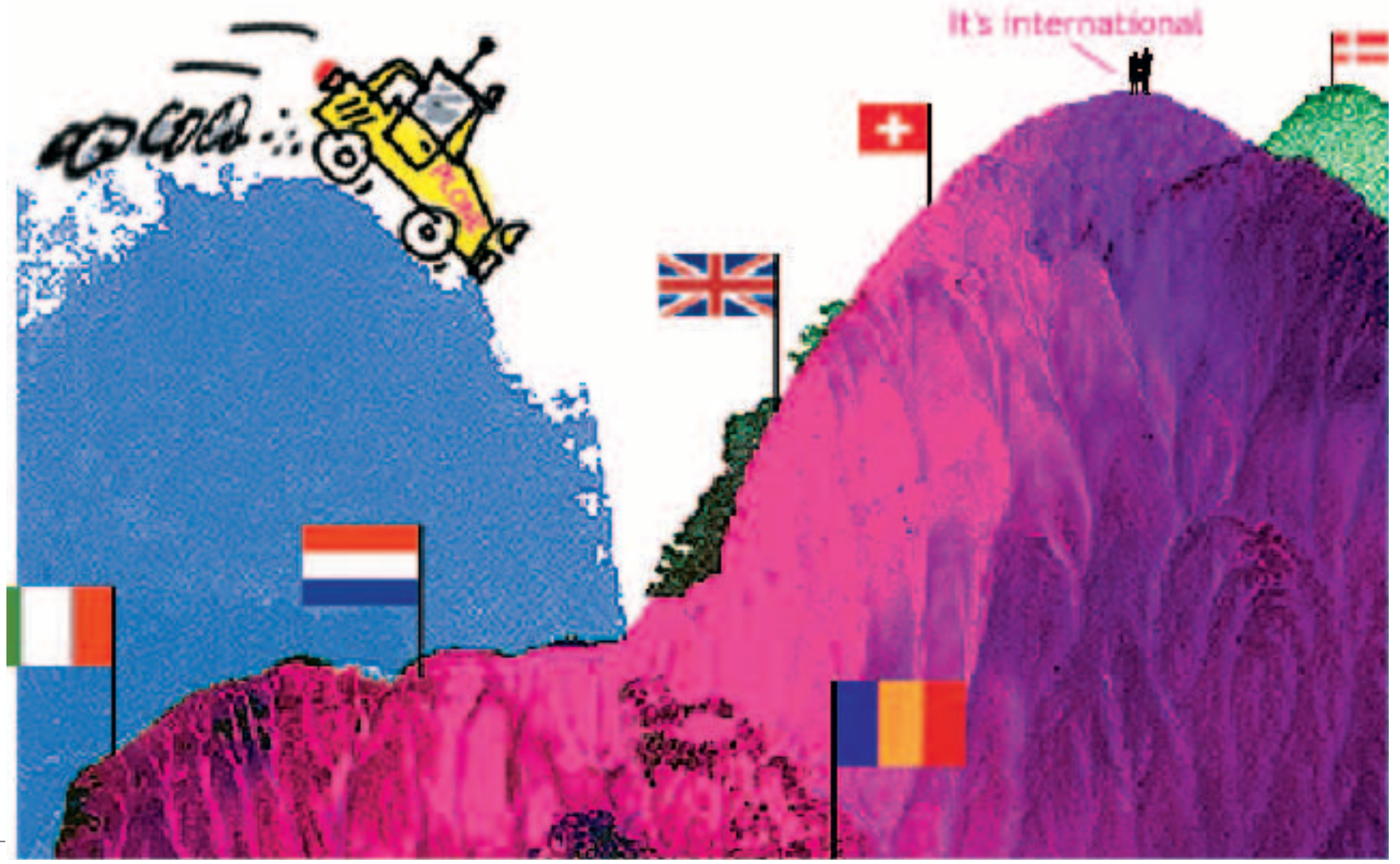
Zope objects are fundamental to everything you do with Zope and Plone. You have different objects for the different things you need to do in a CMS: objects to display information, objects to gather information from a user, and objects to process information. Then there are objects that deal with very specific types of functionality. These objects fit into three approximate categories:

- **Content.** This can include plain text, audio, video, spreadsheets, images, or any other type of content on your website.
- **Presentation.** These kinds of objects deal with presenting information to the user. This includes creating the design and layout of your pages.
- **Logic.** These objects provide scripted logic that can be used to process information and other objects. Logic is critical in ensuring that the CMS does exactly what you want it to do.

Although there are three broad categories, many of the objects' requirements can spread across categories – for example, there's no reason why you can't create a Presentation object that has some Logic coded into it. This is where that complexity issue can rear its head...

What kind of language is used to write these objects and create web applications with Plone and Zope?

The language at the core of the Plone and Zope system is Python. Although this may come as a surprise if you were expecting to see PHP running the show, Python does a remarkable job of creating



an easy-to-use language for dealing with these objects. Python is both simple and effective, and has proved to be really popular in recent times. One of the reasons for this popularity is that so much is built into the core Python libraries.

How does Python compare with PHP and Perl?

Many people seem to consider Perl as a similar language to that of the *bash* shell. In turn, some people consider PHP to share aspects with Perl, but also bits of *bash* shell and possibly even bits of C.

Python, on the other hand, shares many of its similarities with high-level languages such as Visual Basic, and it's more similar to VB than PHP or Perl. Python is also very similar to the C# language that has proved so popular with the .NET platform and the totally hip open source implementation of .NET know as Mono.

You've done a good job of selling Plone. What kind of server do I need to run it on?

With its interpreted Python foundation and comprehensive support built in for so many different kinds of functionality, Plone needs a little more oomph from your server than most other CMSs. Many Plone users and developers believe the best option is to run Plone on a dedicated server or dedicated hosting service.

Running Plone on a dedicated server meets its additional hardware requirements, and it's also useful to be able to SSH into your Plone box and edit your configuration.If you want to target your resources to the most put-upon resource for a Plone server, give it lots of RAM. Plone uses RAM extensively and you should have at least 100MB of it.

And how do I install it?

Plone is packaged for a number of distributions and is available at **www.plone.org**. The Plone team have worked hard to ensure that they release the software in a number of different supported package types. The Plone packages include the Zope application server that you need to run Plone; but if you just want Plone itself, you can get the Plone Core packages.

If you're running a distribution with an archive of available packages, you're best off downloading the software from this supported archive This will ensure that the many different elements in the Plone system are installed and configured correctly.

If you want to run Plone on a Windows or Mac OS X box, you can use one of the stand-alone installers available at **www.plone.org**. These packages provide the familiar Windows/Mac OS X point-and-click installer and are very simple to use.

Where should I start after installing Plone?

You'll first need to access the Zope server on the right port. This tends to differ from machine to machine, but you should check in your Zope configuration file to see which port you need to use. Inside **/etc** there should be a Zope directory. This varies among distributions, but it is likely to be called something like **zope** or **zopectl**.

Inside this directory there should be a file called **default.conf** or **zope.conf**, and inside that file will be a line such as 'HTTP-Port: 9673'. In this example the port has been set to 9673, and you can access this by visiting the local IP address that points to the machine you are currently using (127.0.0.1) and issue the port: **http://127.0.0.1:9673**. When you access this URL, you should see some information about Zope appear. This demonstrates that Zope is working!

Once you're in, begin by familiarising yourself with the Zope Management interface.

What's that?

An extensive web application giving you access to the many intricacies of the Zope application server and the ability to configure virtually anything involved with Zope.

You can access the Zope Management interface by appending **/manage** to the URL, as in **http://127.0.0.1:9673/manage**. You will then be presented with a login box in which you should type the username and password for your user account on your computer, and you will be given access to the management console.

When you fire up the management interface, you'll see a sidebar down the left-hand side and main view. The sidebar contains a list of folders and objects that you have access to. In the main part of the window, you can then view these resources and edit information about them.

Adding resources to the Zope system is also fairly straightforward. In the top right-hand side of the main part of the screen, you should be able to see a drop-down box with a large list of possible actions. This box gives you all the options you need to create events, methods, documents and files; to add scripts, images and more.

You can also use this box to create your Plone-managed website. Select the option to create a new site and then fill in the form in the main part of the screen to create your website.

With of all of this functionality and the companies you discussed earlier using it, how is Plone dealing with increasing demand?

Well, it's a comprehensive project with a large developer base behind it. Then there is a huge community of users who support the software on mailing lists, IRC channels and by writing tutorials and guides about using Plone. It's a good example of a large and successful open source project that benefits from an expansive network of volunteers.

To augment this community, and inkeeping with the trend of large open source projects setting up official organisations, the Plone Foundation was formed in 2004 to offer a more regulated side to Plone. Its duties include acting as the 'voice of Plone' for official announcements, press releases, and other communications; as well as the essential often overlooked function of and generating funds. The foundation has established an electoral process and anyone is welcome to get involved.

As helpful as you've been, I'd like to find out more about Plone. Where should I go next?

Your first port of call should be the Plone website at **www.plone.org**. This has a documentation section that's stacked full of information about Plone and also contains some tutorials. Aside from a general Google search for Plone tutorials, another potential avenue to explore for information is *The Definitive Guide to Plone* by Andy McKay (Apress, $44.99). The book provides a fantastic, easily accesible introduction to Plone and how to create a comprehensive Plone-based website.

If you're looking for specific help and assistance with the CMS, you should also join up on one of the mailing lists at **http://plone.org/documentation/lists**. There is also the #plone channel on the Freenode IRC network. Try using the irc.eu.freenode.net server to connect to the channel. LXF